PENGEMBANGAN SISTEM PENDUKUNG KEPUTUSAN PENGELOLAAN OPERASIONAL ASET TI BERBASIS LARGE LANGUAGE MODEL STUDI KASUS BSI UII



Disusun Oleh:

N a m a : Muhammad Ilham Mahfuzh

NIM : 20523139

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGEMBANGAN SISTEM PENDUKUNG KEPUTUSAN PENGELOLAAN OPERASIONAL ASET TI BERBASIS LARGE LANGUAGE MODEL STUDI KASUS BSI UII

TUGAS AKHIR

Disusun Oleh:

N a m a : Muhammad Ilham Mahfuzh

NIM : 20523139

Yogyakarta, 4 Juli 2026

Pembimbing,

(Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGEMBANGAN SISTEM PENDUKUNG KEPUTUSAN PENGELOLAAN OPERASIONAL ASET TI BERBASIS LARGE LANGUAGE MODEL STUDI KASUS BSI UII

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 30 Juli 2026

Tim Penguji

Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D. _____

Anggota 1

Kholid Haryono, S.T., M.Kom. _____

Anggota 2

Sri Mulyati, S.Kom., M.Kom. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Muhammad Ilham Mahfuzh

NIM : 20523139

Tugas akhir dengan judul:

PENGEMBANGAN SISTEM PENDUKUNG KEPUTUSAN PENGELOLAAN OPERASIONAL ASET TI BERBASIS LARGE LANGUAGE MODEL STUDI KASUS BSI UII

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 1 Juli 2026



(Muhammad Ilham Mahfuzh)

HALAMAN PERSEMBAHAN

Segala puji dan syukur saya panjatkan ke hadirat Allah SWT, Tuhan Yang Maha Esa, atas segala rahmat, hidayah, dan kekuatan yang telah diberikan sehingga skripsi ini dapat diselesaikan. Tanpa izin dan karunia-Nya, perjalanan ini tidak akan mungkin terlaksana.

Atas ridha-Nya pula, skripsi ini saya persembahkan dengan penuh rasa hormat dan cinta kepada Bunda dan Ayah tercinta, yang telah memberikan doa tanpa henti, kasih sayang tanpa syarat, dan pengorbanan tak terhingga. Setiap langkah yang saya ambil dalam perjalanan pendidikan ini tidak lepas dari dukungan moral dan spiritual yang senantiasa kalian berikan. Terima kasih telah mengajarkan arti ketulusan, kerja keras, dan keyakinan bahwa setiap usaha pasti menemukan jalannya.

Bapak Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D., selaku dosen pembimbing yang telah memberikan bimbingan, arahan, dan masukan berharga selama proses penelitian dan penyusunan skripsi ini. Terima kasih atas kesabaran dan dedikasi dalam membimbing saya menyelesaikan karya ini.

Selain dukungan keluarga dan bimbingan akademis, perjalanan skripsi ini juga tidak lepas dari kehadiran seseorang yang senantiasa menemani dalam setiap prosesnya. Munifa Haras, terima kasih telah menemani setiap proses pengerjaan skripsi ini dengan kesabaran dan dukungan yang tak pernah surut. Di saat saya lelah, kamu hadir dengan ketenangan. Di saat saya ragu, kamu hadir dengan keyakinan. Sebagaimana Tony Stark pernah menyampaikan ungkapan kasih sayang yang begitu mendalam kepada orang-orang yang berarti dalam hidupnya, demikian pula saya ingin menyampaikan rasa terima kasih dan penghargaan yang sebesar-besarnya atas setiap dukungan yang telah diberikan. Skripsi ini menjadi bukti bahwa kehadiran dan dukungan tersebut bukan sekadar rangkaian kata, melainkan sumber semangat dan kekuatan yang membantu saya melewati setiap tantangan selama proses penyusunannya.

Dukungan juga datang dari teman-teman yang senantiasa siap membantu. Paulus Caesario Dito, terima kasih telah menjadi teman diskusi yang selalu siap membantu. Di saat saya butuh bantuan teknis, kamu hadir dengan solusi. Di saat saya butuh teman bertukar pikiran, kamu hadir dengan ide. Terima kasih telah berbagi beban dan membuat perjalanan ini lebih ringan. Begitu pula dengan Kevin Martin Hendra Wijaya, terima kasih telah menjadi bagian dari perjalanan ini. Dukungan dan semangat yang kamu berikan turut mewarnai proses pengerjaan skripsi ini.

HALAMAN MOTO

"*Life is like walking, you take one step at a time*"

(Munifa Haras)

"Terbentur, terbentur, terbentur, terbentuk"

(Tan Malaka)

"*The noblest art is that of making others happy*"

(Phineas Taylor Barnum)

"Kesuksesan bukanlah tentang seberapa tinggi Anda melompat, tetapi seberapa keras Anda bangkit setelah jatuh."

(B.J. Habibie)

"*You see things. You keep quiet about them. And you understand*."

(Charlie)

"*Stay hungry, stay foolish*."

(Steve Jobs)

"*Sometimes you gotta run before you can walk*."

(Tony Stark)

KATA PENGANTAR

Dengan menyebut nama Allah SWT Yang Maha Pengasih lagi Maha Penyayang. Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul "PENGEMBANGAN SISTEM PENDUKUNG KEPUTUSAN PENGELOLAAN OPERASIONAL ASET TI BERBASIS LARGE LANGUAGE MODEL STUDI KASUS BSI UII". Skripsi ini disusun sebagai salah satu persyaratan kelulusan Program Studi Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Penelitian ini bertujuan untuk mengembangkan sistem pendukung keputusan pengelolaan aset teknologi informasi yang memanfaatkan kemampuan *Large Language Model* dalam memberikan rekomendasi alokasi, *insight* kondisi aset, dan evaluasi kesehatan aset. Melalui pendekatan *Design Science Research*, sistem ini dirancang untuk menjawab tantangan operasional dalam pengelolaan aset TI di lingkungan Badan Sistem Informasi (BSI) Universitas Islam Indonesia, sekaligus menguji efektivitas format *Token-Oriented Object Notation* (TOON) dalam mengoptimalkan penggunaan konteks LLM.

Selama proses penelitian dan penyusunan skripsi, penulis menghadapi berbagai tantangan, baik dalam hal pemahaman terhadap kompleksitas integrasi sistem, penyesuaian dengan standar pengembangan perangkat lunak, maupun manajemen waktu antara pengerjaan skripsi dan tanggung jawab lainnya. Namun demikian, berkat bimbingan, arahan, serta dukungan dari berbagai pihak, seluruh proses tersebut dapat dilalui dengan baik.

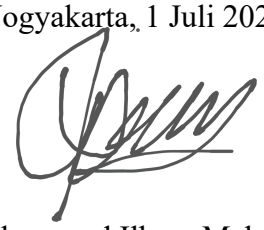
Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang senantiasa memberikan rahmat, kesehatan, dan kemudahan kepada penulis selama proses penelitian dan penyusunan skripsi ini.
2. Kedua orang tua penulis yang senantiasa memberikan dukungan, doa, dan motivasi tanpa henti. Terima kasih telah menjadi fondasi terkuat dalam setiap langkah yang penulis ambil dan mengajarkan arti ketulusan, kerja keras, serta keyakinan bahwa setiap usaha pasti menemukan jalannya.
3. Bapak Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D. selaku dosen pembimbing yang telah memberikan bimbingan, arahan, serta masukan yang sangat berarti dalam proses penyusunan skripsi ini.

4. Tim evaluator domain di lingkungan BSI UII yang telah memberikan waktu, *expertise*, dan masukan berharga untuk menilai kualitas keluaran sistem, serta seluruh pihak yang telah mendukung pelaksanaan penelitian di lingkungan BSI UII.
5. Seluruh dosen Program Studi Teknik Informatika Universitas Islam Indonesia yang telah memberikan bekal ilmu pengetahuan dan wawasan selama penulis menempuh masa studi.
6. Rekan-rekan seperjuangan yang telah memberikan dukungan, semangat, serta pengalaman berharga selama masa perkuliahan dan penyusunan skripsi ini.
7. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dan mendukung penulis selama proses penelitian dan penyusunan skripsi.

Penulis berharap skripsi ini dapat memberikan manfaat dan kontribusi, baik bagi pengembangan ilmu pengetahuan di bidang teknologi informasi, khususnya sistem pendukung keputusan berbasis kecerdasan buatan, maupun sebagai referensi bagi pihak-pihak yang berkepentingan. Penulis juga menyadari bahwa skripsi ini masih memiliki keterbatasan dan kekurangan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan demi penyempurnaan penelitian ini di masa yang akan datang.

Yogyakarta, 1 Juli 2026



(Muhammad Ilham Mahfuzh)

SARI

Badan Sistem Informasi (BSI) Universitas Islam Indonesia (UII) mengelola aset teknologi informasi dalam jumlah besar dengan pendekatan manual yang mengandalkan tracking kondisi berkala dan keputusan berbasis intuisi operator. Pendekatan ini menghadapi tantangan dalam konsistensi evaluasi kondisi aset, kecepatan pengambilan keputusan alokasi, serta efisiensi penggunaan konteks *Large Language Model* (LLM) untuk analisis kontekstual. Sebagai solusi atas permasalahan tersebut, dikembangkan sistem pendukung keputusan pengelolaan operasional aset TI berbasis LLM dengan tiga fitur utama AI *Assign Recommendation* untuk rekomendasi alokasi aset, AI *Asset Insight* untuk interpretasi kondisi aset, dan AI *Health Evaluation* untuk evaluasi kesehatan aset. Sistem ini mengadopsi format *Token-Oriented Object Notation* untuk mengoptimalkan representasi konteks data tabular sebelum diproses oleh LLM.

Pengembangan sistem dilakukan dengan pendekatan Design Science Research yang terdiri dari tiga iterasi desain untuk mengoptimasi representasi konteks, evaluasi kesehatan, dan mekanisme caching. Pengujian sistem meliputi pengujian fungsional, limited load test, fault injection, evaluasi kualitas keluaran AI menggunakan domain-based expert agreement evaluation dengan pasangan evaluator sesuai domain, serta evaluasi dukungan keputusan berbasis umpan balik operator. Hasil penelitian menunjukkan bahwa sistem beroperasi pada ruang uji terbatas dengan validitas struktur keluaran AI mencapai 26/26 kasus, 14 dari 15 kombinasi fitur-aspek memperoleh percent agreement 1,000, dan fault injection terhadap tiga fitur AI menghasilkan 45 uji tanpa kegagalan fatal pada skenario terbatas. Penggunaan TOON juga menghasilkan rata-rata reduksi token 49,8% dibanding JavaScript Object Notation pada konteks data ITAM penelitian ini. Sistem ini diposisikan sebagai artefak awal yang tidak menggantikan keputusan manusia, melainkan menyediakan rekomendasi, insight, dan evaluasi kesehatan aset yang tetap memerlukan verifikasi operator, serta diharapkan memberikan kontribusi pada pengembangan sistem pendukung keputusan berbasis kecerdasan buatan untuk aplikasi operasional.

Kata kunci: *Large Language Model*, Sistem Pendukung Keputusan, *Token-Oriented Object Notation*, Pengelolaan Aset TI, *Design Science Research.*

GLOSARIUM

API	*Application Programming Interface*, sekumpulan protokol dan alat untuk membangun aplikasi yang memungkinkan komunikasi antarsistem.
Backend	Bagian dari sistem aplikasi yang bekerja di sisi server, mengelola database, autentikasi, dan logika bisnis aplikasi.
Baseline	Referensi awal atau kondisi dasar yang digunakan sebagai pembanding untuk mengukur perubahan atau peningkatan performa sistem setelah implementasi solusi.
Boundary Case	Kasus dengan kondisi yang berada pada batas antara dua kategori atau status, sehingga lebih rentan terhadap variasi keluaran model bahasa.
Caching	Mekanisme penyimpanan sementara hasil komputasi untuk digunakan kembali pada permintaan berikutnya, mengurangi beban komputasi dan waktu respons.
Ceiling Effect	Fenomena dalam evaluasi di mana nilai kesepakatan mencapai batas maksimum (1,00) karena distribusi skor yang sangat homogen, sehingga metrik reliabilitas tidak lagi informatif.
Degraded Mode	Mode operasi sistem ketika layanan utama tidak tersedia, di mana sistem tetap berfungsi dengan kemampuan terbatas menggunakan mekanisme cadangan.
Endpoint	Titik akhir pada API yang menerima permintaan HTTP dan mengembalikan respons, mewakili satu fungsi spesifik dalam sistem.
Error Rate	Persentase kesalahan yang terjadi saat pengguna menggunakan sistem atau aplikasi.
Fault Injection	Teknik pengujian dengan sengaja memperkenalkan kegagalan pada komponen sistem untuk memverifikasi ketahanan dan mekanisme pemulihan sistem.
Framework	Kerangka kerja atau struktur konseptual yang digunakan sebagai panduan dalam pengembangan sistem.
Frontend	Bagian dari aplikasi yang berinteraksi langsung dengan pengguna, mencakup antarmuka dan pengalaman pengguna.

Inference	Proses model bahasa menghasilkan keluaran (*output*) berdasarkan masukan (input) yang diberikan, mencakup perhitungan probabilitas dan generasi teks.
ISO 55000	*International Organization for Standardization* yang mengatur pedoman dan prinsip dasar untuk manajemen aset.
ISO 550001	*International Organization for Standardization* yang menetapkan persyaratan untuk membangun, menerapkan, memelihara, dan meningkatkan Sistem Manajemen Aset.
Kappa *Paradox*	Fenomena ketika nilai koefisien Kappa mendekati nol atau tidak terdefinisi meskipun *percent agreement* tinggi, terjadi karena distribusi skor yang sangat homogen.
Latency	Waktu yang dibutuhkan sistem untuk memproses permintaan dan mengembalikan respons, biasanya diukur dalam milidetik.
Loat Test	Pengujian performa sistem dengan memberikan beban permintaan dalam jumlah tertentu untuk mengukur respons sistem di bawah tekanan.
Metrics	Ukuran kuantitatif yang digunakan untuk menilai performa, kualitas, atau karakteristik sistem, seperti latensi, t*hroughput*, atau *error rate*.
Mock Proxy	*Server* tiruan yang mensimulasikan perilaku layanan eksternal untuk tujuan pengujian, memungkinkan pengujian skenario kegagalan tanpa memengaruhi layanan produksi.
P50	Nilai persentil ke-50 dari distribusi latensi, menunjukkan bahwa 50% permintaan dilayani dalam waktu tersebut atau kurang. Merupakan ukuran latensi tipikal sistem.
P95	Nilai persentil ke-95 dari distribusi latensi, menunjukkan bahwa 95% permintaan dilayani dalam waktu tersebut atau kurang. Merupakan ukuran latensi kasus buruk (*worst-case*) yang lebih representatif daripada rata-rata.
Pipeline	Rangkaian komponen pemrosesan data yang terhubung secara sekuensial, di mana keluaran satu komponen menjadi masukan bagi komponen berikutnya.
Prompt Template	*Template* terstruktur yang digunakan untuk menyusun *prompt* yang dikirimkan ke model bahasa, memastikan konsistensi format dan kelengkapan konteks.

Rate Limiting	Mekanisme pembatasan jumlah permintaan yang dapat diproses sistem dalam periode waktu tertentu, mencegah *overload* dan menjaga stabilitas layanan.
Reasoning Model	Model bahasa yang memiliki kemampuan melakukan penalaran bertahap sebelum menghasilkan jawaban akhir, meningkatkan akurasi pada tugas kompleks.
Temperature	Parameter sampling model bahasa yang mengontrol tingkat randomisasi keluaran. Nilai rendah (mendekati 0) menghasilkan keluaran lebih deterministik, nilai tinggi menghasilkan keluaran lebih bervariasi.
Threshold	Nilai ambang batas yang ditentukan untuk metrik tertentu, di mana pelanggaran terhadap nilai ini memicu *alert* atau tindakan yang harus diambil.
Throttle	Mekanisme pengendalian laju pemrosesan permintaan untuk mencegah *overload* pada sistem atau layanan eksternal, biasanya dengan menambahkan jeda antarpermintaan.
Throughput	Jumlah permintaan yang dapat diproses sistem per satuan waktu, biasanya diukur dalam *request per second* (RPS).
Timeout	Batas waktu maksimum yang ditetapkan untuk menunggu respons dari layanan eksternal. Jika batas waktu terlampaui, permintaan dianggap gagal dan mekanisme pemulihan diaktifkan.
Tokenizer	Komponen yang mengubah teks menjadi token (unit dasar pemrosesan model bahasa) dan sebaliknya mengubah token kembali menjadi teks.
Top-p	Parameter sampling model bahasa yang membatasi pilihan token hanya dari subset dengan probabilitas kumulatif tertinggi (p), mengontrol variasi keluaran secara lebih adaptif dibanding *temperature* saja.
Virtual User	Unit beban dalam pengujian performa yang mensimulasikan satu pengguna yang mengirimkan permintaan secara bersamaan.

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI ix
GLOSARIUM x
DAFTAR ISI xiii
DAFTAR TABEL xvii
DAFTAR GAMBAR xix
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 4
1.4 Tujuan Penelitian 5
1.5 Manfaat Penelitian 6
1.6 Metodologi Penelitian 6
1.7 Sistematika Penulisan 7
BAB II LANDASAN TEORI 8
2.1 Manajemen Aset Teknologi Informasi 8
2.1.1 Konsep Dasar dan Siklus Hidup Aset 8
2.1.2 Kerangka Tata Kelola Aset ISO 55000 9
2.1.3 Aset Strategis, CAPEX, OPEX, dan Nilai Organisasi 10
2.2 Sistem Pendukung Keputusan 12
2.2.1 Karakteristik Sistem Pendukung Keputusan 12
2.2.2 Komponen Utama Sistem Pendukung Keputusan 12
2.2.3 Sistem Pendukung Keputusan dalam Konteks Manajemen Aset 13
2.3 Monitoring dan Evaluasi Aset 13
2.3.1 *Condition-Based Monitoring* 14
2.3.2 Pendekatan Pemeliharaan 14
2.3.3 Metrik Monitoring dan Interpretasi Operasional 15
2.4 Large Language Model 16
2.4.1 Gambaran Umum Cara Kerja LLM 16
2.4.2 Kemampuan LLM dalam Sistem Informasi 17
2.4.3 Keterbatasan LLM 17
2.5 Pemanfaatan LLM dalam Sistem Operasional 18
2.5.1 LLM sebagai Pendukung *Insight* Operasional 18
2.5.2 LLM untuk Rekomendasi Operasional 19
2.5.3 Pembatasan Domain dan Efisiensi Konteks 19
2.5.4 *Structured Prompting* dan *Context Engineering* 20
2.6 Sistem Berbasis Aturan 21
2.6.1 Kelebihan Sistem Berbasis Aturan 21
2.6.2 Keterbatasan Sistem Berbasis Aturan 22
2.7 Integrasi Pendekatan Berbasis Aturan dan Kecerdasan Buatan 22
2.7.1 *Human-in-the-Loop* 23

2.7.2 *Guardrail* dan Validasi 23
2.8 *Framework* Evaluasi pada Sistem Pendukung Keputusan Berbasis AI 24
2.8.1 Evaluasi Konsistensi Keluaran *Large Language Model* 25
2.9 Keandalan Sistem dan Ketahanan Layanan 26
2.9.1 Keandalan dalam Sistem Terdistribusi 27
2.9.2 Pola Ketahanan Layanan 27
2.10 Tinjauan Pustaka 28
2.10.1 Komparasi Penelitian Terdahulu 29
2.10.2 Analisis *Gap* dan *Positioning* Penelitian 32
2.10.3 Diagram Positioning Penelitian 33
2.11 Kerangka Pemikiran 35
BAB III METODOLOGI PENELITIAN 37
3.1 Jenis dan Pendekatan Penelitian 37
3.2 Lokasi, Subjek, dan Ruang Lingkup Penelitian 38
3.3 Sumber Data dan Teknik Pengumpulan Data 40
3.3.1 Instrumen Pengumpulan Data 42
3.3.2 Observasi Monitoring NDC 42
3.4 Metode Pengembangan Sistem 42
3.4.1 Analisis Kebutuhan Fungsional dan Nonfungsional 43
3.4.2 Desain Arsitektur dan Model Data 43
3.4.3 Implementasi Sistem Berlapis 43
3.4.4 Pengujian Sistem 44
3.4.5 Evaluasi dan Iterasi 44
3.5 Perancangan Arsitektur dan Alur Sistem 45
3.5.1 Perancangan Proses Bisnis Operasional 45
3.5.2 Arsitektur Umum Sistem 48
3.5.3 *Activity Diagram* 49
3.5.4 Arsitektur Umum Sistem 51
3.5.5 Alur AI *Assign Recommendation* 52
3.5.6 Alur AI *Asset Insight* 53
3.5.7 Alur AI *Health Evaluation* 54
3.5.8 Alur Integrasi Monitoring dan *Alert* 55
3.6 Perancangan Data dan Aturan Bisnis 56
3.7 Perancangan Metode AI pada Sistem 61
3.7.1 Penyusunan Konteks Input 61
3.7.2 Metode AI untuk Assign Recommendation 63
3.7.3 Metode AI untuk *Asset Insight* 63
3.7.4 Metode AI untuk *Health Evaluation* 64
3.7.5 Mekanisme Pengendalian: *Retry, Fallback*, dan Validasi Output 68
3.7.6 Mekanisme Pengendalian: *Retry*, *Fallback*, dan Validasi Output 70
3.7.7 Pemilihan Model *Large Language Model* 71
3.7.8 Validasi Keluaran AI dan Penanganan Keluaran Tidak Valid 73
3.8 Metode Pengujian dan Evaluasi 73
3.8.1 Pengujian Fungsional 74
3.8.2 Pengujian Integrasi 74
3.8.3 Pengujian Performa dan Keandalan 74
3.8.4 Pengujian Skenario Bisnis 75
3.8.5 Evaluasi Keluaran AI Berbasis *Domain-Based Expert Evaluation* 76
3.8.6 Evaluasi DSS Berbasis *Feedback* Operator 79
3.8.7 Metode Pengukuran Eksploratoris Efisiensi Token 81

3.9 Indikator Keberhasilan 82
3.10 Etika Penelitian 84
3.11 Keterbatasan Penelitian 84
3.11.1 Daftar Keterbatasan 84
3.11.2 Ancaman Validitas dan Mitigasi 85
3.12 Reproduksibilitas Penelitian 86
BAB IV HASIL PENELITIAN, PENGUJIAN, DAN PEMBAHASAN 88
4.1 Gambaran Umum Hasil Penelitian 88
4.2 Hasil Fitur Utama Penelitian 89
4.2.1 Ringkasan Domain yang Relevan 89
4.2.2 Hasil AI *Assign Recommendation* 90
4.2.3 Hasil AI *Asset Insight* 94
4.2.4 Hasil AI *Health Evaluation* 96
4.2.5 Hasil Integrasi Monitoring dan *Alert* 98
4.3 Hasil Pengujian Fungsional 99
4.3.1 Evaluasi Dukungan Proses Bisnis 101
4.4 Hasil Pengujian Integrasi 102
4.5 Evaluasi Kuantitatif Performa *Endpoint* dan Efisiensi Data 103
4.5.1 Hasil *Fault Injection* terhadap Mekanisme *Retry* dan *Fallback* 107
4.5.2 Evaluasi Efisiensi Representasi Konteks (TOON vs JSON) 110
4.6 Perbandingan Nilai Sistem 112
4.7 Evaluasi Kualitas Keluaran AI terhadap Keputusan Operasional 112
4.7.1 Tujuan Evaluasi 112
4.7.2 Metode Evaluasi 113
4.7.3 Data Uji dan Prosedur 113
4.7.4 Kriteria Penilaian 114
4.7.5 Output AI Representatif 115
4.7.6 Hasil Evaluasi *Rater* Berbasis *Agreement* 115
4.7.7 Konsistensi Output AI pada Eksekusi Ulang 117
4.7.8 Analisis *Disagreement Rater* 120
4.7.9 Hasil *Feedback* Operator terhadap Dukungan Keputusan 121
4.7.10 Keterbatasan LLM pada Kesimpulan Evaluasi 123
4.8 Rekam Jejak Penyempurnaan Artefak berdasarkan Siklus DSR 123
4.8.1 Iterasi 1: Evolusi Format JSON ke TOON 123
4.8.2 Iterasi 2: Stratifikasi Kelas Risiko *Health Evaluation* 124
4.8.3 Iterasi 3: Mekanisme *Caching* untuk efisiensi 125
4.9 Analisis Keterbatasan dan Risiko 126
4.9.1 Kompatibilitas Integrasi *Endpoint* 126
4.9.2 Ketergantungan pada Layanan AI Eksternal 126
4.9.3 Konsistensi Output AI pada Input Identik 127
4.9.4 Keterbatasan Data *Event* 127
4.9.5 Batas Generalisasi Implementasi 128
4.10 Pembahasan Hasil dan Keterkaitan Teori 129
4.10.1 Implikasi Praktis untuk Operasional BSI UII 131
BAB V KESIMPULAN DAN SARAN 133
5.1 Kesimpulan 133
5.1.1 Kontribusi Penelitian 135
5.1.2 Keterbatasan Hasil 135
5.1.3 Pelajaran dari Siklus *Design Science Research* 136
5.2 Saran 137

DAFTAR PUSTAKA 139
LAMPIRAN 143
Lampiran 1. Daftar Kasus, Rubrik *Rater*, dan Data Mentah Skor *Rater* 143
1.1 Daftar Kasus Evaluasi 143
1.2 Rubrik Penilaian *Rater* 144
1.3 Raw Data Score Rater 145
1.4 *Percent Agreement* 154
1.5 Dokumentasi *Form* 155
Lampiran 2. Instrumen dan Hasil Evaluasi Likert 157
2.1 *Instrument* Likert 157
2.2 Hasil Liker 158
2.3 Dokumentasi Sesi 158
Lampiran 3. Surat Izin Penelitian dan Lembar Persetujuan 160

DAFTAR TABEL

Tabel 2.1 Prinsip tata kelola aset yang diadopsi dari kerangka ISO 55000 10
Tabel 2.2 Perbandingan karakteristik CAPEX dan OPEX 11
Tabel 2.3 Komponen utama sistem pendukung keputusan 13
Tabel 2.4 Contoh metrik monitoring dan interpretasi umum 15
Tabel 2.5 Pola umum ketahanan layanan 28
Tabel 2.6 Komparasi penelitian terdahulu 29
Tabel 2.7 Pemetaan teori ke implementasi sistem 35
Tabel 3.1 Pemetaan tahap *Design Science Research* pada penelitian 38
Tabel 3.2 Ruang lingkup penelitian 39
Tabel 3.3 Kategori dan sumber data penelitian 40
Tabel 3.4 Ringkasan tahapan pengembangan sistem 44
Tabel 3. 5 Perbandingan proses operasional pencatatan terpisah dan berbantuan sistem 47
Tabel 3. 6 *Use case* inti sistem 48
Tabel 3.7 Entitas data utama 56
Tabel 3. 8 Relasi inti perancangan basis data 58
Tabel 3. 9 Klasifikasi kelas risiko aset pada AI *Health Evaluation* 59
Tabel 3.10 *Use case* inti 60
Tabel 3.11 Ketenuan ambang kesehatan operasional berbasis metri 65
Tabel 3. 12 Komponen model pengambilan keputusan AI *Health Evaluation* 68
Tabel 3. 13 Aturan keputusan AI *Health Evaluation* 69
Tabel 3.14 Konfigurasi integrasi LLM pada penelitian 72
Tabel 3.15 Komposisi kasus evaluasi kualitas keluaran AI 75
Tabel 3.16 Profil evaluator penelitian 76
Tabel 3.17 Pemetaan evaluator berbasis domain 76
Tabel 3.18 Instrumen *feedback* operator (Likert 1-5) untuk evaluasi DSS 80
Tabel 3.19 Skenario uji bisnis yang dieksekusi 80
Tabel 3.20 Ringkasan skenario pengukuran token 81
Tabel 3.21 Indikator keberhasilan penelitian 82
Tabel 3.22 Keterlacakan rumusan masalah, metode, dan indikator 83
Tabel 3.23 Ancaman terhadap validitas dan mitigasi 85
Tabel 4.1 *Pair matching* desain BAB III terhadap bukti BAB IV 89
Tabel 4.2 Ringkasan domain pendukung kontribusi penelitian 89

Tabel 4.3 Protokol pengukuran waktu seleksi awal 92
Tabel 4.4 Status pelaksanaan pengukuran waktu seleksi awal..... 93
Tabel 4.5 Ringkasan hasil pengujian fungsional..... 99
Tabel 4.6 Detail pengujian modul inti..... 100
Tabel 4.7 Evaluasi dukungan proses bisnis operasional 101
Tabel 4.8 Ringkasan hasil pengujian integrasi..... 102
Tabel 4.9 Indikator konfigurasi keandalan yang diterapkan 103
Tabel 4.10 Hasil observasi latensi *backend* pada *limited-concurrency load test* 104
Tabel 4.11 Rincian p50/p95 *backend* pada *limited-concurrency load test*..... 104
Tabel 4.12 Validitas struktur output AI representatif..... 106
Tabel 4.13 Hasil validasi respons *error* terkendali 106
Tabel 4.14 Hasil *faulit injection* per skenario kegagalan dan fitur AI (agregat lima iterasi) 108
Tabel 4.15 Verifikasi aktivasi *retry* per lapis dan strategi *fallback* per fitur (agregat 15 uji)109
Tabel 4.16 Hasil Pengukuran Pengurangan Token Konteks (TOON vs JSON) 110
Tabel 4.17 Perbandingan sebelum dan sesudah implementasi..... 112
Tabel 4.18 Perbandingan pendekatan *rule-only* dan kombinasi AI + *rule-based* 112
Tabel 4.19 Ringkasan output AI representatif..... 115
Tabel 4.20 Hasil *domain-based expert agreement evaluation* 116
Tabel 4.21 Ringkasan konsistensi output AI pada eksekusi ulang final 118
Tabel 4.22 Analisis *disagreement rater* 120
Tabel 4.23 Hasil *feedback* operator..... 122
Tabel 4.24 Pemetaan indikator keberhasilan terhadap bukti hasil 128

DAFTAR GAMBAR

Gambar 2.1 Siklus hidup aset teknologi informasi 9
Gambar 2.2 Hubungan data, model, dan antarmuka pada sistem pendukung keputusan..... 13
Gambar 2.3 Alur monitoring aset menuju tindakan pemeliharaan..... 16
Gambar 2.4 Proses umum pemanfaatan LLM pada data operasional 17
Gambar 2.5 *Positioning* penelitian 34
Gambar 2.6 Kerangka pemikiran penelitian..... 36
Gambar 3.1 Alur penelitian 38
Gambar 3. 2 BPMN proses operasional berbasis pencatatan terpisah 46
Gambar 3. 3 BPMN proses operasional berbantuan sistem pendukung keputusan 47
Gambar 3. 4 *Use case* diagram sistem pendukung keputusan pengelolaan aset TI 48
Gambar 3. 5 *Activity diagram* proses pengelolaan aset pascaakuisisi..... 50
Gambar 3. 6 *Activity diagram* AI *Health Evaluation* 51
Gambar 3.7 Arsitektur umum *system* 52
Gambar 3.8 *Sequence* diagram alur AI *Assign Recommendation* 53
Gambar 3.9 *Sequence* diagram alur AI *Asset Insight* 54
Gambar 3.10 Sequence diagram alur AI *Health Evaluation* 55
Gambar 3.11 Ringkasan model data utama (ERD) 57
Gambar 3.12 Representasi konteks TOON untuk aset monitoring. 62
Gambar 3.13 *Template prompt AI Assign*. 63
Gambar 3.14 *Template prompt AI Insight.* 64
Gambar 3.15 Alur keputusan evaluasi kesehatan aset dengan *guardrail* berbasis aturan..... 67
Gambar 3.16 *Template prompt AI Health.* 68
Gambar 4.1 *Dashboard system.*..... 90
Gambar 4.2 *Response* AI *Assign Recommendation* 91
Gambar 4.3 Tampilan hasil AI *Assign Recommendation*..... 92
Gambar 4.4 Respons AI *Asset Insight*..... 94
Gambar 4.5 Tampilan hasil AI *Asset Insight*..... 95
Gambar 4.6 Respons AI *Health Evaluation* 96
Gambar 4.7 Tampilan status kesehatan aset..... 97
Gambar 4.8 *event alert* yang telah dianonimkan..... 99
Gambar 4.9 Plot latensi p95 pada skenario 1, 5, dan 10 VUs untuk tiga *endpoint* 105

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan Teknologi Informasi (TI) telah menjadi faktor penting dalam meningkatkan efisiensi operasional dan produktivitas organisasi modern. Di lingkungan pendidikan tinggi, infrastruktur TI yang andal menjadi penopang utama penyelenggaraan kegiatan akademik, administrasi, dan layanan digital. Dalam konteks tersebut, *Information Technology Asset Management* (ITAM) memegang peran strategis untuk memastikan bahwa aset teknologi dikelola secara optimal sepanjang siklus penggunaanya, mulai dari perencanaan, kebutuhan, pengadaan, pengoperasian, pemeliharaan, hingga penggantian atau penghentian penggunaan (Albalghouni et al., 2018; Stone et al., 2018).

Penelitian ini mengembangkan Sistem Pendukung Keputusan Manajemen Aset TI Berbasis *Large Language Model* (LLM*)* melalui studi kasus di Badan Sistem Informasi (BSI) Universitas Islam Indonesia (UII). BSI UII merupakan unit pelaksana teknis yang bertanggung jawab atas perancangan, pengembangan, operasional, dan layanan sistem informasi bagi seluruh civitas akademika dengan jumlah pengguna sekitar 30.000 orang. Pemilihan studi kasus ini dilakukan karena unit memiliki kebutuhan nyata untuk pengelolaan aset TI yang konsisten pada skala operasional institusi.

Fase pascaakuisisi pada penelitian ini merujuk pada tahapan setelah aset diterima dan tercatat secara administratif, yang mencakup penggunaan operasional, pemeliharaan, monitoring kondisi, hingga penghentian penggunaan (*disposal*). Adapun proses pengadaan dan pencatatan awal aset tetap dikelola melalui sistem perangkat lunak *Enterprise Resorce Planning* (ERP) berupa *Systems, Applications, and Products in Data Processing* (SAP*)* pada tingkat universitas.

Berdasarkan observasi awal, pengelolaan aset di BSI UII belum sepenuhnya terintegrasi pada seluruh siklus hidup aset. Sistem SAP yang digunakan telah mendukung fase akuisisi dan pencatatan awal aset, termasuk pengadaan dan *Purchase Order* (PO). Namun, pada fase pascaakuisisi seperti pencatatan riwayat isu, pemeliharaan, usia pakai, dan status kesehatan aset, pengelolaan masih berjalan sporadis. Data operasional aset setidaknya tersebar pada tiga media pencatatan utama, yaitu SAP, *spreadsheet unit*, dan catatan manual. Kondisi tersebut menyebabkan informasi aset tidak selalu konsisten antara catatan digital, riwayat operasional,

dan kondisi aktual di lapangan. Dalam konteks ini, SAP berperan sebagai sistem akuisisi tingkat universitas, sedangkan kebutuhan operasional pascaakuisisi BSI menjadi fokus pengembangan sistem pada penelitian ini.

Untuk memperkuat penelitian, observasi operasional awal yang dilakukan secara informal di lingkungan BSI UII mengindikasikan bahwa proses triase kondisi aset secara manual membutuhkan waktu yang tidak sedikit, umumnya memerlukan lebih dari lima menit per permintaan karena operator perlu merekonsiliasi data dari beberapa sumber, seperti SAP, *spreadsheet* unit, dan catatan manual, sebelum mengambil keputusan. Indikasi awal ini diperkuat secara kualitatif melalui wawancara terhadap 3 personel operasional BSI UII, yaitu Ketua Tim Administrasi & Manajemen Proyek (AMP), staf *Network Data Center* (NDC), dan staf inventaris aset, yang mengonfirmasi kebutuhan percepatan interpretasi kondisi aset dan standarisasi dasar pengambilan keputusan.

Ketergantungan pada proses manual membuat evaluasi kondisi aset masih banyak bergantung pada pengalaman personal petugas dan belum memiliki standar penilaian yang seragam. Pengambilan keputusan operasional juga cenderung bersifat reaktif dan membutuhkan waktu lebih lama seiring meningkatnya jumlah serta kompleksitas aset. Kondisi tersebut berpotensi menimbulkan dampak negatif terhadap kinerja operasional, seperti keterlambatan respons, kurang tepatnya alokasi sumber daya, dan meningkatnya risiko *downtime* yang tidak terdeteksi lebih awal (Maletič et al., 2020). Salah satu pendekatan yang umum dilakukan untuk mengatasi permasalahan pengelolaan aset adalah digitalisasi proses melalui sistem informasi berbasis aturan (*rule-based*) (Setiawan et al., 2019).

Meskipun sistem berbasis aturan mampu memberikan struktur kontrol yang pasti, pendekatan ini terbatas pada konteks yang stabil dan eksplisit. Untuk konteks yang lebih beragam dan tidak terstruktur, literatur mencatat munculnya alternatif, seperti *fuzzy logic*, *neural network*, dan *case-based reasoning* yang dapat melengkapi pendekatan *rule-based* (Reddy & Fields, 2022). Dalam konteks BSI UII, data operasional aset bersifat semi-terstruktur dan mengandung catatan teks bebas seperti deskripsi isu, catatan pemeliharaan, dan observasi teknisi. Pendekatan berbasis aturan konvensional sulit mengakomodasi variasi narasi ini karena membutuhkan pola yang eksplisit dan stabil, sementara model klasifikasi tradisional memerlukan data latih berlabel yang umumnya tidak tersedia dalam konteks operasional sehari-hari. Keadaan ini membuka peluang pemanfaatan *Large Language Model* (LLM), yang menawarkan kemampuan analisis kontekstual terhadap data semi-terstruktur tanpa memerlukan pelatihan ulang (Durachman et al., 2026). Namun, LLM memiliki kelemahan

inheren terkait konsistensi dan determinisme keluaran yang berisiko pada sistem operasional (Huang et al., 2025).

Berdasarkan kondisi tersebut, integrasi LLM ke proses operasional tidak cukup dilakukan dengan menambahkan komponen AI semata, tetapi harus disertai kontrol proses agar relevan dengan kebutuhan domain dan tetap andal. Akibatnya, dibutuhkan pendekatan yang tidak hanya mendigitalisasi pencatatan, tetapi juga mampu menginterpretasi data operasional secara adaptif sambil menjaga kontrol keputusan. Oleh karena itu, penelitian ini berfokus pada perancangan dan pengembangan Sistem Pendukung Keputusan Manajemen Aset TI Berbasis LLM sebagai sistem pengelolaan aset TI terintegrasi yang mendukung rekomendasi alokasi, evaluasi kondisi, dan penyajian *insight* operasional di lingkungan BSI UII. Berbeda dari pendekatan digitalisasi pencatatan atau otomatisasi berbasis aturan statis pada umumnya, sistem ini mengombinasikan kemampuan LLM dalam memahami konteks historis aset dengan mekanisme *guardrail* berbasis aturan untuk menjaga konsistensi, auditabilitas, dan keandalan proses operasional pada fase pascaakuisisi. Sistem ini dirancang untuk melengkapi SAP pada fase pascaakuisisi, bukan menggantikan fungsi pengadaan dan pencatatan awal aset yang telah berjalan pada tingkat universitas.

Selain integrasi *guardrail*, pemanfaatan LLM pada data operasional aset menghadapi tantangan lain yaitu beban token konteks yang besar berpotensi memotong informasi relevan *context truncation* dan meningkatkan biaya inferensi. Literatur terbaru menempatkan perhatian ini dalam disiplin *context engineering*, yaitu optimasi sistematis muatan konteks LLM melalui kompresi struktural dan penyusunan konteks terstruktur (Mei et al., 2025; White et al., 2023). Namun, kajian konteks LLM secara spesifik untuk domain manajemen aset TI pascaakuisisi masih terbatas, format umum seperti JSON lebih berperan sebagai format pertukaran data, bukan sebagai strategi penyusunan konteks yang mempertimbangkan efisiensi token. Penelitian ini menelaah pilihan format representasi konteks sebagai bagian dari desain artefak.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

a. Bagaimana merancang dan mengimplementasikan Sistem Pendukung Keputusan Manajemen Aset TI Berbasis LLM yang mengintegrasikan modul masterdata, manajemen aset, peminjaman/pengembalian, pencatatan isu, pemeliharaan, dan monitoring pada lingkungan BSI UII?

b. Bagaimana memanfaatkan LLM yang dikendalikan mekanisme *guardrail* berbasis aturan untuk menghasilkan rekomendasi alokasi aset, *insight* operasional aset, dan evaluasi kondisi kesehatan aset, yang kualitas keluarannya diukur melalui validitas struktur output, kesesuaian terhadap kondisi kasus, relevansi alasan, keakuratan klaim, konsistensi hasil pada pengujian berulang terbatas, serta penilaian evaluator berbasis domain?
c. Bagaimana merancang dan menguji secara terbatas mekanisme keandalan pada integrasi LLM ke proses operasional pengelolaan aset di BSI UII, yang diukur melalui waktu respons *endpoint*, *error rate* pada *limited-concurrency load test*, ketahanan dasar sistem terhadap kondisi input atau akses tidak valid, serta ketahanan mekanisme *retry* dan *fallback* terhadap gangguan layanan AI eksternal?

Rumusan masalah pertama memastikan artefak sistem tersedia, terintegrasi, dan berjalan secara fungsional dengan representasi konteks yang efisien. Rumusan masalah kedua memastikan keluaran berbasis LLM, yang tetap dikendalikan oleh aturan domain, memiliki nilai operasional bagi pengguna. Sedangkan rumusan masalah ketiga memastikan rancangan mekanisme keandalan sistem teruji pada ruang uji terbatas ketika komponen AI diintegrasikan ke proses kerja. Hal ini berati, penelitian ini tidak hanya memiliki target keberhasilan teknis pada implementasi, tetapi juga menilai keterkendalian, ketervalidasian mekanisme keandalan, dan nilai operasional artefak secara menyeluruh.

1.3 Batasan Masalah

Agar penelitian ini terarah dan fokus pada tujuan yang telah ditetapkan, batasan masalah dalam penelitian ini adalah sebagai berikut:

a. Studi kasus dan implementasi sistem dibatasi pada lingkungan kerja Badan Sistem Informasi UII.
b. Pengelolaan aset difokuskan pada aset teknologi informasi, baik aset fisik maupun aset digital tertentu, seperti lisensi dengan evaluasi fitur pendukung keputusan berbasis LLM diprioritaskan pada aset infrastruktur kritis atau bernilai tinggi yang mendukung kontinuitas layanan.
c. Cakupan sistem meliputi modul masterdata, manajemen aset, peminjaman/pengembalian, pencatatan isu, pemeliharaan, dan monitoring.
d. Fitur pendukung keputusan berbasis LLM yang dikembangkan mencakup rekomendasi alokasi aset, penyajian *insight* operasional aset, dan evaluasi kondisi kesehatan aset.

e. Integrasi AI difokuskan pada pemanfaatan layanan LLM sebagai komponen pendukung keputusan, bukan sebagai pengganti penuh pengambilan keputusan manusia.
f. Pengujian sistem dilakukan pada lingkungan pengembangan dengan data aset dan histori operasional yang tersedia selama periode observasi penelitian April - Juni 2026. Periode ini dipilih berdasarkan ketersediaan data operasional dari BSI UII dan dinilai memadai untuk menjalankan siklus *design–evaluate–refine* pada paradigma *Design Science Research*, mengingat fokus penelitian adalah evaluasi artefak sistem pada ruang uji terbatas, bukan studi longitudinal terhadap perubahan kondisi aset dalam jangka panjang.
g. Penelitian ini tidak membahas pengembangan *Role-Based Access Control* (RBAC) granular tingkat lanjut, dashboard analitik enterprise tahap akhir, maupun arsitektur multitenant atau multiorganisasi.
h. Penelitian ini tidak mencakup integrasi dua arah secara penuh dengan seluruh sistem eksternal organisasi, termasuk SAP. Pemanfaatan data SAP pada penelitian ini dibatasi sebagai referensi akuisisi awal secara terbatas, misalnya melalui *snapshot* atau pun ekspor data sesuai kebutuhan operasional studi kasus.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

a. Menghasilkan Sistem Pendukung Keputusan Manajemen Aset TI Berbasis LLM pada lingkungan BSI UII yang terverifikasi mampu menjalankan modul inti, seperti masterdata, aset, peminjaman atau pengembalian, isu, pemeliharaan, dan monitoring secara *end-to-end* sesuai skenario uji fungsional yang ditetapkan.
b. Menghasilkan implementasi fitur AI *(assign recommendation, asset insight*, dan *health evaluation*) yang dikendalikan mekanisme *guardrail* berbasis aturan dan memberikan keluaran terstruktur serta relevan untuk dukungan keputusan operasional, dengan indikator validitas struktur output, kesesuaian terhadap kondisi kasus, relevansi alasan, keakuratan klaim, konsistensi hasil pada pengujian terbatas, dan penilaian evaluator berbasis domain.
c. Menghasilkan rancangan dan implementasi mekanisme keandalan integrasi AI berbasis *retry, backoff, fallback*, dan *guardrail* yang teruji menjaga kontinuitas layanan pada skenario uji keandalan terbatas melalui indikator waktu respons *endpoint*, validitas struktur output pada *limited-concurrency load test*, ketahanan dasar sistem terhadap kondisi input atau akses tidak valid, serta ketahanan mekanisme *retry* dan *fallback* terhadap gangguan layanan AI eksternal.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

a. Bagi BSI UII, penelitian ini diharapkan dapat membantu pengelolaan aset TI menjadi lebih terintegrasi, mendukung pengambilan keputusan berbasis data, dan meningkatkan kecepatan identifikasi risiko operasional aset, melalui *dashboard* status kesehatan aset serta alur rekomendasi alokasi berbasis AI yang berjalan di atas data operasional aktual.
b. Bagi pengembangan keilmuan, penelitian ini diharapkan memberikan kontribusi berupa model integrasi LLM dan *rule-based* pada domain IT *Asset Managemen*t yang tetap menjaga kontrol, auditabilitas, dan keandalan proses operasional.
c. Bagi penelitian selanjutnya, penelitian ini diharapkan menjadi referensi implementatif untuk adaptasi model integrasi LLM dan *rule-based* pada institusi pendidikan tinggi yang memiliki karakteristik data semi-terstruktur dan kebutuhan operasional serupa.

1.6 Metodologi Penelitian

Penelitian ini menggunakan pendekatan *Design Science Research* (DSR) yang berfokus pada perancangan, pengembangan, dan evaluasi artefak sistem. Pendekatan ini dipilih karena keluaran utama penelitian adalah artefak sistem yang dapat diimplementasikan dan diuji, bukan sekadar deskripsi fenomena atau persepsi pengguna. Metode survei murni tidak memadai untuk menjawab kebutuhan validasi teknis *end-to-end,* sedangkan eksperimen murni tidak menjadi fokus utama karena penelitian ini menekankan pembangunan solusi operasional pada konteks studi kasus nyata. Pendekatan DSR ini dijalankan melalui kerangka enam tahap Peffers, yaitu *problem identification*, *definition of objectives*, *design and development*, *demonstration*, *evaluation*, dan *communication* (Peffers et al., 2007), dengan fokus utama pada tahap *design and development*, *demonstration*, dan *evaluation*. Oleh karena itu, pendekatan ini dinilai paling sesuai untuk menjembatani kebutuhan desain, implementasi, dan verifikasi sistem.

Tahapan pengembangan meliputi analisis kebutuhan, perancangan arsitektur dan model data, implementasi sistem, serta pengujian dan evaluasi yang mencakup uji fungsional, uji integrasi, uji keandalan layanan AI, dan evaluasi kualitas keluaran AI melalui penilaian evaluator berbasis domain. Rincian lengkap mengenai tahapan penelitian, desain arsitektur, dan metode pengujian diuraikan pada BAB III.

1.7 Sistematika Penulisan

Laporan tugas akhir ini disusun dengan sistematika penulisan sebagai berikut:

a. BAB I PENDAHULUAN.

Bab ini membahas latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan laporan tugas akhir.

b. BAB II LANDASAN TEORI.

Bab ini berisi pembahasan teori-teori yang relevan dan penelitian terdahulu yang mendukung topik penelitian, termasuk konsep IT *Asset Management*, *Decision Support System*, pemantauan kondisi aset, *Large Language Model*, sistem berbasis aturan, integrasi AI dan aturan, keandalan sistem, serta posisi penelitian.

c. BAB III METODOLOGI PENELITIAN.

Bab ini menjelaskan jenis dan pendekatan penelitian, lokasi dan ruang lingkup studi, sumber data, tahapan pengembangan sistem, perancangan arsitektur, perancangan data dan aturan bisnis, strategi integrasi AI, serta metode pengujian dan evaluasi.

d. BAB IV HASIL DAN PEMBAHASAN.

Bab ini menyajikan hasil implementasi sistem, hasil pengujian fungsionalitas, hasil pengujian integrasi dan keandalan, hasil evaluasi kualitas keluaran AI berbasis domain serta pembahasan terhadap temuan penelitian pada konteks operasional di BSI UII.

e. BAB V KESIMPULAN DAN SARAN.

Bab ini berisi kesimpulan dari hasil penelitian serta saran yang dapat digunakan untuk pengembangan sistem dan penelitian selanjutnya.

BAB II
LANDASAN TEORI

Bab ini membahas landasan teori yang mendasari penelitian, meliputi manajemen aset teknologi informasi, sistem pendukung keputusan, monitoring dan evaluasi aset, *Large Language Model*, pemanfaatan kecerdasan buatan dalam sistem operasional, sistem berbasis aturan, keandalan sistem dan penelitian terdahulu yang relevan. Uraian teori pada bab ini digunakan untuk menjelaskan dasar konseptual dari pengembangan Sistem Pendukung Keputusan Manajemen Aset TI Berbasis LLM dan untuk menunjukkan posisi penelitian terhadap studi-studi sebelumnya.

2.1 Manajemen Aset Teknologi Informasi

Manajemen Aset Teknologi Informasi atau *Information Technology Asset Management* (ITAM) merupakan proses pengelolaan aset teknologi secara sistematis sepanjang siklus hidupnya, mulai dari perencanaan, pengadaan, penggunaan, pemeliharaan, hingga penghapusan (Stone et al., 2018). Dalam organisasi modern, ITAM tidak lagi dipahami hanya sebagai aktivitas inventarisasi, tetapi sebagai bagian dari tata kelola yang memastikan setiap aset memberi kontribusi optimal terhadap tujuan operasional organisasi (Gavrikova et al., 2020; Maletič et al., 2020).

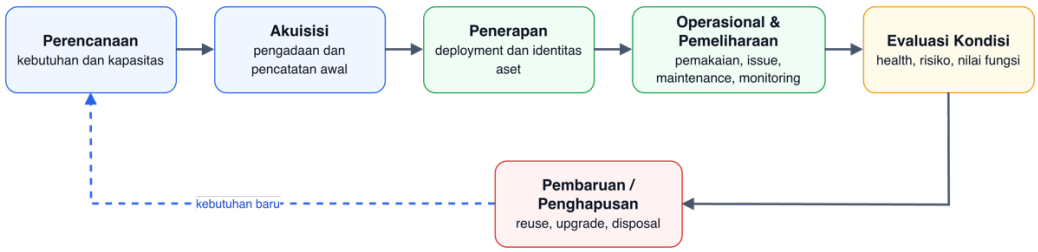
Pada lingkungan pendidikan tinggi, aset teknologi informasi memiliki peran penting karena menjadi penopang layanan akademik, administrasi, dan operasional digital. *Server*, perangkat jaringan inti, sistem penyimpanan, dan perangkat komputasi tertentu merupakan contoh aset yang memiliki dampak langsung terhadap kontinuitas layanan. Oleh sebab itu, pengelolaan aset harus mampu menjawab dua kebutuhan sekaligus, yaitu ketertiban administratif dan kesiapan operasional.

2.1.1 Konsep Dasar dan Siklus Hidup Aset

Pada dasarnya, siklus hidup aset mencakup tahap perencanaan kebutuhan, akuisisi, penerapan, pengoperasian, pemeliharaan, pembaruan, dan penghapusan (Stone et al., 2018). Pada tahap perencanaan, organisasi menentukan kebutuhan aset berdasarkan sasaran layanan dan kapasitas operasional. Tahap akuisisi berfokus pada pengadaan aset yang sesuai dengan kebutuhan tersebut. Setelah aset diterapkan, organisasi memasuki fase penggunaan dan pemeliharaan yang biasanya berlangsung paling lama dalam siklus hidup aset.

Tahap pengoperasian dan pemeliharaan menjadi fase yang sangat menentukan nilai guna aset. Pada fase inilah muncul berbagai informasi penting seperti lokasi aset, status penggunaan, riwayat peminjaman, riwayat isu, tindakan pemeliharaan, metrik performa, dan kecenderungan penurunan kondisi. Bila fase ini tidak terdokumentasi dengan baik, maka organisasi hanya memiliki catatan administratif awal, tetapi tidak memiliki visibilitas yang memadai terhadap kondisi aktual aset.

Pengelolaan aset yang baik menuntut adanya sumber informasi yang mampu menghubungkan data akuisisi dengan data operasional. Hal ini berati, aset tidak hanya tercatat sebagai barang yang dimiliki, tetapi juga sebagai sumber daya yang terus dievaluasi nilai fungsinya. Dalam konteks ini, ITAM memiliki posisi strategis karena berhubungan langsung dengan efisiensi operasional, keandalan layanan, dan pengambilan keputusan terkait pemeliharaan maupun penggantian aset (Gavrikova et al., 2020; Maletič et al., 2020). Visualisasi rangkaian tahap siklus hidup tersebut disajikan pada Gambar 2.1.



Gambar 2.1 Siklus hidup aset teknologi informasi

2.1.2 Kerangka Tata Kelola Aset ISO 55000

ISO (*International Organization for Standardization*) adalah badan standardisasi internasional yang menerbitkan standar di berbagai bidang, termasuk manajemen aset. Dalam konteks organisasi yang mengelola aset fisik secara serius, standar ISO 55000 sering digunakan sebagai rujukan konseptual tata kelola aset (Albalghouni et al., 2018). ISO 55000 menyediakan prinsip dan kosakata dasar tata kelola aset, sedangkan ISO 55001 menetapkan persyaratan formal yang digunakan sebagai basis sertifikasi sistem manajemen aset. Penelitian ini hanya mengadopsi ISO 55000 sebagai landasan konseptual, tanpa mengejar kepatuhan atau sertifikasi terhadap ISO 55001. Standar ini menempatkan aset bukan sekadar objek inventaris, melainkan bagian dari sistem nilai organisasi yang harus dikelola secara terarah, terukur, dan berkelanjutan. Pada konteks institusi pendidikan tinggi, studi kasus penerapan ISO 55000 menunjukkan bahwa adopsi standar ini menuntut perubahan signifikan pada struktur kebijakan,

sistem informasi, dan praktik pemeliharaan, sehingga implementasinya kerap tidak dapat dilakukan secara penuh dalam jangka pendek (Albalghouni et al., 2018).

Prinsip penting dalam ISO 55000 adalah penyelarasan antara tujuan organisasi dengan pengelolaan aset. Aset harus dikelola agar mampu mendukung misi institusi, bukan sekadar tercatat secara administratif. Standar ini juga menekankan pentingnya kepemimpinan, evaluasi risiko, pengendalian informasi aset, pengukuran kinerja, dan perbaikan berkelanjutan melalui siklus *plan-do-check-act* (Albalghouni et al., 2018).

Pada konteks penelitian ini, ISO 55000 digunakan sebagai kerangka konseptual untuk memahami bahwa pengelolaan aset perlu terhubung dengan nilai organisasi, risiko operasional, pengendalian informasi aset, dan pengukuran kinerja. Prinsip-prinsip yang relevan kemudian dipetakan secara terbatas ke desain sistem sebagaimana ditunjukkan pada Tabel 2.1.

Tabel 2.1 Prinsip tata kelola aset yang diadopsi dari kerangka ISO 55000

Prinsip ISO 55000	**Refleksi pada sistem**
Lifecycle asset management	Pencatatan isu, pemeliharaan, dan *health snapshot*
Value from assets	*Insight* operasional dan evaluasi status kesehatan untuk membantu prioritas tindakan
Risk-based decision	*Alert severity* dan *health evaluation* dalam penentuan prioritas
Information management	Histori aset, histori operasional, dan *snapshot* kondisi
Continual improvement	Rencana penguatan evaluasi melalui SOP evaluasi, pengukuran p50/p95, dan *limited-concurrency load test* lanjutan

2.1.3 Aset Strategis, CAPEX, OPEX, dan Nilai Organisasi

Dalam pengelolaan aset, penting untuk membedakan antara *Capital Expenditures* (CAPEX) dan *Operational Expenditures* (OPEX). CAPEX merujuk pada investasi organisasi terhadap aset yang bernilai tinggi dan memiliki umur pakai relatif panjang, seperti *server*, perangkat jaringan inti, dan infrastruktur pendukung pusat data. OPEX umumnya berkaitan dengan biaya operasional rutin, seperti langganan layanan, lisensi tertentu, atau kebutuhan pendukung yang berulang (Rahadi et al., 2024).

Perbedaan CAPEX dan OPEX bukan semata istilah akuntansi, tetapi juga berpengaruh pada prioritas pengelolaan. Aset CAPEX biasanya memiliki konsekuensi operasional yang lebih besar bila mengalami gangguan. Kerusakan pada *server* inti atau *edge router* dapat

menyebabkan gangguan layanan dalam skala luas, sedangkan gangguan pada aset *non*-kritis mungkin hanya berdampak lokal atau terbatas pada unit tertentu.

Dalam perspektif operasional, memelihara aset CAPEX berarti menjaga nilai fungsi aset agar tetap mendukung layanan organisasi. Pada tingkat konseptual, hal ini berkaitan dengan perlindungan nilai investasi dan efektivitas pemanfaatan aset. Dengan kata lain, organisasi tidak hanya berupaya memiliki aset, tetapi juga memastikan aset tersebut tetap memberi manfaat secara maksimal sepanjang masa pakainya.

Dalam praktik organisasi, aset TI perlu dipahami dalam dua sisi sekaligus yaitu sisi operasional dan sisi strategis. Pada sisi operasional, aset TI menjaga keberlangsungan layanan harian (misalnya kestabilan *server*, jaringan, dan layanan aplikasi). Pada sisi strategis, data dan informasi yang dihasilkan dari pengelolaan aset TI mendukung pengambilan keputusan terkait cadangan aset, perencanaan investasi, dan penganggaran jangka menengah.

Pemahaman ini menjadi landasan mengapa sistem yang dikembangkan dalam penelitian ini tidak hanya memperlakukan aset sebagai objek inventaris, tetapi juga mempertimbangkan dimensi kritisitas dan nilai strategis aset dalam fitur-fitur pendukung keputusan seperti evaluasi kesehatan dan rekomendasi alokasi. Secara lebih spesifik, perbedaan konsekuensi operasional yang melekat pada distinksi CAPEX dan OPEX menjadi salah satu landasan konseptual dalam menstratifikasi aset ke dalam kelas risiko. Aset bernilai strategis yang umumnya bersifat CAPEX (seperti *server* dan perangkat jaringan) cenderung memerlukan kontrol evaluasi yang lebih ketat karena konsekuensi kegagalannya luas, sedangkan aset operasional rutin dengan dampak gangguan lebih terbatas cenderung cukup dengan perlakuan evaluasi yang lebih ringan. Perbandingan karakteristik kedua kategori pengeluaran tersebut dirangkum pada Tabel 2.2.

Tabel 2.2 Perbandingan karakteristik CAPEX dan OPEX

Aspek	**CAPEX**	**OPEX**
Sifat pengeluaran	Investasi jangka panjang	Biaya operasional rutin
Contoh aset/biaya	*Server*, *router* inti, dan *data center*	Langganan, lisensi berkala, dan biaya dukungan
Umur pakai	Relatif panjang	Relatif pendek atau periodik
Dampak gangguan	Tinggi pada layanan inti	Umumnya lebih terbatas
Prioritas pemeliharaan	Tinggi, terutama untuk aset resiko tinggi	Disesuaikan dengan fungsi operasional

2.2 Sistem Pendukung Keputusan

Pengelolaan aset yang efektif tidak berhenti pada pencatatan kondisi, tetapi menuntut dukungan dalam mengambil keputusan atas data yang terkumpul. Di sinilah sistem pendukung keputusan menjadi relevan. Sistem Pendukung Keputusan *Decision Support Syste* (DSS) adalah sistem informasi berbasis komputer yang dirancang untuk membantu proses pengambilan keputusan, terutama pada situasi semi-terstruktur dan tidak terstruktur (Rashidi et al., 2018). DSS tidak menggantikan peran manusia sebagai pengambil keputusan, tetapi menyediakan informasi, model, dan mekanisme analisis yang membantu penilaian menjadi lebih cepat, konsisten serta berbasis data.

Pada umumnya, DSS digunakan ketika keputusan tidak cukup ditentukan oleh satu aturan sederhana. Pengambilan keputusan, seperti prioritas pemeliharaan aset, penentuan aset yang paling sesuai untuk dialokasikan kepada pengguna tertentu, atau evaluasi kondisi aset berdasarkan berbagai indikator, merupakan contoh kebutuhan yang sesuai untuk DSS.

2.2.1 Karakteristik Sistem Pendukung Keputusan

DSS memiliki beberapa karakteristik utama. Pertama, sistem ini mendukung proses analisis, bukan hanya penyimpanan data. Kedua, DSS biasanya mengolah kombinasi data historis, data aktual, dan model penilaian. Ketiga, DSS dirancang untuk meningkatkan kualitas keputusan tanpa menghilangkan peran pengambil keputusan (Rashidi et al., 2018).

Dibandingkan sistem informasi operasional biasa, DSS memiliki orientasi yang lebih kuat pada evaluasi, rekomendasi, dan alternatif keputusan. Dalam konteks manajemen aset, DSS membantu menjawab pertanyaan, seperti aset mana yang perlu diprioritaskan pemeliharaannya, aset mana yang paling sesuai untuk kebutuhan pengguna tertentu, dan kondisi aset mana yang memerlukan perhatian segera.

2.2.2 Komponen Utama Sistem Pendukung Keputusan

Secara garis besar, DSS dapat dibedakan ke dalam beberapa lapisan utama yang saling melengkapi, yaitu subsistem manajemen data, subsistem manajemen model, dan antarmuka pengguna (Rashidi et al., 2018). Subsistem manajemen data menyimpan serta mengelola data yang diperlukan dalam analisis. Subsistem manajemen model berisi logika, aturan, atau teknik analisis yang digunakan untuk menghasilkan rekomendasi. Antarmuka pengguna berfungsi menjembatani interaksi pengguna dengan sistem.

Pada sistem modern, komponen model dapat mencakup pendekatan berbasis aturan, analisis statistik, hingga kecerdasan buatan. Integrasi kecerdasan buatan pada DSS memungkinkan sistem memberikan rekomendasi yang lebih adaptif, terutama bila data yang diproses tidak seluruhnya berbentuk terstruktur. Tiga komponen utama DSS tersebut dirangkum pada Tabel 2.3.

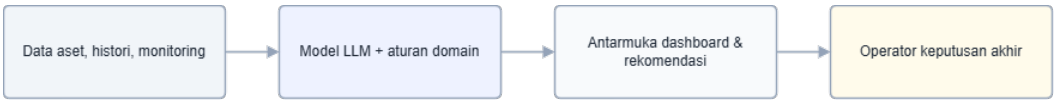
Tabel 2.3 Komponen utama sistem pendukung keputusan

Komponen	Fungsi utama
Manajemen data	Mengelola data historis, data aktual, dan data referensi
Manajemen model	Menyediakan logika analisis, aturan, dan mekanisme rekomendasi
Antarmuka pengguna	Menyajikan hasil analisis dan memfasilitasi interaksi pengguna

2.2.3 Sistem Pendukung Keputusan dalam Konteks Manajemen Aset

Dalam manajemen aset, DSS dapat digunakan untuk membantu penentuan prioritas, rekomendasi, dan evaluasi kondisi. Misalnya, sistem dapat membantu menentukan aset yang paling relevan untuk dialokasikan kepada pegawai berdasarkan kebutuhan peran kerja. Sistem juga dapat membantu menampilkan *insight* operasional dari kombinasi data aset, data pemeliharaan, dan data monitoring. Selain itu, DSS dapat mendukung evaluasi kesehatan aset dengan memperhatikan data historis, kondisi operasional, dan aturan bisnis organisasi.

Oleh karena itu, DSS menjadi relevan dalam penelitian ini karena fitur AI yang dikembangkan pada dasarnya berfungsi sebagai pendukung keputusan. Sistem tidak mengambil alih otoritas manusia sepenuhnya, tetapi membantu menyajikan dasar pertimbangan yang lebih cepat, lebih terstruktur, dan lebih mudah ditelusuri. Hubungan antara data, model analisis, dan antarmuka dalam kerangka DSS ini divisualisasikan pada Gambar 2.2.



Gambar 2.2 Hubungan data, model, dan antarmuka pada sistem pendukung keputusan

2.3 Monitoring dan Evaluasi Aset

Agar dukungan keputusan berpijak pada kondisi aktual aset, sistem memerlukan aliran data kondisi yang berkelanjutan. Monitoring aset merupakan proses pengumpulan dan

pengamatan data kondisi aset secara berkala untuk memastikan bahwa aset tetap berada pada kondisi operasional yang diharapkan (Teixeira et al., 2020). Dalam aset teknologi informasi, monitoring biasanya berkaitan dengan metrik performa, status konektivitas, beban kerja, log kejadian, dan tanda-tanda penurunan kinerja.

Monitoring menjadi penting karena catatan administratif tidak cukup untuk menjelaskan kondisi aktual aset. Sebuah aset dapat masih tercatat aktif, tetapi secara operasional sudah menunjukkan gejala penurunan performa. Oleh karena itu, monitoring menyediakan lapisan informasi yang melengkapi data master aset dan riwayat operasional. Alur dari pemantauan kondisi hingga tindak lanjut pemeliharaan ini dirangkum pada Gambar 2.3.

2.3.1 *Condition-Based Monitoring*

Salah satu pendekatan yang banyak digunakan adalah *Condition-Based Monitoring* (CBM), yaitu pemantauan aset berdasarkan kondisi aktual yang diukur melalui indikator tertentu (Teixeira et al., 2020). Berbeda dengan pemeliharaan berbasis jadwal semata, CBM menekankan bahwa keputusan pemeliharaan sebaiknya mempertimbangkan data nyata dari kondisi aset sehingga tindakan pemeliharaan dapat dilakukan pada waktu yang lebih tepat dan proporsional terhadap kondisi aktual.

Dalam aset teknologi informasi, indikator yang dapat dipantau mencakup penggunaan CPU, penggunaan memori, pemakaian *disk*, kondisi jaringan, suhu, status layanan tertentu dan riwayat kondisi (Zhaoxue et al., 2021a). Data tersebut tidak selalu bermakna secara langsung bila berdiri sendiri. Namun, ketika dikaitkan dengan histori aset, tingkat kritisitas, dan catatan isu sebelumnya, data tersebut dapat membantu organisasi mengenali pola penurunan kondisi lebih dini.

2.3.2 Pendekatan Pemeliharaan

Dalam praktik, pemeliharaan dapat dibedakan menjadi pemeliharaan reaktif, preventif, dan prediktif (Teixeira et al., 2020). Pemeliharaan reaktif dilakukan setelah terjadi gangguan. Pemeliharaan preventif dilakukan secara berkala untuk mencegah gangguan. Pemeliharaan prediktif dilakukan dengan memanfaatkan data kondisi aktual untuk memperkirakan potensi gangguan sebelum benar-benar terjadi.

Dalam sistem pengelolaan aset yang lebih matang, monitoring berfungsi sebagai landasan menuju pendekatan prediktif. Organisasi dapat bergerak dari pola kerja yang menunggu kerusakan menjadi pola kerja yang mampu mengenali tanda-tanda risiko lebih awal.

Perubahan ini penting terutama untuk aset bernilai tinggi dan berdampak besar terhadap layanan.

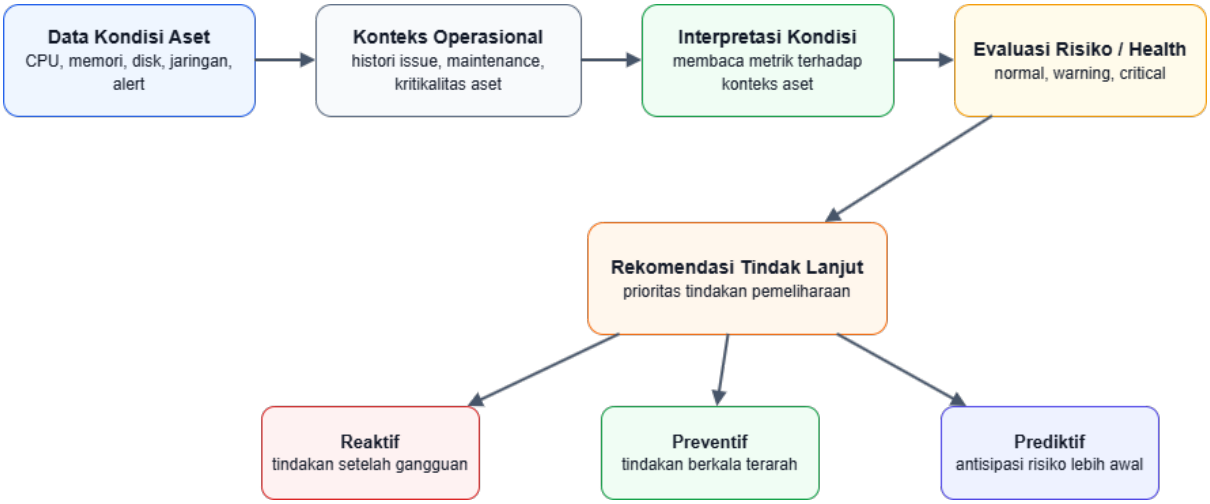
2.3.3 Metrik Monitoring dan Interpretasi Operasional

Data monitoring yang baik harus dapat diterjemahkan ke dalam konteks operasional. Nilai penggunaan CPU yang tinggi, misalnya, belum tentu berarti gangguan jika sesuai dengan beban kerja normal. Sebaliknya, lonjakan penggunaan memori atau *disk* yang berlangsung terus-menerus dapat menjadi indikasi perlunya evaluasi lebih lanjut.

Karena itu, monitoring tidak cukup hanya mengumpulkan data. Sistem perlu memiliki kerangka interpretasi agar metrik teknis dapat dikaitkan dengan kondisi aset dan kebutuhan operasional. Pada titik inilah integrasi monitoring dengan ITAM dan DSS menjadi penting. Contoh metrik monitoring dan interpretasi operasionalnya disajikan pada Tabel 2.4.

Tabel 2.4 Contoh metrik monitoring dan interpretasi umum

Matriks	**Indikasi umum**	**Potensi tindak lanjut**
CPU *usage* tinggi terus-menerus	Beban kerja tinggi atau proses tidak efisien	Evaluasi kapasitas dan proses
Memory usage tinggi	Risiko *bottleneck* atau kebocoran memori	Pemeriksaan layanan atau aplikasi
Disk usage mendekati penuh	Risiko gangguan penyimpanan	Pembersihan atau ekspansi
Network usage abnormal	Lalu lintas tidak normal atau *bottleneck*	Pemeriksaan konektivitas
Riwayat *alert* berulang	Potensi masalah berulang pada aset	Prioritas evaluasi teknis



Gambar 2.3 Alur monitoring aset menuju tindakan pemeliharaan.

Dalam penelitian ini, monitoring tidak diperlakukan sebagai sistem terpisah, tetapi sebagai sumber konteks bagi AI *Asset Insight* dan AI *Health Evaluation*.

2.4 Large Language Model

Large Language Model (LLM) adalah model kecerdasan buatan berbasis arsitektur *Transformer* yang dilatih pada data teks dalam skala sangat besar (Vaswani et al., 2023). Model ini dirancang untuk memahami konteks bahasa alami dan menghasilkan keluaran berbasis teks yang relevan terhadap masukan yang diberikan. Berbeda dengan model klasifikasi tradisional yang biasanya dirancang untuk tugas sempit, LLM memiliki fleksibilitas yang tinggi karena dapat digunakan pada berbagai tugas melalui instruksi dan konteks.

Kemampuan LLM menjadi menarik dalam sistem informasi karena banyak data operasional organisasi tidak seluruhnya berbentuk numerik atau terstruktur. Catatan isu, deskripsi aset, riwayat pemeliharaan, hasil observasi teknisi, dan ringkasan kondisi merupakan contoh data yang lebih dekat dengan bahasa alami. Dalam konteks ini, LLM dapat dimanfaatkan untuk membantu menafsirkan, merangkum, dan menyajikan informasi secara lebih mudah dipahami.

2.4.1 Gambaran Umum Cara Kerja LLM

LLM memproses masukan dalam bentuk token dan memanfaatkan konteks sebelumnya untuk menghasilkan keluaran berikutnya. Arsitektur *Transformer* memungkinkan model memperhatikan hubungan antarkata atau antarpotongan informasi dalam suatu konteks

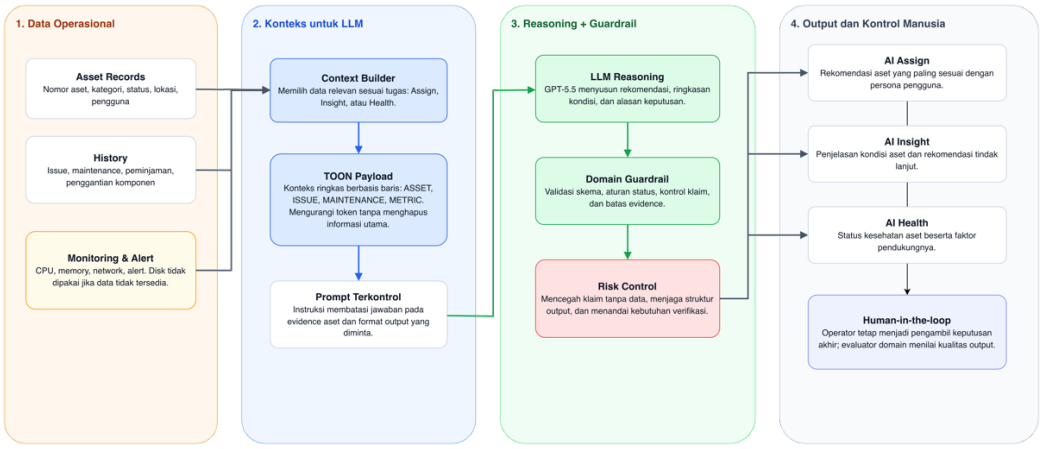
(Vaswani et al., 2023). Oleh karena itu, model dapat menghasilkan respons yang tampak koheren dan sesuai dengan pola instruksi yang diberikan.

Dalam praktiknya, kualitas hasil LLM sangat dipengaruhi oleh kualitas konteks dan kejelasan instruksi. Model tidak memahami dunia, seperti manusia, tetapi memprediksi hasil yang paling mungkin berdasarkan pola yang dipelajarinya. Oleh sebab itu, konteks yang baik dan terstruktur akan sangat memengaruhi ketepatan hasil.

2.4.2 Kemampuan LLM dalam Sistem Informasi

LLM memiliki beberapa kemampuan yang relevan untuk sistem informasi operasional. Pertama, model dapat melakukan ringkasan dari data teks yang panjang atau tersebar. Kedua, model dapat membantu menyusun narasi analitis berdasarkan informasi yang diberikan. Ketiga, model dapat membantu menghasilkan rekomendasi awal jika diberikan konteks domain yang tepat (Durachman et al., 2026).

Dalam konteks manajemen aset, kemampuan ini berguna untuk mengubah data yang tersebar menjadi informasi yang lebih mudah dipahami pengguna. Sebagai contoh, data spesifikasi aset, riwayat isu, dan ringkasan monitoring dapat diolah menjadi *insight* operasional yang lebih ringkas dan lebih mudah digunakan sebagai bahan pertimbangan. Gambaran umum alur pemanfaatan LLM pada data operasional disajikan pada Gambar 2.4.



Gambar 2.4 Proses umum pemanfaatan LLM pada data operasional

2.4.3 Keterbatasan LLM

Walaupun memiliki potensi besar, LLM juga memiliki keterbatasan yang perlu dipahami. Salah satu keterbatasan yang paling sering dibahas adalah *hallucination* (halusinasi

AI), yaitu kecenderungan model menghasilkan informasi yang terdengar meyakinkan, tetapi tidak didukung oleh data yang benar (Huang et al., 2025; Zhou et al., 2024). Selain itu, LLM dapat menghasilkan respons yang tidak konsisten apabila konteks, formulasi instruksi, atau kondisi pemanggilan model berubah.

Keterbatasan lain berkaitan dengan auditabilitas. Proses penalaran model tidak selalu mudah dijelaskan secara rinci (Huang et al., 2025). Dalam sistem operasional yang menuntut akuntabilitas tinggi, hal tersebut menjadi tantangan tersendiri. Oleh karena itu, pemanfaatan LLM tidak dapat dilepaskan begitu saja tanpa pembatasan domain, validasi, dan kontrol tambahan.

2.5 Pemanfaatan LLM dalam Sistem Operasional

Setelah memahami cara kerja LLM secara umum, perlu ditinjau bagaimana kemampuan tersebut diterapkan pada konteks operasional teknologi informasi. Pemanfaatan kecerdasan buatan dalam sistem operasional sering dikaitkan dengan konsep *Artificial Intelligence for IT Operations* (AIOps), yaitu pendekatan yang memanfaatkan analitik data, pembelajaran mesin, dan otomasi untuk mendukung deteksi, analisis serta respons terhadap kondisi operasional TI (Kumar et al., 2024). AIOps berkembang sebagai respons terhadap semakin besarnya volume data operasional yang harus dikelola organisasi.

Hasilnya LLM dapat berkontribusi dalam konteks ini karena mampu membantu memproses data operasional yang tidak sepenuhnya terstruktur. Alih-alih hanya menampilkan data mentah, sistem dapat memanfaatkan model untuk menyusun *insight*, memberi rekomendasi, atau menjelaskan kondisi aset dalam bentuk yang lebih komunikatif.

2.5.1 LLM sebagai Pendukung *Insight* Operasional

Dalam sistem operasional, salah satu nilai tambah LLM adalah kemampuannya mengubah potongan data teknis menjadi narasi yang lebih mudah dipahami. Pengguna tidak selalu membutuhkan deretan angka atau log secara mentah. Sering kali pengguna justru membutuhkan ringkasan yang menjelaskan kondisi aset, potensi masalah, dan alasan mengapa kondisi tersebut perlu diperhatikan.

Dengan demikian, LLM dapat berperan sebagai lapisan interpretasi yang menghubungkan data teknis dengan kebutuhan pemahaman pengguna. Nilai ini menjadi penting dalam sistem manajemen aset karena pengguna sistem tidak selalu memiliki latar belakang teknis yang sama.

2.5.2 LLM untuk Rekomendasi Operasional

Selain *insight*, LLM juga dapat mendukung pemberian rekomendasi operasional. Dalam konteks pengelolaan aset, rekomendasi tersebut dapat berupa saran alokasi aset berdasarkan profil kebutuhan atau penjelasan alasan mengapa suatu aset perlu diperhatikan lebih dahulu. Namun, rekomendasi yang dihasilkan model sebaiknya dipandang sebagai bahan pertimbangan, bukan keputusan final yang sepenuhnya otonom.

Agar rekomendasi tetap relevan, model harus dibatasi pada domain yang sesuai. Data yang dimasukkan ke model harus dipilih secara hati-hati dan konteks harus disusun dalam format yang efisien serta terarah. Dalam literatur, pendekatan ini sering dikaitkan dengan *structured prompting* (White et al., 2023) dan *context engineering* yang mencakup pemilihan, integrasi, serta kompresi konteks (Mei et al., 2025). Oleh karena itu, pemanfaatan LLM dalam sistem operasional perlu dibatasi pada konteks yang relevan dan dikombinasikan dengan mekanisme validasi untuk menjaga konsistensi dan keandalan hasil.

2.5.3 Pembatasan Domain dan Efisiensi Konteks

Salah satu tantangan dalam penggunaan LLM adalah keterbatasan konteks dan biaya pemanggilan model. Karena itu, tidak semua data perlu dikirim secara mentah. Sistem yang baik justru harus mampu memilih bagian informasi yang paling relevan, menyusunnya dalam format yang ringkas, dan tetap menjaga makna pentingnya. (Mei et al., 2025) menyebut disiplin ini sebagai *context engineering*, yaitu pekerjaan sistematis untuk mengoptimasi muatan informasi (*information payload*) yang melampaui sekadar penulisan *prompt*. *Context engineering* mencakup pemilihan konteks, pemrosesan konteks yang mencakup integrasi informasi terstruktur, serta pengelolaan konteks yang mencakup kompresi.

Kompresi konteks penting dalam sistem operasional karena memungkinkan data aset, histori, dan monitoring dirangkum secara efisien sebelum diproses model. Selain meningkatkan efisiensi, kompresi membantu menjaga fokus model pada atribut yang relevan terhadap tugas yang sedang dijalankan. Survei (Mei et al., 2025) mencatat bahwa *structured context* dan *context compression* merupakan dua komponen yang saling melengkapi pada *context engineering modern*, di mana penyusunan informasi dalam representasi terstruktur sekaligus dapat menurunkan konsumsi token tanpa kehilangan atribut keputusan.

Token-Oriented Object Notation (TOON) adalah format representasi data *open-source* yang dirancang untuk efisiensi token pada *prompt* LLM (Schopplich, 2025). Pada penelitian ini, TOON diadopsi sebagai format konteks terstruktur untuk domain manajemen aset TI.

Format ini menyusun atribut domain dalam unit konsisten, seperti objek, *field*, dan tabel ringkas, agar atribut penting tetap eksplisit, urutan informasi stabil, dan konsumsi token lebih terkontrol. Secara operasional, TOON menggunakan penamaan bagian yang eksplisit, urutan atribut yang tetap, serta representasi tabular ringkas untuk data berulang seperti metrik, histori *issue*, dan histori *maintenanc*e. Dengan cara ini, informasi domain tetap dapat dibaca oleh model tanpa membawa seluruh redundansi sintaksis *JavaScript Object Notation* (JSON) mentah.

Perbedaan utama TOON dengan *JavaScript Object Notation* (JSON), *YAML Ain't Markup Language* (YAML), atau *Tom's Obvious, Minimal Language* (TOML) terletak pada tujuan desainnya. JSON, YAML, dan TOML merupakan format pertukaran data atau konfigurasi umum, sedangkan TOON diadopsi pada penelitian ini sebagai format penyusunan konteks untuk prompt LLM pada domain ITAM. Karena itu, TOON tidak dianggap sebagai pengganti format pertukaran data antarsistem, melainkan sebagai lapisan representasi konteks yang dibentuk setelah data operasional dipilih dan dinormalisasi. Berdasarkan kerangka (Mei et al., 2025), TOON dapat dikelompokkan ke dalam kategori *structured context* yang sekaligus berfungsi sebagai mekanisme *context compression* domain-spesifik untuk ITAM. Implikasi, TOON bukan sekadar preferensi format teknis, melainkan implementasi konsep *context engineering* yang memiliki landasan teoritis. TOON merupakan format *open-source* yang sudah mapan dengan spesifikasi publik dan implementasi multibahasa (Schopplich, 2025) kontribusi penelitian ini terletak pada penerapan TOON serta perancangan skema konteks domain ITAM di atasnya, bukan pada perancangan format TOON itu sendiri. Efek kompresi TOON diukur secara empiris melalui perbandingan token antara JSON mentah dan TOON.

2.5.4 *Structured Prompting* dan *Context Engineering*

Prompt engineering dalam konteks operasional tidak hanya berarti menulis instruksi, tetapi juga merancang cara konteks disajikan supaya model menghasilkan hasil yang konsisten. (White et al., 2023) menunjukkan bahwa pola *prompt*, seperti pola instruksi eksplisit, pola format keluaran, dan pola pembatasan ruang jawaban yang dapat meningkatkan kualitas serta keterkendalian keluaran model. Studi LLM untuk dukungan keputusan juga menegaskan bahwa kualitas hasil tidak hanya bergantung pada model, tetapi juga pada desain interaksi, konteks tugas, dan cara penjelasan disampaikan kepada pengguna keputusan (Durachman et al., 2026; Lazaros et al., 2026). Prinsip ini menjadi dasar bahwa kualitas jawaban LLM dipengaruhi kuat oleh kualitas struktur konteks, bukan hanya oleh model yang digunakan.

Secara konsepsual, *context engineering* pada penelitian ini dilakukan melalui tiga langkah yaitu pemilihan atribut domain yang relevan, normalisasi format konteks (TOON), dan validasi struktur keluaran agar dapat diproses oleh sistem. Pendekatan ini sejalan dengan kebutuhan sistem pendukung keputusan yang menuntut hasil dapat ditelusuri dan digunakan ulang antarmodul.

Penting untuk ditegaskan bahwa penelitian ini tidak mengimplementasikan *Retrieval-Augmented Generation* (RAG). Konteks LLM dibangun sepenuhnya dari data operasional internal yang sudah terstruktur melalui format TOON sehingga posisi penelitian berada pada spektrum *structured prompting* atau *context engineering*.

2.6 Sistem Berbasis Aturan

Fleksibilitas LLM dalam menafsirkan data perlu diimbangi mekanisme yang menjamin kepatuhan pada aturan domain. Pendekatan berbasis aturan menawarkan sisi komplementer tersebut. Sistem berbasis aturan (*rule-based system*) merupakan pendekatan yang menggunakan aturan formal, biasanya dalam bentuk logika *if-then*, untuk menghasilkan keputusan atau kesimpulan (Reddy & Fields, 2022). Pendekatan ini telah lama digunakan pada berbagai sistem karena bersifat jelas, konsisten, dan mudah ditelusuri. Sistem berbasis aturan menjadi salah satu fondasi dalam pengembangan sistem pakar dan sistem pendukung keputusan yang membutuhkan transparansi dan keterjelasan logika.

Dalam sistem operasional, *rule-based system* berguna ketika organisasi memiliki aturan yang cukup stabil dan dapat dijelaskan secara eksplisit. Misalnya, organisasi dapat memiliki aturan bahwa *alert* tertentu harus menaikkan prioritas evaluasi atau kondisi tertentu pada metrik aset harus dikategorikan sebagai risiko tinggi.

2.6.1 Kelebihan Sistem Berbasis Aturan

Kelebihan utama sistem berbasis aturan adalah konsistensi dan keterlacakan logika. Sistem yang dibangun dari aturan eksplisit umumnya bersifat *robust*, dapat diskalakan, dan cepat dijalankan (Reddy & Fields, 2022). Selain itu, *rule-based system* mudah dijelaskan kepada pengguna karena logika keputusannya berasal dari aturan yang didefinisikan secara eksplisit.

Pada domain manajemen aset, transparansi ini penting. Pengguna perlu memahami mengapa suatu aset diberi status tertentu atau mengapa suatu tindakan direkomendasikan.

Dalam kondisi ini, *rule-based system* memberikan fondasi yang kuat untuk kontrol dan konsistensi.

2.6.2 Keterbatasan Sistem Berbasis Aturan

Meskipun memiliki kelebihan, sistem berbasis aturan juga memiliki keterbatasan fundamental ketika dihadapkan pada situasi yang tidak tercakup dalam aturan yang tersedia (*rule gap*), situasi ketika beberapa aturan berlaku sekaligus dengan konsekuensi yang berpotensi bertentangan (*rule conflict*), atau ketika masukan berupa data naratif yang tidak terstruktur, seperti catatan isu dan deskripsi kondisi aset dalam bahasa alami. Pada kondisi tersebut, sistem berbasis aturan cenderung tidak mampu memberikan interpretasi yang memadai apabila aturan eksplisit belum didefinisikan sebelumnya. Selain itu, semakin kompleks domain yang ditangani, semakin besar pula basis aturan yang perlu dipelihara sehingga beban pemeliharaan aturan dapat meningkat seiring bertambahnya variasi aset dan skenario operasional (Reddy & Fields, 2022). Untuk mengatasi keterbatasan tersebut, literatur mencatat munculnya pendekatan alternatif, seperti *fuzzy logic*, *neural network*, *database system*, *case-based reasoning*, hingga pendekatan berorientasi objek yang dapat diintegrasikan dengan *rule-based* (Reddy & Fields, 2022).

Oleh karena itu, *rule-based system* sangat kuat untuk kontrol, tetapi tidak selalu cukup untuk interpretasi konteks yang kompleks. Di sinilah integrasi dengan kecerdasan buatan menjadi relevan.

Dalam penelitian ini, pendekatan berbasis aturan digunakan terutama sebagai *guardrail* untuk memastikan status kesehatan akhir tetap sesuai dengan referensi domain.

2.7 Integrasi Pendekatan Berbasis Aturan dan Kecerdasan Buatan

Integrasi antara pendekatan berbasis aturan dan kecerdasan buatan bertujuan untuk menggabungkan kelebihan kedua paradigma. Sistem berbasis aturan memberi kontrol, konsistensi, dan keterjelasan logika. Kecerdasan buatan memberi fleksibilitas dalam menafsirkan data yang lebih kompleks dan tidak sepenuhnya terstruktur.

Pendekatan integratif ini relevan dalam domain yang menuntut dua hal sekaligus, yaitu kemampuan analisis yang adaptif dan jaminan bahwa keputusan penting tetap dapat dikendalikan. Dalam konteks aset teknologi informasi, kebutuhan tersebut sangat nyata, terutama ketika sistem menangani aset kritis yang berdampak besar terhadap operasional.

2.7.1 *Human-in-the-Loop*

Salah satu konsep penting dalam integrasi AI dan aturan adalah *human-in-the-loop* (HITL), yaitu pendekatan yang mempertahankan keterlibatan manusia dalam proses evaluasi dan pengambilan Keputusan (Lazaros et al., 2026). Dalam pendekatan ini, teknologi digunakan untuk membantu menyusun analisis, rekomendasi, atau penjelasan, sementara manusia tetap memiliki peran dalam melakukan peninjauan, interpretasi, dan pengambilan keputusan akhir.

Keterlibatan manusia pada HITL bukanlah pilihan biner antara *manual approval* setiap keluaran dan *full automation* tanpa pengawasan, melainkan suatu spektrum yang dapat dikalibrasi sesuai risiko dan dampak operasional setiap keputusan. Pada ujung yang lebih ketat, setiap keluaran AI perlu divalidasi langsung oleh operator sebelum dieksekusi, seperti pada keputusan yang berdampak terhadap aset kritis atau alokasi yang sulit dianulir. Pada ujung yang lebih longgar, sistem dapat menjalankan evaluasi secara otomatis berdasarkan kombinasi model dan aturan yang telah ditetapkan, dengan peran manusia difokuskan pada audit, pemantauan, dan evaluasi hasil sistem secara berkala. Posisi pada spektrum ini ditentukan oleh kombinasi antara tingkat risiko keputusan, ketersediaan *guardrail* berbasis aturan, dan kebutuhan auditabilitas pada domain terkait (Lazaros et al., 2026).

Dengan kerangka spektrum tersebut, sistem dapat tetap efisien dalam menjalankan proses bervolume tinggi yang berisiko rendah, sekaligus menjaga akuntabilitas pada keputusan berisiko tinggi melalui validasi langsung oleh pengguna. Pada penelitian ini, posisi HITL dikalibrasi pada sisi yang lebih ketat untuk keluaran AI yang berdampak operasional sehingga rekomendasi alokasi, *insight*, dan evaluasi kesehatan aset berperan sebagai dukungan keputusan, bukan eksekusi otomatis.

2.7.2 *Guardrail* dan Validasi

Jika HITL menempatkan manusia sebagai lapis kendali akhir, *guardrail* berperan sebagai lapis kendali otomatis yang bekerja lebih dulu di dalam sistem. Keduanya bersifat komplementer *guardrail* menyaring dan membatasi keluaran model secara deterministik, sedangkan HITL memastikan keputusan berdampak tetap ditinjau manusia. Agar kecerdasan buatan tetap berada dalam jalur yang aman, sistem dapat menerapkan *guardrail*. Secara konseptual, *guardrail* merupakan mekanisme pembatas yang menjaga supaya hasil model tetap sesuai dengan batasan domain, format, dan aturan organisasi (Dong et al., 2024). Bentuknya dapat berupa validasi terhadap struktur keluaran, pembatasan konteks, hingga pemeriksaan hasil terhadap aturan bisnis tertentu.

Gagasan integrasi AI dan aturan penting karena sistem tidak hanya dituntut memberikan *insight*, tetapi juga harus dapat dipertanggungjawabkan. Oleh karena itu, kecerdasan buatan berfungsi sebagai pendukung keputusan yang bekerja dalam batasan tertentu, bukan sebagai pengganti logika organisasi secara penuh.

2.8 *Framework* Evaluasi pada Sistem Pendukung Keputusan Berbasis AI

Pada sistem pendukung keputusan berbasis AI, evaluasi tidak cukup dinilai hanya dari akurasi model, seperti pada tugas klasifikasi murni. Fokus evaluasi perlu diarahkan pada tingkat kesesuaian keluaran AI terhadap kebutuhan keputusan operasional pengguna karena keluaran AI digunakan sebagai bahan pertimbangan, bukan keputusan otomatis final. Dalam konteks ini, pendekatan *human-in-the-loop evaluation* menjadi kerangka yang relevan (Durachman et al., 2026; Lazaros et al., 2026) .

Salah satu bentuk evaluasi yang praktis pada domain operasional Adalah *expert judgment and agreement*, yaitu membandingkan keluaran AI dengan penilaian operator atau ahli domain sebagai representasi *grounded operational judgment* (Yusof & Abdul Rahim, 2024). Pendekatan ini berbeda dari konsep *ground truth* tunggal pada *supervised learning* karena keputusan operasional sering memuat konteks kebijakan, toleransi risiko, dan pertimbangan lapangan yang tidak selalu terwakili oleh satu label absolut.

Untuk memperkuat objektivitas, evaluasi keluaran AI dapat melibatkan lebih dari satu penilai. Namun, pada evaluasi berbasis domain, pemilihan ukuran kesepakatan perlu disesuaikan dengan desain penilaian. Apabila penilai memiliki domain keahlian berbeda dan tidak semua penilai valid untuk menilai seluruh item, evaluasi lebih tepat dibaca melalui kesepakatan deskriptif per pasangan evaluator, distribusi skor, dan analisis *disagreement*.

Pada penelitian ini, evaluasi kualitas keluaran AI dilakukan menggunakan pendekatan *domain-based expert agreement evaluation*, yaitu penilaian oleh operator BSI UII berdasarkan domain operasional yang relevan. Pasangan penilai ditentukan berdasarkan relevansi domain AMP dan NDC untuk fitur terkait aset *monitorable*, AMP, dan Inventaris untuk fitur non-monitoring dan rekomendasi alokasi aset. Evaluasi diarahkan pada pembacaan kesepakatan pasangan penilai melalui *percent agreement*, distribusi skor, dan analisis naratif terhadap perbedaan penilaian. Kerangka ini menjadi dasar metodologis bagi BAB III dalam menetapkan evaluasi kualitas keluaran AI sebagai validasi operasional terbatas, bukan sebagai generalisasi reliabilitas populasi.

Selain evaluasi kesesuaian keluaran oleh evaluator domain, sistem pendukung keputusan juga memerlukan evaluasi dari sisi persepsi pengguna terhadap manfaat operasional sistem. Berbeda dengan *expert agreement* yang menilai kesesuaian keluaran AI terhadap konteks domain, evaluasi persepsi pengguna diarahkan pada dimensi pengalaman penggunaan, seperti efisiensi, keterpahaman, relevansi rekomendasi, kejelasan alasan, prioritas tindakan, dan kepercayaan terhadap prinsip *human-in-the-loop*. Instrumen yang umum digunakan untuk evaluasi persepsi semacam ini adalah skala Likert, yaitu skala sumatif berskala ordinal yang meminta responden menyatakan tingkat persetujuan terhadap pernyataan tertentu pada rentang diskrit tertentu (Joshi et al., 2015). Pada konteks evaluasi DSS, skala Likert 1–5 umum dipakai karena memberi granularitas cukup untuk membedakan persepsi negatif, netral, dan positif tanpa membebani responden dengan terlalu banyak pilihan, serta sesuai untuk jumlah responden terbatas pada studi kasus operasional (Joshi et al., 2015). Skala Likert berposisi sebagai instrumen pelengkap ia mengukur persepsi penggunaan, bukan kualitas teknis keluaran AI. Oleh karena itu, hasil skor Likert tidak dimaksudkan sebagai pengganti *expert agreement* maupun metrik konsistensi keluaran, melainkan sebagai dimensi evaluasi tambahan yang melaporkan sisi penggunaan sistem dari sudut pandang operator. Karakteristik ordinal skala Likert juga menyiratkan bahwa, pada jumlah responden terbatas, skor diperlakukan sebagai indikasi persepsi awal, bukan sebagai dasar inferensi statistik populasi. Desain operasional instrumen Likert pada penelitian ini diuraikan pada BAB III, dan status pelaksanaannya dilaporkan pada BAB IV.

2.8.1 Evaluasi Konsistensi Keluaran *Large Language Model*

Selain kesesuaian keluaran terhadap penilaian domain, dimensi evaluasi yang relevan untuk sistem pendukung keputusan berbasis LLM adalah konsistensi keluaran pada input identik. Berbeda dengan sistem berbasis aturan yang bersifat deterministik dan menghasilkan keluaran identik untuk input yang sama, LLM bersifat probabilistik sehingga dapat menghasilkan keluaran yang bervariasi meski *prompt* dan konteks input tidak berubah (Lin et al., 2024). Variasi tersebut bukan kegagalan teknis, melainkan konsekuensi dari mekanisme *sampling* pada arsitektur model. Namun, pada konteks sistem pendukung keputusan operasional, variasi keluaran perlu diukur agar operator memperoleh dasar keputusan yang dapat ditelusuri dan diaudit antareksekusi.

Metrik konsistensi perlu disesuaikan dengan bentuk keluaran yang dihasilkan. Untuk keluaran berupa daftar identifier (urutan rekomendasi aset), konsistensi dapat diukur melalui

Overlap Coefficient pada himpunan elemen teratas. Untuk keluaran berupa status diskrit (*HEALTHY*/*WARNING*/*CRITICAL*), konsistensi diukur melalui proporsi status dominan antareksekusi. Untuk keluaran berupa narasi teks bebas (*insight* kondisi aset), konsistensi diukur melalui kemiripan teks antareksekusi.

Salah satu metrik kemiripan teks yang relevan untuk narasi LLM Adalah *Jaccard Similarity*. Metrik ini mengukur kemiripan dua himpunan sebagai rasio antara jumlah elemen irisan terhadap jumlah elemen gabungan, dengan rentang nilai 0 - 1, dimana 1 menandakan dua himpunan identik dan 0 menandakan tidak ada kesamaan. Pada konteks narasi, teks keluaran dipecah menjadi himpunan kata setelah *preprocessing* sederhana (*lowercasing*, penghapusan *stopword*, dan normalisasi istilah teknis), kemudian dibandingkan antareksekusi (Jalal et al., 2022). *Jaccard Similarity* dipilih untuk narasi *insight* karena metrik ini beroperasi pada himpunan kata tanpa memperhitungkan urutan, sehingga tahan terhadap variasi susunan kalimat yang wajar pada keluaran LLM. Selain itu, metrik ini tidak memerlukan representasi vektor (*embedding*) seperti *cosine similarity* dan tidak memerlukan kalimat referensi *ground truth* seperti BLEU, sehingga sesuai untuk mengukur variasi leksikal antareksekusi pada input identik.

Meskipun demikian, *Jaccard Similarity* memiliki keterbatasan yang perlu diperhitungkan dalam interpretasi hasil. Metrik ini mengukur kemiripan leksikal (kesamaan kata), bukan kemiripan semantik (kesamaan makna). Akibatnya, dua narasi yang menyampaikan makna sama dengan pilihan kata berbeda misalnya "CPU 92%" dan "penggunaan prosesor sangat tinggi" akan dianggap berbeda meskipun arah rekomendasi yang disampaikan setara. Oleh karena itu, nilai *Jaccard* yang rendah pada narasi LLM tidak secara otomatis menunjukkan inkonsistensi substansi. Variasi tersebut dapat bersumber dari perbedaan frasa dalam merumuskan observasi yang sebenarnya konsisten. Dalam kerangka evaluasi penelitian ini, konsistensi keluaran diperlakukan sebagai bukti tambahan yang melengkapi penilaian domain, bukan sebagai pengganti penilaian operator.

2.9 Keandalan Sistem dan Ketahanan Layanan

Kualitas keluaran hanya bermakna apabila sistem yang menghasilkannya tetap tersedia saat dibutuhkan. Karena fitur AI bergantung pada layanan eksternal, keandalan menjadi dimensi yang harus dirancang secara eksplisit. Sistem yang terintegrasi dengan layanan eksternal, terutama layanan model berbasis *Application Programming Interface* (API), menghadapi tantangan keandalan yang berbeda dibandingkan sistem tertutup. Gangguan

jaringan, pembatasan laju permintaan, latensi yang berubah-ubah, dan kegagalan sementara dapat memengaruhi konsistensi layanan. Dalam konteks sistem operasional, masalah ini penting karena pengguna membutuhkan layanan yang tetap stabil walaupun komponen eksternal tidak selalu ideal. Literatur terbaru reliabilitas LLM juga menekankan pentingnya definisi profil operasional dan ketahanan terhadap ketidakpastian ketika model dipakai pada lingkungan nyata (Lin et al., 2024).

2.9.1 Keandalan dalam Sistem Terdistribusi

Pada sistem terdistribusi, keandalan tidak hanya berarti sistem dapat berjalan, tetapi juga berarti sistem dapat tetap memberikan layanan yang layak ketika sebagian komponen mengalami gangguan. Hal ini berkaitan dengan konsep toleransi kesalahan (*fault tolerance*) dan *graceful degradation*, yaitu kemampuan sistem untuk tetap berfungsi dalam kondisi terdegradasi tanpa langsung gagal total (Punithavathy & Priya, 2024).

Dalam sistem yang memanfaatkan LLM, keandalan menjadi semakin penting karena kualitas layanan bergantung pada komponen yang tidak sepenuhnya berada di bawah kendali organisasi. Oleh sebab itu, desain keandalan perlu menjadi bagian dari perancangan sistem.

2.9.2 Pola Ketahanan Layanan

Beberapa pola umum yang sering digunakan untuk meningkatkan ketahanan layanan adalah *retry*, *exponential backoff*, dan *fallback* (Punithavathy & Priya, 2024). *Retry* digunakan untuk mencoba kembali permintaan yang gagal, terutama bila kegagalan bersifat sementara. *Exponential backoff* menambahkan jeda yang meningkat supaya sistem tidak terus membebani layanan yang sedang bermasalah. *Fallback* menyediakan alternatif respons ketika layanan utama tidak berhasil.

Pola *circuit breaker* yang berfungsi menghentikan sementara permintaan ke layanan tidak stabil, diakui dalam literatur sebagai mekanisme ketahanan tambahan (Punithavathy & Priya, 2024). Namun, pola ini belum diimplementasikan pada sistem yang dikembangkan dalam penelitian ini dan direkomendasikan sebagai pengembangan lanjutan pada BAB V.

Caching juga dapat digunakan untuk meningkatkan efisiensi dan mengurangi kebutuhan pemanggilan berulang terhadap layanan eksternal. Dalam konteks sistem operasional, pola-pola ini penting bukan hanya dari sisi teknis, tetapi juga dari sisi kualitas layanan yang diterima pengguna.

Tabel 2.5 Pola umum ketahanan layanan

Pola	Fungsi umum	Manfaat
Retry	Mengulang permintaan yang gagal	Mengatasi kegagalan sementara
Exponential backoff	Memberi jeda bertahap sebelum pengulangan	Mengurangi tekanan pada layanan
Fallback	Memberi respons alternatif	Mencegah kegagalan total
Circuit breaker	Menghentikan sementara permintaan ke layanan bermasalah	Menekan kegagalan berantai
Caching	Menyimpan hasil yang sering dipakai	Meningkatkan efisiensi dan kecepatan

Tabel 2.5, memuat pola ketahanan layanan secara konseptual. Pada sistem penelitian ini, pola yang diterapkan adalah *retry*, *exponential backoff*, dan *fallback*, sedangkan *circuit breaker* belum diimplementasikan dan diperlakukan sebagai pola konseptual. Adapun *caching* telah diimplementasikan sebagai bentuk perbaikan artefak pada iterasi DSR.

Pada sistem yang dikembangkan dalam penelitian ini, pola *retry* dan *exponential backoff* diterapkan pada komunikasi *backend* ke LangChain AI *service* untuk menangani kegagalan sementara. Sementara itu, pola *fallback* dirancang untuk menjaga ketersediaan fitur *insight* dan evaluasi status kesehatan (*health evaluation*) saat layanan AI mengalami gangguan. Pada alur *insight*, *fallback* digunakan agar layanan tetap dapat merespons ketika *endpoint* mode baru belum tersedia pada alur evaluasi status kesehatan, *fallback* okal digunakan supaya *pipeline* evaluasi tetap berjalan ketika pemanggilan model gagal.

2.10 Tinjauan Pustaka

Tinjauan pustaka dilakukan untuk memetakan posisi penelitian ini terhadap studi-studi sebelumnya yang relevan. Dalam konteks ini, penelitian terdahulu yang berkaitan dengan sistem aset dapat dibagi ke dalam tiga kelompok besar antara lain digitalisasi pencatatan administratif, monitoring dan evaluasi kondisi aset serta pemanfaatan kecerdasan buatan pada pengelolaan aset dan operasi TI.

Pada konteks pendidikan tinggi di Indonesia, banyak penelitian sistem aset masih berpusat pada pengelolaan data administratif, pencatatan inventaris, atau pelaporan. Sistem seperti ini penting sebagai fondasi, tetapi umumnya belum mengintegrasikan data monitoring kondisi aktual aset dan belum mengarah pada sistem pendukung keputusan. Akibatnya, sistem lebih kuat dalam fungsi pencatatan dibandingkan dalam fungsi analisis operasional.

Sebaliknya, perkembangan di sektor industri menunjukkan arah yang berbeda. Studi-studi yang lebih baru mulai memanfaatkan kecerdasan buatan untuk membantu evaluasi kondisi aset, penyusunan *insight*, atau penyajian rekomendasi berbasis data operasional. Namun, penelitian pada sektor industri sering berangkat dari asumsi bahwa data lebih terstruktur, integrasi sistem lebih matang, dan lingkungan implementasi lebih terstandarisasi daripada yang biasa dijumpai pada institusi pendidikan tinggi (Zhou et al., 2024).

2.10.1 Komparasi Penelitian Terdahulu

Literatur pada tinjauan pustaka ini dipilih berdasarkan relevansi terhadap tiga pilar penelitian yaitu manajemen aset TI, pemanfaatan LLM pada operasi TI, dan evaluasi sistem pendukung keputusan berbasis AI. Pencarian dilakukan melalui basis data Scopus, IEEE Xplore, dan Google Scholar dengan rentang waktu utama 2019–2026, dilengkapi beberapa referensi klasik yang bersifat fundamental pada bidang manajemen aset dan sistem pendukung keputusan.

Tabel 2.6, menyajikan sintesis komparatif dari berbagai penelitian terdahulu. Komparasi ini bertujuan secara spesifik untuk mengidentifikasi celah riset (*research gap*) yang menjadi pijakan kebaruan penelitian ini, khususnya dalam menjembatani kebutuhan antara pencatatan administratif murni dengan pemanfaatan AI yang memerlukan kontrol deterministik.

Tabel 2.6 Komparasi penelitian terdahulu

No.	Penelitian	Fokus utama	Kelebihan	Keterbatasan	Relevansi terhadap penelitian ini
1	(Maletič et al., 2020)	Praktik inti manajemen aset fisik dan pengaruhnya ke kinerja operasional	Menunjukkan hubungan kuat antara disiplin manajemen aset dan performa operasi	Fokus pada kerangka praktik umum, belum membahas integrasi LLM	Menjadi dasar teoritis bahwa kualitas pengelolaan aset memengaruhi kinerja operasional sistem
2	(Zhou et al., 2024)	Pemanfaatan LLM untuk penyusunan solusi pada industrial *asset management*	Memberikan bukti awal bahwa LLM dapat membantu interpretasi domain aset industri	Konteks industri dan data readiness berbeda dari institusi pendidikan tinggi	Menjadi rujukan untuk desain fitur *insight* operasional berbasis konteks aset

No.	Penelitian	Fokus utama	Kelebihan	Keterbatasan	Relevansi terhadap penelitian ini
3	(Mei et al., 2025)	Survei *Context Engineering* untuk LLM: kompresi konteks, *structured context*, dan *context management* (review 1.400+ paper)	Memformalkan disiplin *context engineering* sebagai optimasi sistematis muatan informasi LLM, mencakup kompresi struktural dan integrasi informasi terstruktur	Bersifat survei lintas domain, belum memuat realisasi spesifik untuk format aset TI	Menjadi landasan teoretis untuk memposisikan TOON sebagai realisasi *structured context* dan *context compression* pada domain ITAM
4	(White et al., 2023)	Katalog pola *prompt engineering* untuk meningkatkan kualitas keluaran LLM	Menyediakan pola praktis untuk instruksi, struktur konteks, dan format keluaran	Bersifat umum lintas domain, perlu adaptasi ke domain aset	Menjadi landasan metodologis untuk *structured prompting* dan *context engineering* (termasuk TOON)
5	(Rahadi et al., 2024)	Tinjauan praktik manajemen aset pada BUMN Indonesia	Menegaskan pentingnya tata kelola aset sebagai isu strategis organisasi	Analisis berada pada tataran tata kelola makro dan belum menjabarkan desain sistem operasional harian	Menjadi pijakan konseptual untuk memprioritaskan aset bernilai investasi tinggi (bernuansa CAPEX) dan menerjemahkannya ke kontrol operasional pada sistem yang dikembangkan
6	(Zhaoxue et al., 2021b)	Survei metode AIOps berbasis log untuk deteksi anomali, korelasi kejadian, dan dukungan operasi TI	Memberikan peta metode yang sistematis untuk analisis *event*/*alert* pada operasi TI	Fokus pada domain operasi TI umum dan belum spesifik ke integrasi histori aset	Relevan sebagai dasar konseptual integrasi monitoring-*alert* dengan fitur AI aset
7	(Setiawan et al., 2019)	Perancangan sistem informasi manajemen aset pada	Memberikan baseline digitalisasi inventarisasi dan	Fokus pada pencatatan administratif (CRUD aset + perhitungan	Menjadi *baseline* representatif AMIS pendidikan tinggi Indonesia. Posisi penelitian ini mengisi

No.	Penelitian	Fokus utama	Kelebihan	Keterbatasan	Relevansi terhadap penelitian ini
		institusi pendidikan tinggi	administrasi aset kampus	depresiasi) belum mengintegrasikan data monitoring kondisi aktual, belum memuat layanan AI sebagai pendukung keputusan, dan belum memiliki *guardrail* berbasis aturan untuk auditabilitas keluaran	tiga aksis gap: (i) integrasi data monitoring real-time, (ii) layanan AI sebagai pendukung keputusan, dan (iii) *guardrail* deterministik untuk auditabilitas
8	(Salamun et al., 2022)	Sistem pendukung keputusan manajemen aset Universitas Abdurrab berbasis ISO 55000 dengan metode TOPSIS	Menyajikan SPK manajemen aset pendidikan tinggi yang lebih maju dari sekadar pencatatan, mengangkat ISO 55000 dan menerapkan MCDM klasik	Pendekatan SPK bersifat deterministik (TOPSIS) sehingga tidak adaptif terhadap konteks naratif aset belum mengintegrasikan data monitoring atau layanan AI berbasis LLM	Memperkuat *baseline* paper terbaru di Indonesia: penelitian SPK aset kampus terbaru pun masih bertumpu pada paradigma deterministik (MCDM). Penelitian ini melanjutkan dengan paradigma adaptif berbasis LLM yang dikendalikan *guardrail* aturan
9	(Notaro et al., 2021)	Survei AIOps untuk *failure* management pada sistem terdistribusi	Mengelompokkan metode AIOps berdasarkan jendela intervensi dan target kegagalan	Tidak spesifik pada domain manajemen aset kampus	Menguatkan landasan integrasi monitoring, deteksi gangguan, dan respons operasional bertahap
10	(Durachman et al., 2026)	Pertanyaan riset baru LLM untuk *Decision Support System*	Memberi kerangka konseptual tahap keputusan (*intelligence-design-choice*) pada LLM-DSS	Bersifat konseptual dan lintas domain, belum memberikan desain artefak operasional rinci	Menjadi landasan positioning bahwa LLM dipakai sebagai dukungan keputusan, bukan pengambil keputusan otomatis
11	(Lazaros et al., 2026)	*Human-in-the-Loop AI, A Systematic*	*Systematic review* HITL (*loop*	Bersifat survei lintas domain, belum spesifik ITAM	Memperkuat kerangka HITL sebagai spektrum

No.	Penelitian	Fokus utama	Kelebihan	Keterbatasan	Relevansi terhadap penelitian ini
		Review of Concepts, Methods, and Applications	*placement, interaction granularity, taxonomy*) untuk aplikasi *safety-critical*		keterlibatan manusia pada evaluasi keluaran AI
12	(Lin et al., 2024)	Kerangka reliabilitas LLM berbasis probabilistik hierarkis	Menawarkan pendekatan reliabilitas yang memperhitungkan profil operasional dan ketidakpastian	Tidak dirancang spesifik untuk alur bisnis aset pascaakuisisi	Menguatkan perspektif keandalan pada integrasi layanan AI eksternal berbasis API

2.10.2 Analisis *Gap* dan *Positioning* Penelitian

Berdasarkan komparasi di atas, terlihat bahwa penelitian lokal pada domain sistem aset masih cenderung menempatkan aset sebagai objek administratif. Penelitian-penelitian tersebut berkontribusi dalam perbaikan pencatatan, tetapi belum banyak menjawab kebutuhan pengelolaan aset sebagai sumber daya operasional yang harus dipantau dan dievaluasi secara berkelanjutan.

Sementara itu, penelitian yang lebih maju mulai menunjukkan bahwa kecerdasan buatan, termasuk LLM, dapat digunakan untuk membantu interpretasi data operasional dan menghasilkan *insight* yang lebih adaptif. Kajian literatur terbaru juga menegaskan empat aspek penting yang relevan untuk penelitian ini, yaitu desain LLM-DSS operasional, evaluasi *human-in-the-loop* berbasis kesepakatan ahli, ketahanan layanan AI berbasis API eksternal, dan praktik AIOps untuk manajemen kegagalan (Durachman et al., 2026; Lazaros et al., 2026; Lin et al., 2024) . Namun, banyak penelitian tersebut berada pada konteks industri dengan asumsi kesiapan data yang lebih baik.

Berdasarkan landasan teori dan tinjauan pustaka, dapat diidentifikasi tiga celah penelitian. Pertama, banyak sistem aset masih berorientasi pada pencatatan administratif dan belum mengintegrasikan data kondisi aktual aset. Kedua, pemanfaatan AI dalam pengelolaan aset belum banyak diterapkan dalam konteks institusi pendidikan tinggi, khususnya dengan mempertimbangkan keterbatasan data dan kebutuhan auditabilitas. Ketiga, penelitian terkait LLM pada manajemen aset lebih banyak dilakukan pada konteks industri dan belum banyak

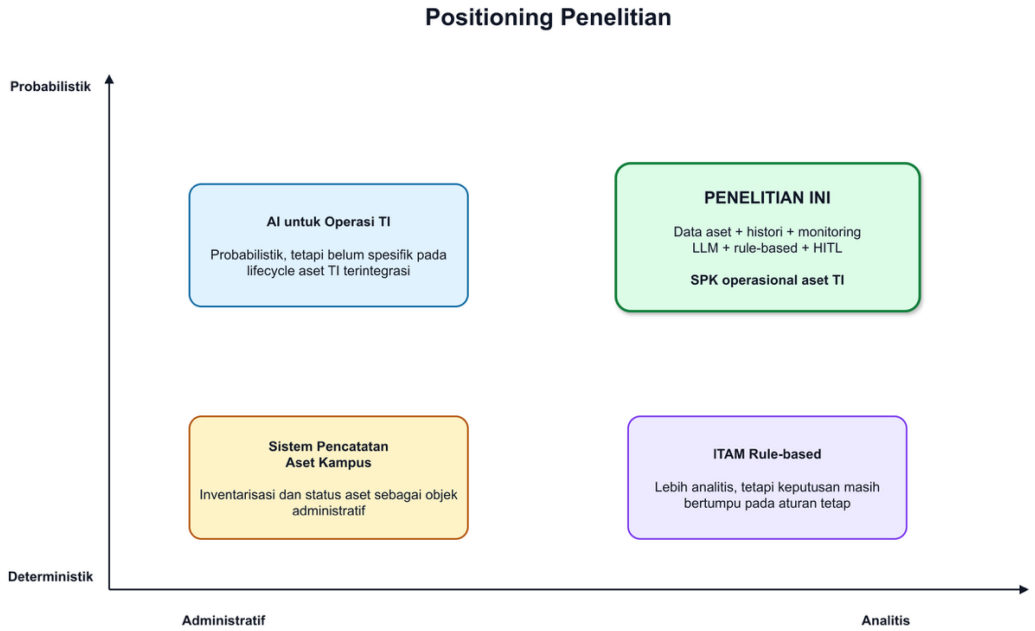
menyoroti integrasi data aset inti, histori operasional, dan data monitoring dalam satu alur analisis yang terpadu.

Penelitian ini mengisi celah tersebut melalui pengembangkan Sistem Pendukung Keputusan Manajemen Aset TI Berbasis LLM yang menghubungkan tiga lapisan utama. Lapisan pertama adalah pengelolaan aset inti sebagai fondasi administratif dan operasional. Lapisan kedua adalah data monitoring dan histori operasional sebagai sumber evaluasi kondisi aset. Lapisan ketiga adalah layanan AI dan evaluasi internal yang membantu rekomendasi alokasi aset, penyusunan *insight* operasional, serta penilaian kesehatan aset secara lebih adaptif dan terkendali. Kecerdasan buatan ditempatkan dalam batasan yang terkontrol sebagai pendukung keputusan, bukan sebagai pengambil keputusan tunggal.

Pada level desain, pendekatan penelitian ini menggunakan prinsip ISO 55000 sebagai inspirasi konseptual (Albalghouni et al., 2018) pada beberapa aspek utama. Pertama, penyelarasan tujuan organisasi dengan pengelolaan aset diwujudkan melalui fokus sistem pada dukungan keputusan operasional yang berdampak langsung pada kontinuitas layanan BSI UII. Kedua, pengendalian informasi aset diwujudkan melalui integrasi data aset, histori operasional, dan monitoring dalam alur yang lebih terstruktur serta dapat ditelusuri. Ketiga, pengukuran kinerja operasional diwujudkan melalui indikator hasil pengujian seperti ketercapaian skenario bisnis, kualitas keluaran AI, dan keandalan integrasi layanan. Hal ini menunjukkan keselarasan prinsip desain yang diadopsi secara terbatas, bukan klaim kepatuhan (*compliance*) atau sertifikasi formal ISO.

2.10.3 Diagram Positioning Penelitian

Untuk memperjelas posisi penelitian secara visual, digunakan pemetaan dua sumbu yaitu sumbu horizontal dari orientasi administratif ke orientasi analitis, dan sumbu vertikal dari pendekatan deterministik ke pendekatan probabilistik**.**



Gambar 2.5 *Positioning* penelitian

Pada kuadran kiri-bawah, sistem aset kampus tradisional menonjol pada fungsi inventarisasi administratif dengan logika yang dominan deterministik. Kuadran kanan-bawah merepresentasikan sistem ITAM berbasis *rule engine* yang sudah lebih analitis, tetapi masih mengandalkan aturan tetap. Posisi penelitian ini berada pada kuadran kanan-atas karena menggabungkan kemampuan analitis LLM yang bersifat probabilistik dengan *guardrail* berbasis aturan untuk menjaga kontrol keputusan operasional.

Dari perspektif metodologi sistem informasi, posisi penelitian ini juga sejalan dengan paradigma *Design Science Research* (DSR), yaitu pendekatan yang menekankan pembangunan dan evaluasi artefak untuk menyelesaikan masalah praktis organisasi (Peffers et al., 2007). Relevansi DSR pada penelitian ini terlihat dari orientasi artefak (sistem terintegrasi), evaluasi multiaspek (fungsional, integrasi, keandalan, dan dukungan keputusan) serta komunikasi hasil dalam bentuk naskah ilmiah. Pemetaan posisi penelitian terhadap studi terdahulu pada spektrum orientasi dan pendekatan ini telah divisualisasikan pada Gambar 2.5. Sebagai penutup tinjauan pustaka, Tabel 2.7 merangkum keterkaitan setiap landasan teori dengan wujud implementasinya pada sistem.

Tabel 2.7 Pemetaan teori ke implementasi sistem

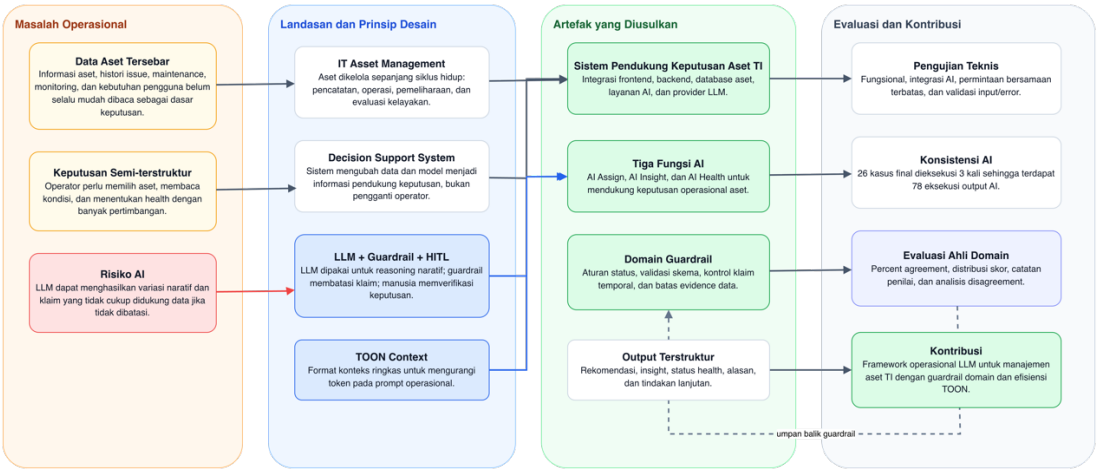
Teori / konsep	Subbab teori	Wujud pada sistem	Bukti pada BAB IV
ITAM *lifecycle* pascaakuisisi	2.1.1	Modul peminjaman atau pengembalian, pencatatan isu, pemeliharaan, dan *health snapshot*	Tabel 4.6
ISO 55000 dan keputusan berbasis risiko	2.1.2	Stratifikasi aset *high-risk* dan l*ow-risk* pada AI *Health Evaluation*	Subbab 4.7.11
Decision Support System	2.2.2	Kombinasi data, model analisis, dan antarmuka untuk mendukung keputusan operator	Tabel 4.1 dan Tabel 4.7
Condition-Based Monitoring	2.3.1	Metrik monitoring dan *alert* digunakan sebagai konteks AI *Insight* dan AI *Health*	Subbab 4.2.5
LLM dan *structured prompting*	2.5.4	*Prompt* terstruktur, konteks TOON, dan validasi *output* untuk tiga fitur AI	Subbab 4.2.2-4.2.4
TOON sebagai *context compression*	2.5.3	Representasi *payload* domain ITAM yang lebih ringkas dibanding JSON	Tabel 4.16
Rule-based system	2.6.1	*Guardrail* pada penetapan status kesehatan akhir, terutama pada aset berisiko tinggi	Subbab 4.8
Human-in-the-loop	2.7.1	*Output* AI bertindak sebagai rekomendasi dan *insight*, bukan keputusan otomatis	Tabel 4.7
Pola keandalan layanan	2.9.2	*Retry*, *backoff*, *fallback*, *throttling*, dan *caching* pada integrasi AI	Tabel 4.9
Evaluasi berbasis domain	2.8	Penilaian *output* AI oleh pasangan *evaluator* sesuai relevansi domain	Tabel 4.20

2.11 Kerangka Pemikiran

Kerangka pemikiran penelitian disusun untuk menunjukkan alur logis dari masalah awal menuju dampak operasional. Penelitian ini berangkat dari persoalan data aset dan data operasional yang tersebar, lalu menawarkan solusi terintegrasi melalui tiga komponen sistem, menghasilkan keluaran analitis, dan menargetkan dampak langsung pada kualitas operasi BSI UII.

Secara lebih rinci, alur kerangka pemikiran dapat dijelaskan sebagai berikut. Titik awal penelitian adalah permasalahan operasional berupa data aset yang tersebar pada beberapa media pencatatan, seperti SAP, *spreadsheet*, dan catatan manua sehingga evaluasi kondisi aset masih bergantung pada proses manual yang memakan waktu dan belum terstandarisasi. Dari permasalahan tersebut, penelitian ini mengembangkan solusi berupa sistem terintegrasi yang

menghubungkan tiga komponen utama antara lain pengelolaan data aset inti, integrasi data monitoring dan histori operasional, serta layanan AI berbasis LLM yang dikontrol oleh *guardrail* berbasis aturan. Hasil sistem berupa *insight* operasional, rekomendasi alokasi aset, dan evaluasi kesehatan aset yang lebih terstruktur dan dapat ditelusuri. Dampak yang ditargetkan adalah percepatan proses triase kondisi aset, standarisasi dasar pengambilan keputusan operasional, dan peningkatan visibilitas terhadap kondisi aktual aset di BSI UII. Visualisasi keseluruhan alur kerangka pemikiran ini disajikan pada Gambar 2.6.



Gambar 2.6 Kerangka pemikiran penelitian

Berdasarkan kerangka tersebut, kontribusi penelitian dapat dipahami sebagai penghubung antara digitalisasi administratif dan dukungan keputusan operasional berbasis data. Keluaran sistem tidak berhenti pada informasi deskriptif, tetapi diarahkan menjadi masukan keputusan yang lebih terstruktur, lebih mudah ditelusuri, dan mendukung pembacaan kondisi operasional pada konteks BSI UII.

Landasan teori dan *positioning* penelitian pada BAB II ini menjadi dasar bagi penyusunan metodologi pada BAB III, terutama dalam perancangan artefak berbasis *Design Science Research*, penyusunan konteks TOON, pengujian keandalan layanan AI, dan evaluasi kualitas keluaran AI melalui pendekatan *human-in-the-loop*.

BAB III
METODOLOGI PENELITIAN

Bab ini menjelaskan metode yang digunakan dalam penelitian untuk merancang, membangun, dan mengevaluasi Sistem Pendukung Keputusan Manajemen Aset TI Berbasis LLM pada lingkungan Badan Sistem Informasi Universitas Islam Indonesia. Pembahasan mencakup jenis dan pendekatan penelitian, definisi operasional, lokasi dan ruang lingkup penelitian, sumber data, tahapan pengembangan sistem, perancangan arsitektur dan data, metode AI yang diterapkan, metode pengujian, indikator keberhasilan, etika penelitian, serta keterbatasan penelitian.

3.1 Jenis dan Pendekatan Penelitian

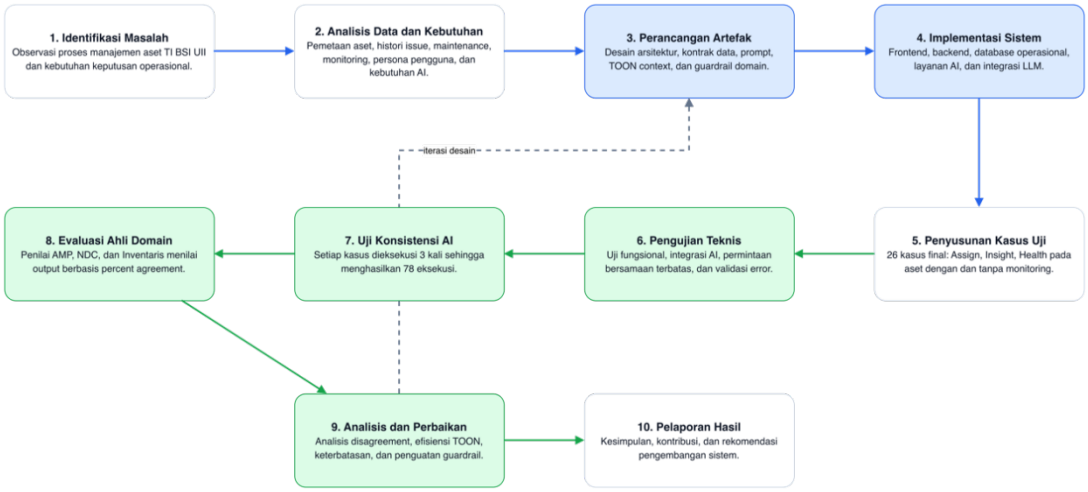
Penelitian ini menggunakan pendekatan *Design Science Research* (DSR) (Peffers et al., 2007) yang berfokus pada perancangan, pengembangan, demonstrasi, dan evaluasi artefak sistem untuk menjawab permasalahan praktis pada pengelolaan aset TI pascaakuisisi di BSI UII. Pendekatan ini dipilih karena keluaran utama penelitian adalah artefak sistem yang dapat diimplementasikan dan dievaluasi pada konteks operasional nyata.

Secara operasional, pendekatan DSR ini diwujudkan melalui aktivitas *design and development*, yaitu analisis kebutuhan, perancangan artefak, implementasi sistem, pengujian, dan evaluasi. Istilah *design and development* ini tidak digunakan sebagai metodologi terpisah, melainkan sebagai tahap operasional dalam kerangka DSR. Alur ini digunakan agar desain solusi tidak berhenti pada konseptualisasi, tetapi berlanjut pada pembuktian fungsi dan kelayakan operasional artefak.

Pendekatan ini sesuai dengan karakter penelitian yang menempatkan sistem sebagai hasil utama penelitian. Akibatnya, kontribusi penelitian tidak hanya terletak pada penjelasan konseptual, tetapi juga pada kemampuan artefak dalam mendukung proses operasional pengelolaan aset TI secara lebih terintegrasi, adaptif, dan andal. Pemetaan tahap DSR pada penelitian ini dirangkum pada Tabel 3.1, sedangkan alur umum penelitian divisualisasikan pada Gambar 3.1.

Tabel 3.1 Pemetaan tahap *Design Science Research* pada penelitian

Tahap DSR	Implementasi pada penelitian
Problem identification	Data aset tersebar, evaluasi kondisi masih manual, dan status *health* belum terstruktur
Objective	Merancang sistem ITAM *pascaakuisisi* berbasis LLM + *rule-based* untuk dukungan keputusan operasional
Design & development	Pengembangan fr*ontend, backend,* basis data, LangChain AI *service,* serta integrasi monitoring
Demonstration	Demonstrasi skenario modul inti dan tiga fitur AI (*Assign, Insight, Health*)
Evaluation	Evaluasi fungsional, integrasi, keandalan, skenario proses bisnis, dan umpan balik DSS operator
Communication	Penyusunan skripsi serta dokumentasi rekam jejak iterasi artefak hasil evaluasi



Gambar 3.1 Alur penelitian

3.2 Lokasi, Subjek, dan Ruang Lingkup Penelitian

Penelitian ini dilaksanakan pada Badan Sistem Informasi Universitas Islam Indonesia (BSI UII) sebagai lokasi studi kasus. Pemilihan lokasi dilakukan karena BSI UII merupakan unit yang memiliki tanggung jawab langsung terhadap pengelolaan infrastruktur dan layanan sistem informasi di lingkungan universitas. Selain itu, BSI UII mengelola aset TI dalam jumlah besar dan memiliki kebutuhan operasional yang relevan dengan tujuan penelitian, khususnya pada aspek integrasi data aset, pencatatan operasional, pemantauan kondisi aset, dan dukungan keputusan berbasis data.

Subjek penelitian ini adalah proses operasional pengelolaan aset TI yang berlangsung di BSI UII, termasuk proses pencatatan aset, peminjaman dan pengembalian, pencatatan isu, pemeliharaan, pemantauan kondisi, serta kebutuhan analisis untuk mendukung keputusan operasional. Pihak yang terlibat sebagai sumber informasi operasional meliputi operator atau personel yang berhubungan langsung dengan pengelolaan aset dan data operasional terkait.

Ruang lingkup penelitian dibatasi pada modul inti pengelolaan aset, modul dukungan keputusan berbasis AI, serta integrasi data monitoring dan *alert*. Modul inti mencakup masterdata, aset, peminjaman dan pengembalian, pencatatan isu, serta pemeliharaan. Modul AI mencakup AI *Assign Recommendation*, AI *Asset Insight*, dan *pipeline* AI *Health Evaluation*. Integrasi monitoring digunakan sebagai konteks analisis kondisi aset, tanpa bermaksud membangun ulang infrastruktur monitoring organisasi. SAP universitas berperan sebagai sistem akuisisi dan pencatatan awal asset, sistem yang dikembangkan berfokus pada proses pascaakuisisi di BSI UII. Detail sumber data SAP dan data monitoring dijelaskan pada Subbab 3.3.

Penelitian ini juga memiliki batasan operasional. Pengujian dilakukan dalam konteks *single* operator pada lingkungan studi kasus. Pengembangan *Role-Based Access Control* granular belum menjadi fokus utama penelitian. Demikian pula, dashboard analitik tingkat lanjut dan arsitektur *multitenant* tidak dibahas secara mendalam.

Istilah *single* operator pada konteks pengujian sistem merujuk pada pelaksana utama skenario operasional. Adapun evaluasi kualitas keluaran AI menggunakan pendekatan *domain-based expert evaluation* dengan tiga penilai berlatar domain berbeda, tetapi tidak semua penilai menilai seluruh fitur karena cakupan domain operasionalnya berbeda. Ringkasan ruang lingkup penelitian disajikan pada Tabel 3.2.

Tabel 3.2 Ruang lingkup penelitian

Komponen	**Cakupan**
Lokasi studi kasus	Badan Sistem Informasi UII
Subjek penelitian	Proses operasional pengelolaan aset TI
Modul inti	Masterdata, aset, pinjam - kembali, isu, pemeliharaan
Modul AI	*Assign recommendation, asset insight,* dan *pipeline health evaluation* internal
Integrasi	Metrik dan *alert* pemantauan, referensi data akuisisi awal SAP melalui impor manual berkas Excel satu arah, serta data pemantauan aset melalui impor manual berkas CSV

Komponen	Cakupan
Di luar cakupan	RBAC granular lanjut, *multitenant, dashboard enterprise final*

3.3 Sumber Data dan Teknik Pengumpulan Data

Data yang digunakan dalam penelitian ini terdiri atas data primer dan data sekunder. Data primer diperoleh melalui observasi terhadap proses operasional pengelolaan aset di lingkungan BSI UII serta wawancara singkat dengan pihak operasional yang berkaitan dengan penggunaan dan pengelolaan data aset. Data primer digunakan terutama untuk memahami alur kerja, kebutuhan pengguna, masalah yang dihadapi, serta aturan bisnis yang perlu diakomodasi dalam sistem.

Data sekunder diperoleh dari sumber data operasional yang telah tersedia pada lingkungan studi kasus. Data ini mencakup data aset, data peminjaman dan pengembalian, data isu, data pemeliharaan, serta data metrik dan *alert* dari sistem pemantauan yang digunakan organisasi. Dalam konteks akuisisi awal, data SAP digunakan secara terbatas sebagai referensi baseline aset melalui input manual berkas Excel yang diberikan oleh Ketua Tim AMP BSI UII, tanpa integrasi transaksi dua arah. Data peminjaman dan pengembalian dimaknai sebagai konteks operasional untuk membaca status serta ketersediaan aset, bukan sebagai fokus analisis mandiri yang terpisah. Data sekunder berfungsi sebagai dasar penyusunan model data, pengembangan fitur, simulasi skenario operasional, serta pengujian fungsi sistem.

Untuk transparansi, Tabel 3.3 berikut merinci kategori dan sumber data yang digunakan pada penelitian ini. Dataset evaluasi akhir menggunakan pendekatan *scenario-based synthetic dataset anchored on real NDC traces*. Data NDC digunakan sebagai jejak operasional nyata untuk kondisi normal, sedangkan data sintetis digunakan secara terkontrol untuk membentuk skenario uji yang tidak tersedia pada data asli, seperti CPU *spike*, CPU *critical* berkelanjutan, *memory warning*, anomali jaringan, dan *alert* sintetis sinkron metrik. Rincian keterbatasan data *disk* dijelaskan pada Tabel 3.3.

Tabel 3.3 Kategori dan sumber data penelitian

Kategori data	Sumber	Keterangan	Kegunaan
Data inventaris aset	Data operasional nyata (BSI UII)	Diperoleh dari Kepala Tim AMP melalui berkas Excel *baseline*	Penyusunan model data dan modul inti

Kategori data	Sumber	Keterangan	Kegunaan
Data peminjaman, isu, pemeliharaan	Data operasional nyata (BSI UII)	Diperoleh dari catatan operasional unit	Konteks status/ketersediaan aset dan pengembangan alur operasional
Data metrik monitoring	Data operasional nyata (BSI UII)	Diperoleh dari *export* CSV Grafana melalui staff NDC sebagai *real anchor* data	Konteks AI *Insight* dan AI *Health*
Synthetic metric injection	Data sintetis terkontrol	Digunakan untuk CPU *spike*, CPU *critical* berkelanjutan, *memory warning*, dan anomali jaringan	Pembentukan skenario uji yang tidak tersedia pada data operasional nyata
Synthetic alert	Data sintetis terkontrol	Digunakan karena akses *Alertmanager live* tidak tersedia, *alert* disinkronkan dengan *synthetic metric injection*	Pembentukan skenario uji yang tidak tersedia pada data operasional nyata
Disk placeholder	Placeholder kontrak JSON	Data *disk* NDC tidak tersedia, nilai *disk* diisi 0 dan tidak digunakan sebagai dasar evaluasi	-
Data skenario batas	Data sintetis (disusun peneliti)	Digunakan untuk kondisi *edge case,* aset tidak *borrowable*, *alert* tanpa *asset*_uuid, simulasi kegagalan *endpoint* AI	Pembentukan skenario uji yang tidak tersedia pada data operasional nyata

Seluruh penyajian hasil pada dokumen penelitian menggunakan identifier yang telah dianonimkan.

Teknik pengumpulan data dilakukan melalui enam cara berikut.

1. Studi dokumen dan proses eksisting untuk memahami formulir, struktur data, aturan kerja, dan pola pengelolaan aset yang telah berjalan.
2. Observasi operasional untuk melihat alur kerja nyata pada pengelolaan aset dan titik masalah yang muncul.
3. Wawancara singkat untuk mengonfirmasi kebutuhan sistem dan ekspektasi pengguna.
4. Observasi dengan NDC secara terbatas untuk menelaah ketersediaan metrik serta pola *alert* pada panel Grafana/Prometheus sebagai konteks evaluasi *health*.
5. Pemanfaatan log sistem dan *snapshot* basis data untuk menyediakan data pengembangan dan pengujian.

6. Catatan uji skenario untuk merekam hasil pengujian terhadap fungsi dan perilaku sistem.

3.3.1 Instrumen Pengumpulan Data

Untuk meningkatkan kemudahan metodologi, data primer dikumpulkan menggunakan dua instrumen utama, yaitu panduan wawancara semi-terstruktur dan *checklist* observasi operasional. Panduan wawancara digunakan untuk menggali alur pengelolaan aset pascaakuisisi, kendala operasional, ketidaksinkronan data antarsumber, faktor prioritas penanganan aset, kebutuhan rekomendasi alokasi, serta persepsi risiko penggunaan AI dalam proses operasional. Pertanyaan terkait estimasi durasi triase digunakan sebagai indikasi awal operasional, sedangkan pengukuran waktu seleksi yang lebih terstruktur dirancang pada Subbab 3.8.3 dan belum dieksekusi dalam lingkup penelitian ini.

Checklist observasi digunakan untuk mencatat ketersediaan dan konsistensi data operasional, meliputi data induk aset, status peminjaman, histori isu dan pemeliharaan, ketersediaan data pemantauan dan *alert,* metrik, konsistensi identitas aset, serta titik keputusan yang masih bergantung pada pengalaman personal operator. Instrumen *feedback* operator berfungsi sebagai instrumen pelengkap setelah responden mencoba alur sistem secara langsung dan tidak diposisikan sebagai hasil utama.

3.3.2 Observasi Monitoring NDC

Observasi monitoring NDC difokuskan pada pengambilan data metrics, *asset* dan *alert* dari Grafana/Prometheus untuk aset termonitor, terutama *server*, *router*, dan *firewall.* Fokus observasi mencakup ketersediaan data metrik utama (CPU, memori, dan jaringan), konsistensi identitas aset pada label monitoring, serta pemanfaatan data *alert* sebagai evaluasi status *health* aset. Terdapat beberapa data yang tidak tersedia dari NDC seperti *disk*, *allert network* pada *asset server*. Sehingga diputuskan untuk menggunakan data sintetis (lihat Tabel 3.3).

Observasi ini berfungsi sebagai dukungan konteks operasional bagi fitur AI *Insight* dan AI *Health Evaluation*, tanpa bermaksud menjadi audit menyeluruh terhadap arsitektur NDC organisasi.

3.4 Metode Pengembangan Sistem

Pengembangan sistem dilakukan secara bertahap dan iteratif. Tahapan pengembangan disusun untuk memastikan bahwa artefak yang dibangun berangkat dari kebutuhan

operasional, memiliki rancangan yang jelas, dapat diimplementasikan, dan dapat diuji secara sistematis. Tahapan tersebut meliputi analisis kebutuhan, desain sistem, implementasi, pengujian, serta evaluasi dan perbaikan.

3.4.1 Analisis Kebutuhan Fungsional dan Nonfungsional

Analisis kebutuhan dilakukan dengan menurunkan masalah operasional BSI UII menjadi kebutuhan sistem yang dapat dirancang dan diuji. Kebutuhan fungsional utama meliputi pengelolaan data aset pascaakuisisi, peminjaman dan pengembalian, pencatatan isu, pemeliharaan, integrasi konteks monitoring, serta penyediaan dukungan keputusan melalui AI *Assign*, AI *Insight*, dan AI *Health*. Kebutuhan nonfungsional meliputi keterlacakan histori aset, validasi keluaran AI, keamanan akses internal, kemampuan *fallback* saat layanan AI gagal, dan auditabilitas *reasoning* rekomendasi.

3.4.2 Desain Arsitektur dan Model Data

Tahap desain dilakukan dengan menyusun arsitektur berlapis yang memisahkan *frontend*, *backend* utama, basis data, dan LangChain AI *service*. Pemisahan LangChain AI *service* dari *backend* utama dipilih agar integrasi LLM, penyusunan *prompt*, parsing keluaran, dan konfigurasi provider model dapat dikelola pada lapisan khusus tanpa mengubah modul operasional inti. Model data dirancang untuk mendukung relasi aset, histori penggunaan, i*ssue, maintenance, snapshot health, cache insight,* dan referensi status domain.

3.4.3 Implementasi Sistem Berlapis

Sistem diimplementasikan dalam empat lapisan yang dipisahkan berdasarkan tanggung jawab. Pemisahan ini dipilih agar logika bisnis inti tidak bergantung secara langsung pada penyedia model AI, sehingga sistem tetap dapat beroperasi meskipun layanan AI mengalami gangguan.

Lapisan *frontend* berfungsi sebagai antarmuka operator untuk mengelola data aset dan menampilkan keluaran AI. Lapisan *backend core* mengelola proses bisnis, validasi data, serta orkestrasi pemanggilan AI. Lapisan *database* menyimpan data operasional, histori, dan hasil evaluasi. Lapisan LangChain AI *service* bertanggung jawab atas penyusunan konteks, pemanggilan model, dan parsing keluaran sebelum dikembalikan ke *backend*.

Keputusan desain penting pada implementasi ini adalah penggunaan TOON sebagai format pertukaran konteks AI secara *end-to-end*. *Backend* menyusun konteks dalam TOON

dan mengirimkannya langsung ke LangChain AI *service* tanpa pembungkus JSON. Pendekatan ini dipilih untuk efisiensi konteks prompt, sebagaimana diukur pada Subbab 4.7.1. Selain itu, sistem menyediakan *endpoint* pengujian internal untuk AI *Health Evaluation* agar *pipeline* yang biasanya dijalankan oleh job terjadwal tetap dapat dieksekusi secara reproduksibel selama penelitian.

3.4.4 Pengujian Sistem

Pengujian sistem dirancang untuk memverifikasi bahwa artefak yang dikembangkan memenuhi kebutuhan fungsional, keandalan integrasi, dan kualitas keluaran yang diharapkan. Strategi pengujian dibagi menjadi empat kategori berdasarkan objek dan tujuan evaluasi.

Pengujian fungsional dilakukan terhadap modul inti sistem menggunakan *black box testing* untuk memastikan fungsi utama seperti pengelolaan aset, peminjaman, isu, dan pemeliharaan berjalan sesuai spesifikasi. Pengujian integrasi memverifikasi komunikasi antara *backend* dan LangChain AI *service*, termasuk validasi pertukaran data TOON dan penanganan *error*. Pengujian performa dan keandalan mengevaluasi respons sistem pada beban terbatas serta ketahanan mekanisme *retry* dan *fallback* terhadap gangguan layanan AI. Evaluasi kualitas keluaran AI menggunakan pendekatan *domain-based expert evaluation* untuk menilai relevansi dan keakuratan *insight*, rekomendasi alokasi, dan status kesehatan dari perspektif ahli domain. Rincian desain pengujian per kategori diuraikan pada Subbab 3.8.

3.4.5 Evaluasi dan Iterasi

Tahap evaluasi dilakukan dengan meninjau hasil pengujian dan umpan balik domain untuk mengetahui apakah artefak memenuhi kebutuhan yang telah dirumuskan. Temuan seperti keluaran AI tidak valid, *fallback* belum berjalan, atau konteks input belum cukup jelas digunakan sebagai dasar perbaikan terbatas pada *prompt*, validasi output, aturan *pipeline*, atau konfigurasi integrasi. Ringkasan tahapan pengembangan sistem dirangkum pada Tabel 3.4.

Tabel 3.4 Ringkasan tahapan pengembangan sistem

Tahap	Fokus kegiatan	Output utama
Analisis kebutuhan	Identifikasi masalah operasional, kebutuhan pengguna, dan batasan studi kasus	Daftar kebutuhan fungsional dan nonfungsional
Desain sistem	Perancangan arsitektur, model data, aturan bisnis, dan kontrak integrasi AI	Rancangan arsitektur, ERD, aturan bisnis, dan *endpoint*

Tahap	Fokus kegiatan	Output utama
Implementasi	Pembangunan *frontend*, *backend* utama, basis data, dan layanan AI internal	Artefak sistem berjalan pada lingkungan uji
Pengujian	Pengujian fungsional, integrasi, performa terbatas, *fault injection*, skenario bisnis, dan evaluasi AI	Bukti hasil pengujian dan catatan evaluasi
Evaluasi dan iterasi	Analisis temuan, perbaikan prompt/validasi/*fallback*, dan dokumentasi perubahan	Artefak revisi dan rekam jejak iterasi

3.5 Perancangan Arsitektur dan Alur Sistem

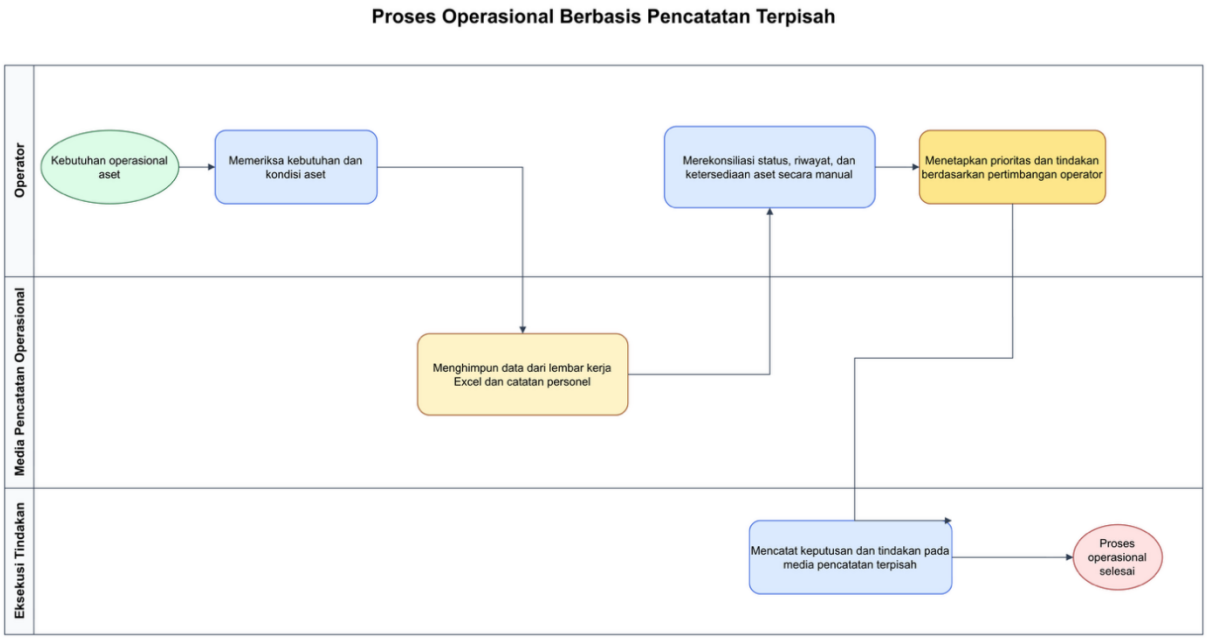
Perancangan arsitektur sistem bertujuan untuk memisahkan tanggung jawab antarkomponen agar sistem lebih mudah dikembangkan, diuji, dan dipelihara. Sistem dirancang dalam tiga layanan utama, yaitu *frontend, backend core,* dan LangChain AI *service. Frontend* berfungsi sebagai lapisan interaksi dengan pengguna. *Backend* berfungsi sebagai pusat logika bisnis, pengelolaan data, dan pengaturan alur proses. LangChain *service* berfungsi sebagai komponen khusus yang memproses permintaan terkait inferensi AI sebelum diteruskan ke penyedia LLM. Seluruh diagram pada Subbab 3.5 disajikan dalam format gambar pada naskah akhir.

Alur umum sistem dimulai dari permintaan pengguna pada *frontend*. Permintaan tersebut diteruskan ke *backend* untuk divalidasi dan dipetakan ke proses bisnis yang sesuai. Jika proses membutuhkan komponen AI, *backend* menyiapkan konteks yang relevan berdasarkan data aset, histori operasional, dan data monitoring, lalu meneruskannya ke LangChain *service.* Hasil dari LangChain *service* kemudian dikembalikan ke *backend* untuk divalidasi, disimpan bila diperlukan, dan ditampilkan kembali ke pengguna melalui *frontend*.

3.5.1 Perancangan Proses Bisnis Operasional

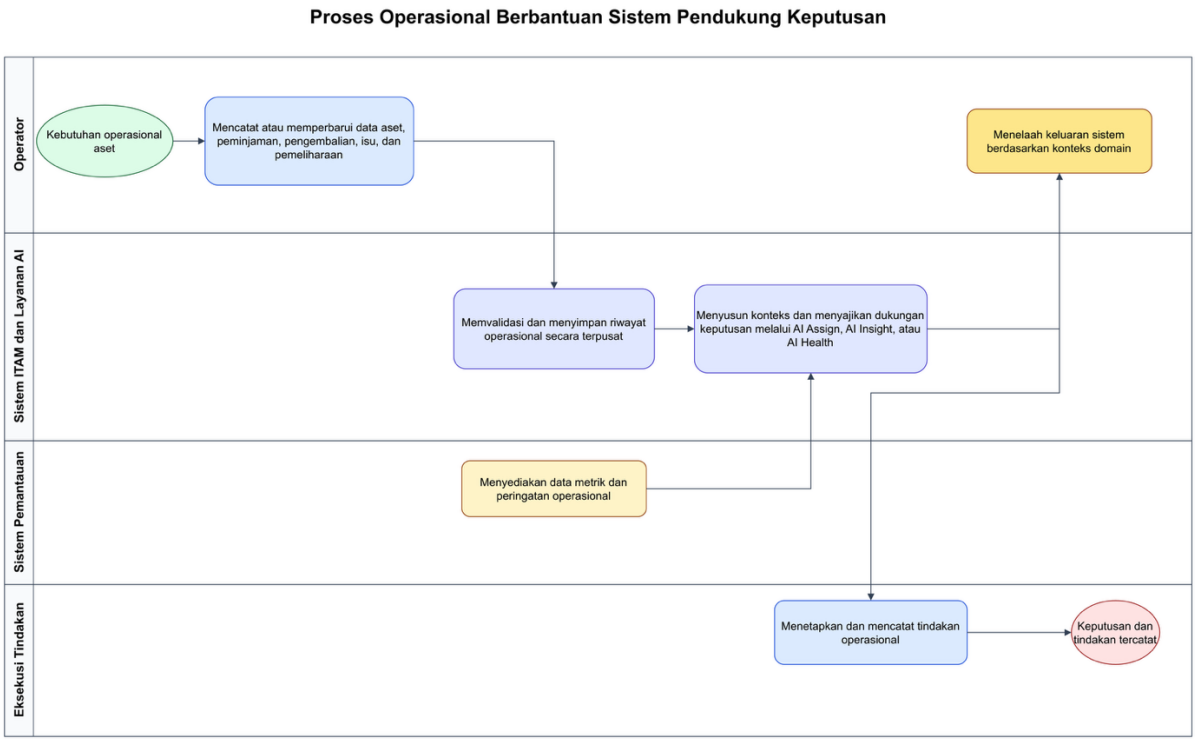
Perancangan proses bisnis operasional digunakan untuk memperjelas perubahan alur kerja pengelolaan aset pascaakuisisi dari proses berbasis pencatatan terpisah menuju proses berbantuan sistem pendukung keputusan. Karena sistem yang dirancang merupakan sistem internal BSI UII, penyajian pada penelitian ini tidak menggunakan Business Model Canvas, melainkan pemodelan proses operasional berbasis BPMN. Pendekatan ini dipilih karena fokus penelitian bukan model komersial, tetapi perbaikan alur kerja, integrasi data, dan dukungan keputusan operasional.

Proses pertama menggambarkan proses operasional berbasis pencatatan terpisah. Pada proses ini, data referensi aset diperoleh dari pencatatan tingkat universitas melalui berkas Excel atau *snapshot* data, sedangkan aktivitas operasional harian di internal BSI masih memanfaatkan spreadsheet dan catatan personel. Operator perlu merekonsiliasi beberapa sumber data sebelum menentukan kondisi aset, ketersediaan aset, atau prioritas tindak lanjut. Alur ini membuat interpretasi kondisi aset bergantung pada pengalaman personel dan belum memiliki jejak keputusan yang sepenuhnya terpusat.



Gambar 3. 2 BPMN proses operasional berbasis pencatatan terpisah

Proses kedua menggambarkan proses operasional berbantuan sistem pendukung keputusan. Pada proses ini, data aset, peminjaman, *issue, maintenance*, *monitoring*, dan hasil evaluasi AI dipusatkan pada sistem. Sistem menyediakan rekomendasi alokasi, *insight* kondisi aset, serta evaluasi kesehatan aset sebagai bahan pendukung keputusan. Hasil AI tetap melewati validasi sistem dan verifikasi operator sehingga keputusan akhir tidak berpindah kepada model, tetapi tetap berada pada pihak operasional yang berwenang.



Gambar 3. 3 BPMN proses operasional berbantuan sistem pendukung keputusan

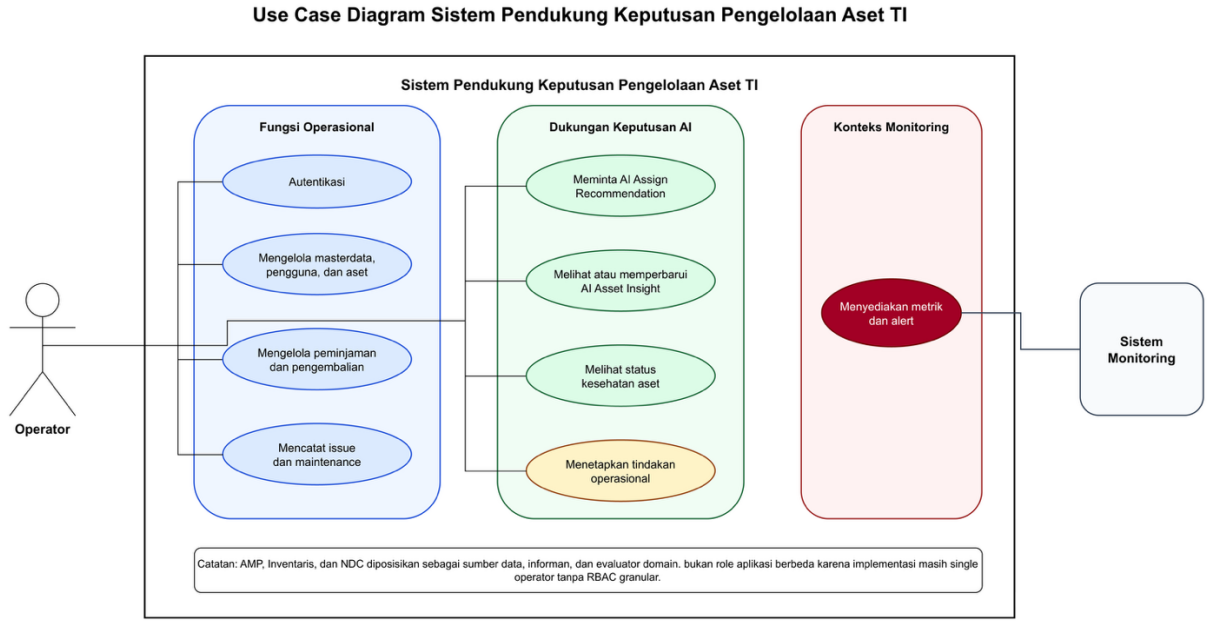
Perbandingan ringkas antara kedua proses operasional tersebut disajikan pada Tabel 3. 5.

Tabel 3. 5 Perbandingan proses operasional pencatatan terpisah dan berbantuan sistem

Aspek	**Proses berbasis pencatatan terpisah**	**Proses berbantuan sistem pendukung keputusan**
Sumber data operasional	Spreadsheet, catatan personel, dan referensi aset dari pencatatan tingkat universitas	Data operasional aset disatukan dalam sistem pascaakuisisi
Pencatatan histori	Tersebar pada beberapa media pencatatan	Histori peminjaman, *issue, maintenance,* dan *health snapshot* tersimpan terpusat
Evaluasi kondisi aset	Dilakukan manual berdasarkan pengalaman operator	Didukung AI *Health* dan *guardrail* berbasis aturan
Interpretasi kondisi	Operator membaca data mentah dari beberapa sumber	AI *Asset Insight* menyajikan ringkasan kondisi dan rekomendasi awal
Alokasi aset	Seleksi kandidat dilakukan manual	AI *Assign* memberi prioritas kandidat sebagai bahan pertimbangan
Keputusan akhir	Operator mengambil keputusan berdasarkan rekonsiliasi manual	Operator tetap mengambil keputusan setelah memverifikasi output sistem

3.5.2 Arsitektur Umum Sistem

Use case diagram digunakan untuk menunjukkan hubungan antara aktor dengan fungsi utama sistem. Aktor utama yang dimodelkan adalah Operator, yaitu pengguna terautentikasi yang menjalankan fungsi operasional pada aplikasi. Selain itu, Sistem Pemantauan dimodelkan sebagai aktor eksternal yang menyediakan metrik dan *alert* sebagai konteks analisis. Pembagian hak akses granular antara AMP, Inventaris, dan NDC tidak dimodelkan sebagai aktor terpisah karena ruang lingkup implementasi dan pengujian sistem masih berada pada konteks *single* operator. Peran organisasi tersebut tetap digunakan sebagai sumber data dan evaluator domain pada tahap evaluasi, tetapi bukan sebagai *role* aplikasi yang berbeda. Komponen teknis seperti LangChain AI *service* dan penyedia LLM tidak ditempatkan sebagai aktor *use case* karena keduanya merupakan bagian dari alur implementasi teknis yang dijelaskan pada *sequence* diagram.



Gambar 3. 4 *Use case* diagram sistem pendukung keputusan pengelolaan aset TI

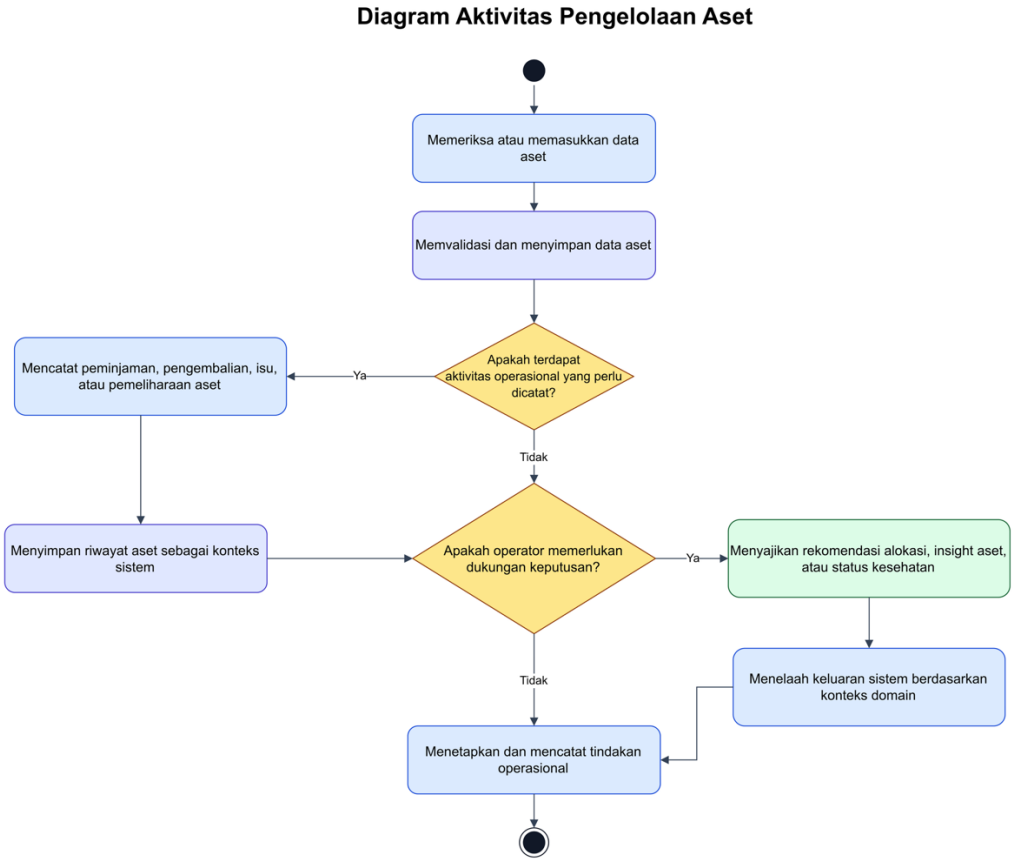
Tabel 3. 6 *Use case* inti sistem

Aktor	***Use case***	**Tujuan operasional**
Operator	Autentikasi	Membatasi akses aplikasi pada pengguna internal yang berwenang
Operator	Mengelola masterdata, pengguna, dan aset	Menjaga data pascaakuisisi tersedia sebagai basis operasional

Aktor	*Use case*	Tujuan operasional
Operator	Mengelola peminjaman dan pengembalian	Mencatat status penggunaan dan ketersediaan aset
Operator	Mencatat *issue* dan *maintenance*	Menyimpan histori gangguan dan tindakan pemeliharaan aset
Operator	Meminta AI *Assign Recommendation*	Memperoleh prioritas kandidat aset sebagai bahan keputusan alokasi
Operator	Melihat atau memperbarui AI *Asset Insight*	Memperoleh ringkasan kondisi aset dan rekomendasi awal
Operator	Melihat status kesehatan aset	Memantau hasil evaluasi kesehatan aset yang tersimpan pada sistem
Operator	Menetapkan tindakan operasional	Mengambil keputusan akhir berdasarkan data sistem dan verifikasi domain
Sistem Pemantauan	Menyediakan metrik dan *alert*	Memberikan konteks operasional untuk *insight* dan *health evaluation*

3.5.3 *Activity Diagram*

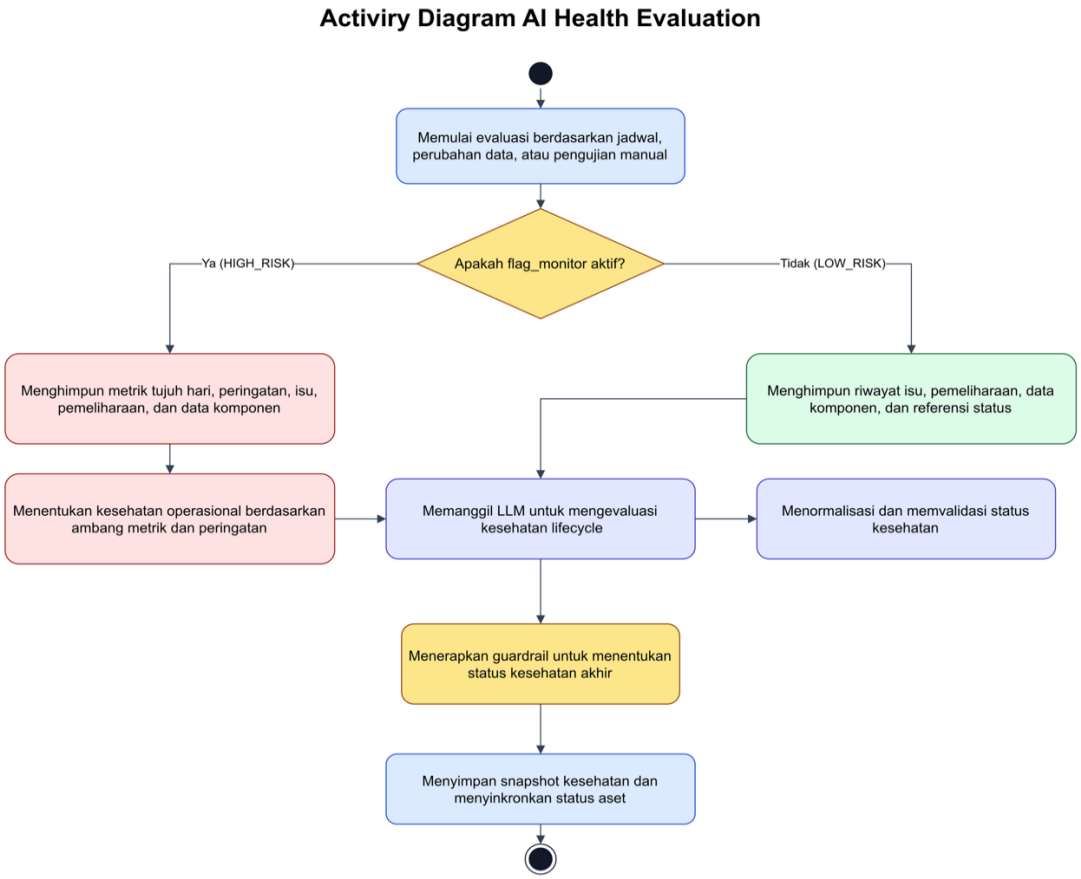
Activity diagram digunakan untuk menjelaskan alur aktivitas utama pada proses operasional dan pada *pipeline* AI *Health Evaluation*. Diagram pertama memodelkan proses pascaakuisisi secara umum, mulai dari pemeriksaan data aset, pencatatan aktivitas operasional, pembentukan dukungan keputusan, sampai penetapan tindakan oleh operator.



Gambar 3. 5 *Activity diagram* proses pengelolaan aset pascaakuisisi

Alur aktivitas pada proses pascaakuisisi dimulai ketika operator memeriksa atau memasukkan data aset. Sistem memvalidasi data tersebut, kemudian operator dapat mencatat peminjaman, pengembalian, *issue*, atau *maintenance*. Setiap aktivitas tersimpan sebagai histori aset dan menjadi konteks bagi fitur AI. Apabila operator membutuhkan dukungan keputusan, sistem menampilkan rekomendasi, *insight*, atau status *health*. Operator kemudian menilai keluaran tersebut bersama konteks domain yang tersedia dan menetapkan tindakan operasional. Dengan demikian, aktivitas verifikasi tidak dimaknai sebagai fitur persetujuan otomatis di dalam aplikasi, melainkan sebagai posisi *human-in-the-loop* pada proses kerja operator.

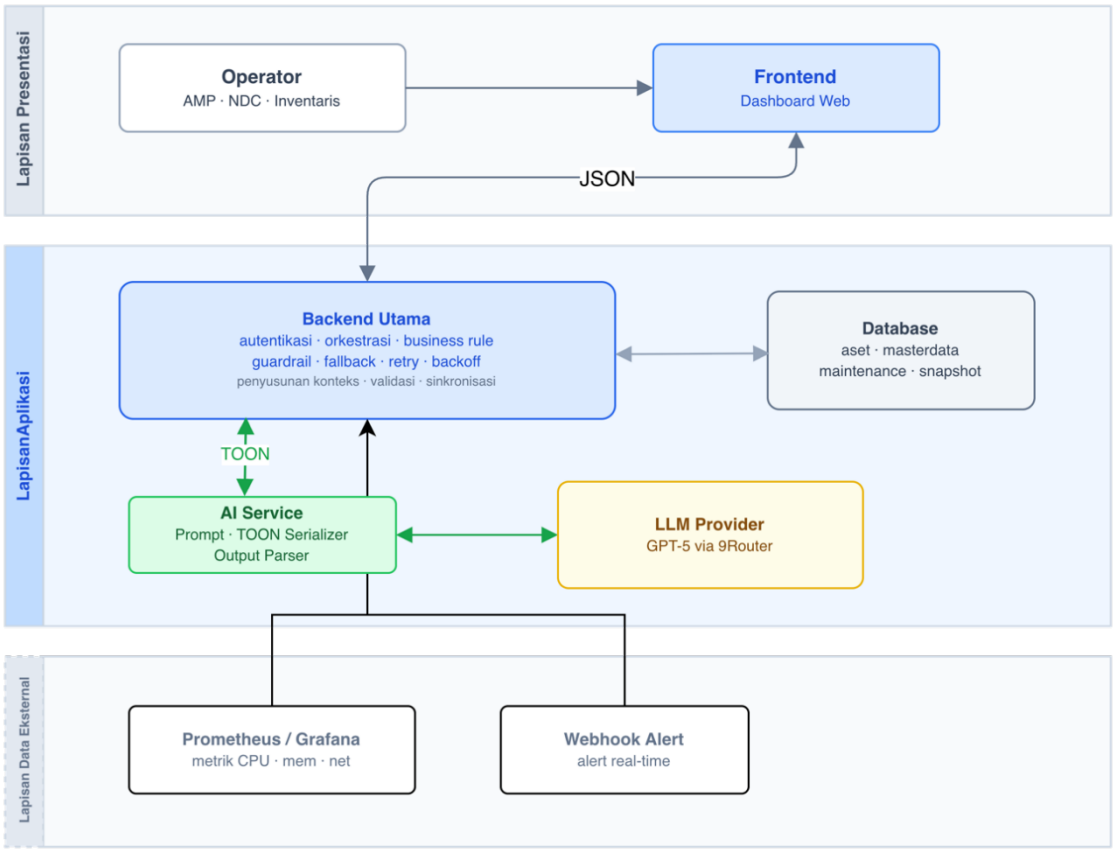
Diagram kedua memodelkan aktivitas AI *Health Evaluation*. Alur dimulai dari pengambilan data aset dan penentuan kelas risiko. Untuk aset *high-risk*, sistem mengambil metrik dan *alert*, menjalankan aturan *operational health*, memanggil evaluasi *lifecycle* berbasis LLM, lalu menerapkan *guardrail*. Untuk aset *low-risk*, sistem lebih bertumpu pada histori aset dan keluaran LLM, tetapi status tetap divalidasi terhadap masterdata. Hasil akhir disimpan sebagai *health snapshot* dan ditampilkan kepada operator.



Gambar 3. 6 *Activity diagram* AI *Health Evaluation*

3.5.4 Arsitektur Umum Sistem

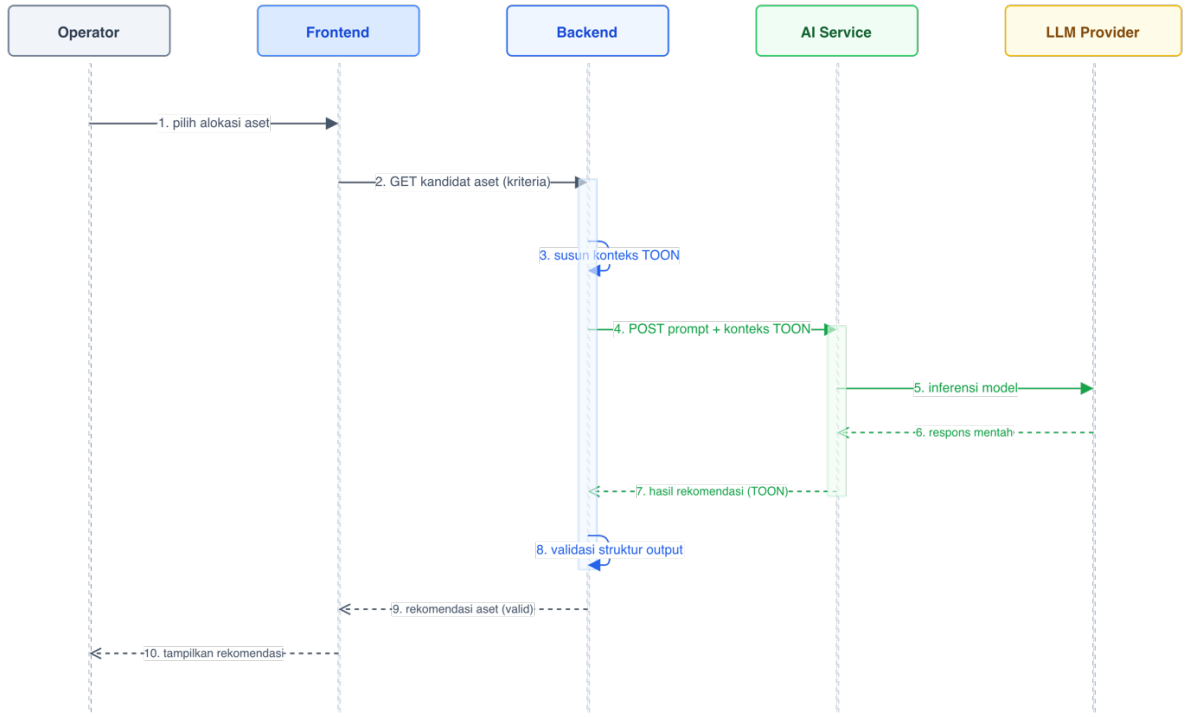
Arsitektur umum sistem dirancang agar komponen inti pengelolaan aset tidak bergantung secara langsung pada penyedia model AI. Pemisahan ini penting agar sistem tetap dapat beroperasi meskipun layanan AI mengalami gangguan. Dengan pendekatan ini, *backend* bertindak sebagai pengendali utama proses bisnis dan hanya memanggil LangChain *service* pada fitur yang memang membutuhkan analisis berbasis LLM. Konfigurasi tiga layanan tersebut diilustrasikan pada Gambar 3. 2.



Gambar 3.7 Arsitektur umum *system*

3.5.5 Alur AI *Assign Recommendation*

Fitur AI *Assign Recommendation* dirancang untuk membantu rekomendasi alokasi aset kepada calon pengguna aset berdasarkan data aset dan konteks kebutuhan tertentu. Alur dimulai ketika pengguna atau operator memilih proses alokasi aset. *Backend* mengambil data kandidat aset yang relevan, data status aset, dan konteks kebutuhan pengguna. Data tersebut disusun menjadi konteks terstruktur dan dikirim ke LangChain *service*. Hasil rekomendasi yang diterima kemudian divalidasi oleh *backend* sebelum ditampilkan kepada pengguna sebagai bahan pertimbangan. Alur lengkap interaksi antarlayanan pada fitur ini disajikan pada Gambar 3. 3.

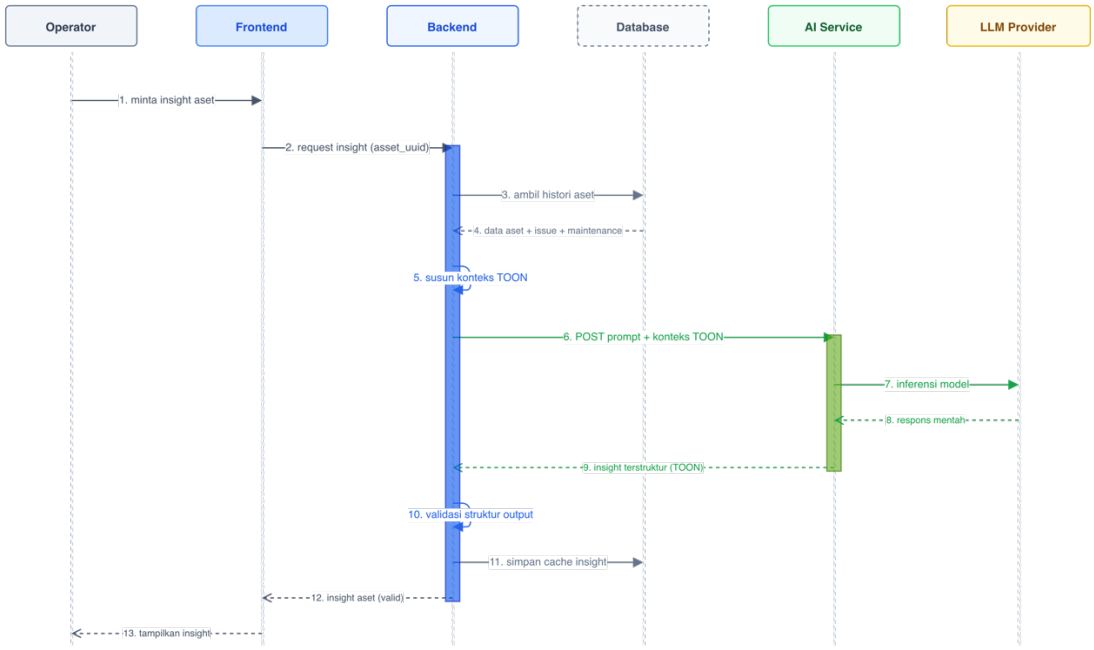


Gambar 3.8 *Sequence* diagram alur AI *Assign Recommendation*

3.5.6 Alur AI *Asset Insight*

Fitur AI *Asset Insight* dirancang untuk menghasilkan narasi analitis mengenai kondisi aset. Alur dimulai ketika sistem atau pengguna meminta *insight* terhadap aset tertentu. *Backend* mengambil data aset, histori isu, histori pemeliharaan, serta data monitoring apabila tersedia. Konteks ini kemudian diproses oleh LangChain *service* untuk menghasilkan narasi yang bersifat informatif. Hasil akhir ditampilkan kepada pengguna dan dapat disimpan sebagai *cache* atau riwayat *insight* untuk kebutuhan akses berikutnya.

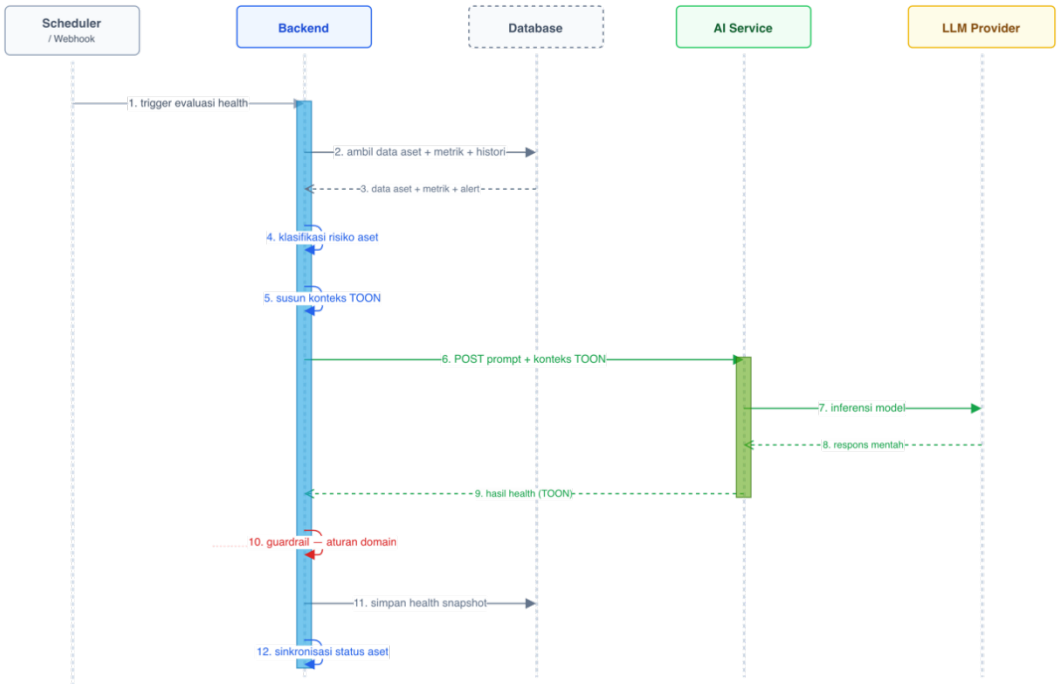
Hasil *insight* disimpan dalam *cache* dengan masa berlaku 24 jam. Apabila *cache* masih segar, permintaan *insight* berikutnya untuk aset yang sama dikembalikan langsung dari *cache* tanpa pemanggilan LLM ulang. Pada permintaan *insight* dengan *force refresh*, *pipeline health evaluation* dijalankan terlebih dahulu sebelum inferensi *insight*, sehingga data kesehatan aset yang digunakan sebagai konteks selalu mencerminkan kondisi terkini. Alur lengkap fitur AI *Asset Insight*, termasuk logika *cache* dan *force refresh*, diilustrasikan pada Gambar 3. 4.



Gambar 3.9 *Sequence* diagram alur AI *Asset Insight*

3.5.7 Alur AI *Health Evaluation*

Fitur AI *Health Evaluation* dirancang untuk membantu evaluasi status kesehatan aset melalui alur internal sistem. Pada alur ini, sistem membedakan aset berdasarkan kelas risiko. Untuk aset dengan risiko tinggi, proses evaluasi memadukan hasil analisis AI dengan validasi aturan tertentu. Untuk aset dengan risiko lebih rendah, AI dapat berperan lebih dominan dalam menyusun evaluasi, namun hasilnya tetap diperiksa terhadap referensi status yang berlaku pada sistem. Pemisahan alur ini dimaksudkan agar pendekatan evaluasi tetap proporsional terhadap tingkat kritisitas aset. Hasil evaluasi kemudian digunakan untuk memperbarui *snapsho*t dan status kesehatan aset pada *backend*. Alur lengkap pipeline evaluasi kesehatan, termasuk pemisahan jalur berdasarkan kelas risiko, disajikan pada Gambar 3. 5.



Gambar 3.10 Sequence diagram alur AI *Health Evaluation*

Sistem menerapkan dua tingkat penjadwalan *health evaluation*. Tingkat pertama menjalankan evaluasi untuk seluruh aset dengan *flag monitor* aktif (aset risiko tinggi) setiap hari. Tingkat kedua menjalankan evaluasi untuk aset *non*-monitoring (aset risiko rendah) dengan *interval refresh* 30 hari per *asset*, aset yang *snapshot* kesehatannya masih berusia kurang dari 30 hari dilewati untuk menghindari penggunaan token yang tidak perlu.

3.5.8 Alur Integrasi Monitoring dan *Alert*

Sistem dirancang untuk memanfaatkan data monitoring dan *alert* sebagai konteks operasional melalui dua mekanisme. Pertama, data metrik dapat diambil secara berkala melalui job terjadwal. Kedua, data *alert* dapat masuk melalui mekanisme *webhook* untuk memperbarui konteks secara *real-time.* Kedua mekanisme ini memperkaya konteks analisis dan membantu sistem menangkap perubahan kondisi aset secara lebih cepat.

Pada alur *webhook, backend* terlebih dahulu memvalidasi status alert dengan berpacuan pada keberadaan tag `asset_uuid`. *Alert* yang memiliki identitas aset valid dipetakan ke data aset internal dan dapat memicu penyegaran konteks health evaluation. *Alert* tanpa tag `asset_uuid` atau tidak dapat dipetakan ke aset tidak digunakan untuk memperbarui status, tetapi tetap ditangani secara terkendali agar tidak menghentikan *pipeline*.

Pada konteks studi kasus BSI UII, implementasi integrasi *real-time* dengan sistem monitoring Prometheus belum dapat dilakukan karena memerlukan pemetaan identitas aset dan koordinasi lintas unit yang membutuhkan waktu dan pembahasan lebih lanjut. Sebagai gantinya, data monitoring untuk penelitian ini diperoleh dari export CSV NDC sebagai representasi data operasional aktual pada periode tertentu. *Health evaluation* kemudian dijalankan melalui endpoint testing internal yang memungkinkan *pipeline* dieksekusi secara *reproduksibel* menggunakan data monitoring tersebut. Dengan demikian, meskipun mekanisme *real-time* belum terdeploy pada lingkungan penelitian, sistem tetap dapat dievaluasi menggunakan data operasional nyata.

3.6 Perancangan Data dan Aturan Bisnis

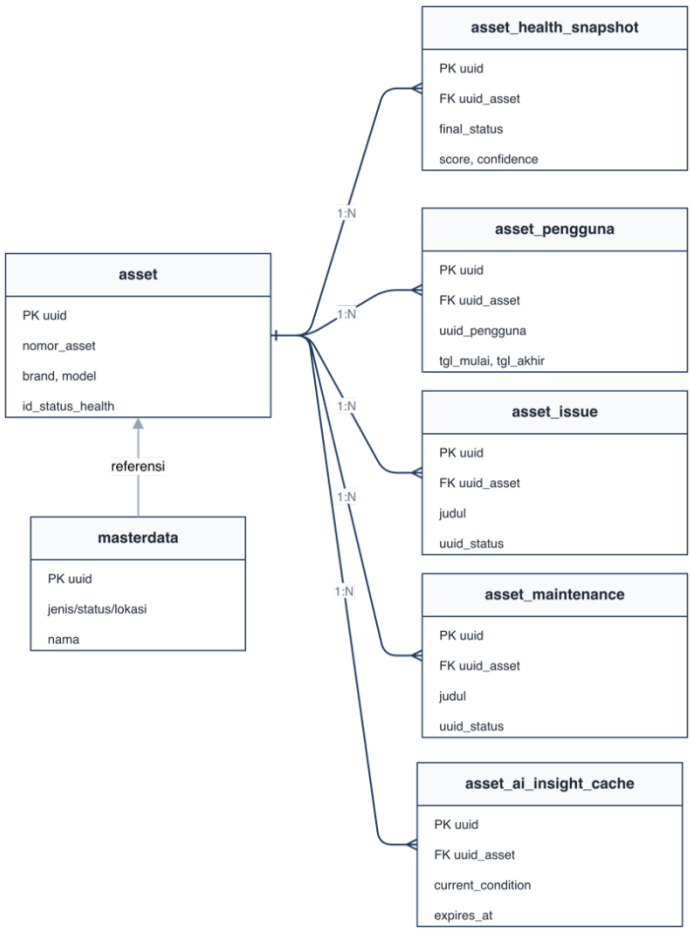
Perancangan data berfokus pada entitas yang mendukung pengelolaan aset dan fitur pendukung keputusan. Entitas utama yang digunakan meliputi data aset, data relasi penggunaan aset, data isu, data pemeliharaan, data *snapshot* kesehatan aset, *cache insight* AI, serta masterdata yang berfungsi sebagai referensi status dan jenis data. Entitas-entitas tersebut saling berhubungan untuk mendukung proses operasional dan proses analisis yang dibutuhkan sistem.

Data aset menjadi pusat relasi karena hampir seluruh proses operasional berangkat dari keberadaan aset tertentu. Relasi penggunaan aset digunakan untuk melacak peminjaman dan pengembalian aset. Data isu dan pemeliharaan digunakan untuk mencatat kejadian operasional dan tindakan penanganan. Data *snapshot* kesehatan aset digunakan untuk menyimpan hasil evaluasi kondisi aset pada waktu tertentu. Sementara itu, *cache insight* AI digunakan untuk menyimpan hasil analisis yang dapat dimanfaatkan kembali tanpa perlu selalu memanggil layanan AI untuk mengejar efisiensi. Daftar entitas utama beserta fungsinya dirangkum pada Tabel 3. 5, sedangkan relasi antarentitas divisualisasikan pada Gambar 3. 6.

Tabel 3.7 Entitas data utama

Entity	Fungsi utama
‘asset’	Menyimpan data utama aset
‘asset_pengguna’	Menyimpan riwayat peminjaman dan pengembalian aset
‘asset_issue’	Menyimpan catatan isu atau gangguan aset
‘asset_maintenance’	Menyimpan catatan pemeliharaan aset
‘asset_health_snapshot’	Menyimpan hasil evaluasi kesehatan aset

Entity	Fungsi utama
‘asset_ai_insight_cache’	Menyimpan hasil *insight* AI per aset
‘asset_metric_hourly’	Menyimpan ringkasan metrik monitoring per jam
‘asset_alerts’	Menyimpan data *alert* yang dapat dipetakan ke aset
masterdata.status	Referensi status sistem
masterdata.jenis	Referensi kategori atau jenis data



Gambar 3.11 Ringkasan model data utama (ERD)

Perancangan basis data pada penelitian ini disajikan pada tingkat konseptual dan logis, bukan sebagai dump skema SQL penuh. Fokus rancangan adalah memastikan setiap data operasional dan keluaran AI dapat ditelusuri kembali ke aset yang sama. Ringkasan relasi inti basis data disajikan pada Tabel 3. 8.

Tabel 3. 8 Relasi inti perancangan basis data

Entitas	Primary key	Relasi utama	Fungsi dalam sistem
asset	uuid	Berelasi ke seluruh histori operasional dan evaluasi	Pusat data aset pascaakuisisi
asset_pengguna	uuid	FK ke *asset* dan data pengguna	Mencatat histori peminjaman dan pengembalian
asset_issue	uuid	FK ke *asset*	Mencatat gangguan atau isu aset
asset_maintenance	uuid	FK ke *asset*	Mencatat tindakan pemeliharaan aset
asset_health_snapshot	uuid	FK ke *asset*	Menyimpan hasil evaluasi health pada waktu tertentu
asset_ai_insight_cache	uuid	FK ke *asset*	Menyimpan hasil *insight* aktif sesuai masa berlaku *cache*
asset_metric_hourly	uuid	FK ke *asset*	Menyimpan ringkasan metrik monitoring sebagai konteks analisis
asset_alerts	uuid	FK ke *asset* bila 'asset_uuid valid'	Menyimpan *alert* yang relevan dengan aset tertentu
masterdata.status	uuid	Direferensikan oleh aset dan hasil evaluasi	Menjaga konsistensi nilai status domain
masterdata.jenis	uuid	Direferensikan oleh aset	Menyimpan kategori dan atribut jenis aset

Dengan rancangan tersebut, hubungan utama dalam ERD mengikuti pola satu-ke-banyak dari *asset* menuju histori penggunaan, *issue, maintenance*, metrik, *alert*, dan *health snapshot*.

Untuk *insight*, sistem menyimpan *cache* per aset agar hasil analisis dapat digunakan ulang selama masih aktif dan belum melewati masa berlaku. Pola relasi ini dipilih agar perubahan kondisi aset, alasan evaluasi AI, dan riwayat tindakan operasional tetap dapat ditelusuri secara konsisten.

Selain model data, penelitian ini juga mendefinisikan aturan bisnis inti. Salah satu aturan penting adalah pemetaan kelas risiko aset berdasarkan penanda kemampuan monitoring pada jenis aset, sehingga aset yang memang ditetapkan sebagai aset monitoring diperlakukan berbeda dari aset non-monitoring. Klasifikasi ini terdiri dari dua kelas, yaitu aset risiko tinggi dan aset risiko rendah. Aset risiko tinggi adalah aset dengan penanda monitoring aktif yang merepresentasikan infrastruktur kritis seperti *server* dan perangkat jaringan inti, sehingga evaluasinya memadukan keluaran LLM dengan lapisan *rule-based guardrail* yang dapat menimpa status final. Aset risiko rendah adalah aset tanpa data monitoring, umumnya *workstation* dan periferal, sehingga evaluasinya lebih mengandalkan LLM dengan validasi terhadap referensi status domain. Pembedaan ini berakar pada perbedaan konsekuensi operasional antara aset CAPEX dan OPEX sebagaimana diuraikan pada Subbab 2.1.3, serta sejalan dengan prinsip manajemen aset berbasis risiko (Albalghouni et al., 2018). Perlu dicatat bahwa penanda monitoring pada jenis aset digunakan sebagai proksi operasional terhadap kritisitas aset, bukan sebagai hasil asesmen risiko formal per individu aset. Dengan demikian, aset kritis yang belum dilengkapi monitoring tidak otomatis diklasifikasikan sebagai aset risiko rendah, dan keterbatasan ini diakui sebagai konsekuensi ruang lingkup data yang tersedia pada penelitian ini. Aturan lain adalah bahwa pembaruan status kesehatan aset hanya dilakukan melalui *pipeline* evaluasi kesehatan, sehingga perubahan status tetap terkendali. Untuk *insight* AI, sistem menggunakan mekanisme penyimpanan hasil terkini per aset agar proses akses ulang lebih efisien dan tidak selalu memerlukan inferensi baru jika konteks belum berubah secara signifikan.

Klasifikasi kelas risiko aset dirangkum pada Tabel 3. 9 untuk memperjelas dasar pemisahan perlakuan evaluasi pada AI *Health Evaluation*.

Tabel 3. 9 Klasifikasi kelas risiko aset pada AI *Health Evaluation*

Kelas risiko	**Kriteria operasional**	**Contoh aset**	**Data evaluasi utama**	**Perlakuan sistem**
High-risk	Memiliki penanda monitoring aktif dan	Server, *router,*	Metrik CPU/RAM/network,	Evaluasi *hybrid* antara

Kelas risiko	Kriteria operasional	Contoh aset	Data evaluasi utama	Perlakuan sistem
	merepresentasikan aset infrastruktur dengan konsekuensi gangguan lebih luas	*firewall*, dan perangkat jaringan inti	*alert, issue, maintenance*, dan konteks *lifecycle*	LLM dan operasional *rule* dengan *guardrail override* pada kondisi kritis
Low-risk	Tidak memiliki monitoring langsung dan umumnya berperan sebagai aset operasional pengguna akhir	Laptop, *workstation*, dan *periferal*	Histori *issue, maintenance,* status inventaris, dan konteks *lifecycle*	Evaluasi LLM lebih dominan dengan validasi status terhadap masterdata

Klasifikasi tersebut merupakan stratifikasi operasional untuk kebutuhan rancangan sistem, bukan asesmen risiko formal per individu aset. Dengan demikian, flag_monitor dipakai sebagai proksi terhadap ketersediaan sinyal operasional dan tingkat kritisitas jenis aset pada ruang lingkup penelitian. Keterbatasan ini penting karena aset yang secara organisasi kritis tetapi belum memiliki monitoring langsung tidak otomatis dapat disimpulkan sebagai aset risiko rendah secara substantif.

Aturan bisnis juga menjaga konsistensi hubungan antarmodul. Misalnya, aset yang sedang dipinjam tidak dapat diperlakukan sama dengan aset yang tersedia. Aset yang memiliki isu aktif dan belum selesai dapat memengaruhi prioritas penanganan. Catatan pemeliharaan menjadi bagian dari histori yang akan memengaruhi analisis kondisi pada fitur AI *Health* dan AI *Insight*. Aturan bisnis inti yang diterapkan dirangkum pada Tabel 3. 6.

Tabel 3.10 *Use case* inti

Aturan	Tujuan
Status kesehatan aset diperbarui melalui *pipeline* evaluasi	Menjaga konsistensi status
Insight AI disimpan per aset	Meningkatkan efisiensi akses hasil
Kelas risiko aset dipetakan dari penanda monitoring pada jenis aset	Membantu diferensiasi evaluasi
Histori isu dan pemeliharaan menjadi konteks analisis	Memperkaya evaluasi kondisi

3.7 Perancangan Metode AI pada Sistem

Metode AI pada penelitian ini dirancang untuk mendukung tiga fitur pendukung keputusan, yaitu rekomendasi alokasi aset (AI *Assign*), penyusunan *insight* operasional (AI *Insight*), dan evaluasi kesehatan aset (AI *Health*). Ketiga fitur tersebut berbagi prinsip desain yang sama, yaitu penggunaan TOON sebagai format penyusunan konteks input, penerapan *guardrail* berbasis aturan untuk menjaga auditabilitas keluaran, serta mekanisme *retry* dan *fallback* pada integrasi LLM untuk mempertahankan keandalan layanan. Namun, ketiga fitur ini juga memiliki karakteristik yang berbeda dalam hal tingkat kontrol terhadap keluaran model, kompleksitas konteks yang diperlukan, serta bentuk output yang dihasilkan. Perbedaan ini memengaruhi desain *pipeline*, validasi keluaran, dan mekanisme *fallback* yang diterapkan pada masing-masing fitur.

3.7.1 Penyusunan Konteks Input

Agar LLM dapat memberikan keluaran yang relevan, sistem terlebih dahulu menyusun konteks input dari berbagai sumber data. Konteks tersebut dapat mencakup data identitas aset, spesifikasi dasar, data status, histori isu, histori pemeliharaan, data monitoring, dan informasi lain yang relevan dengan fitur yang dipanggil. Konteks utama disusun dalam bentuk *Token-Oriented Object Notation* (TOON) sebagai representasi terstruktur yang ringkas (Schopplich, 2025). Pada saat pertukaran antar-*services*, konteks tersebut dikirim dalam format TOON, bukan di-*wrap* dalam JSON.

Pemilihan TOON didasarkan pada pertimbangan desain, bukan hanya preferensi format. JSON dipertahankan sebagai format pertukaran antar-*service* utama karena stabil untuk kontrak API, tetapi kurang ringkas ketika seluruh konteks dimasukkan ke *prompt*. YAML dan TOML lebih mudah dibaca manusia, namun lebih rentan menghasilkan variasi indentasi atau struktur saat diproduksi ulang oleh model. *Comma-Separated Values* (CSV) efisien untuk data tabular, tetapi kurang sesuai untuk konteks aset yang memuat campuran objek, histori, *alert,* dan ringkasan naratif. *Markdown* (MD) terstruktur mudah dibaca, tetapi batas antar *field* tidak selalu tegas untuk kebutuhan parsing ulang. TOON dipilih karena mempertahankan struktur *field,* ringkas untuk daftar, dan cukup mudah diaudit oleh peneliti maupun evaluator domain. Meskipun pembandingan format di atas dilakukan secara konseptual, keputusan final untuk mengadopsi TOON dikonfirmasi melalui proses evaluasi iteratif yang menunjukkan bahwa format JSON standar menghasilkan *overhead* token signifikan pada konteks tabular.

Referensi (Mei et al., 2025) digunakan sebagai rujukan format TOON yang mendasari struktur representasi konteks pada penelitian ini. Lalu, penyusunan konteks dilakukan secara selektif, yaitu sistem hanya memasukkan elemen konteks yang relevan dengan tujuan analisis tertentu. Pendekatan ini penting untuk menjaga fokus analisis, mengurangi data yang tidak perlu, dan meningkatkan efisiensi pertukaran data.

Penerapan TOON pada penelitian ini dirancang agar konteks domain mudah dibaca model dan stabil untuk diparsing ulang. Contoh berikut menunjukkan konteks ringkas untuk satu aset pada *mode insight* monitoring, sebagaimana ditampilkan pada Gambar 3.7.

```
asset:
  uuid: 9e505b7f-991d-4d32-a60d-537e8f31b0c2
  nomor_asset_utama: NAS-001
  brand: Dell
  model: PowerEdge R740
  uuid_jenis: uuid-jenis-nas
  uuid_lokasi: uuid-lokasi-dc-jogja

maintenance_history[1]{uuid,tgl,judul,uuid_status}:
  mt-001,2026-04-20,Preventive RAID check,st-maint-done

known_issue_risks[1]{uuid,tgl,judul,uuid_status,flag_aktif}:
  is-009,2026-04-22,CPU usage spike,st-issue-open,1

metrics_hourly:
  cpu_avg_pct: 42
  ram_avg_pct: 55
  net_avg_kbps: 2500

alerts_recent[1]{alert_name,severity,starts_at,ends_at}:
  HighCPUUsage,warning,2026-04-22T09:30:00Z,2026-04-22T09:40:00Z

insight_context:
  risk_class: HIGH_RISK
  insight_mode: monitoring
```

Gambar 3.12 Representasi konteks TOON untuk aset monitoring.

Pemilihan *field* pada contoh TOON tersebut didasarkan pada kebutuhan keterlacakan konteks. Kelompok *asset* digunakan sebagai identitas inti agar keluaran model tetap terikat pada aset yang sedang dianalisis. *Field* `maintenance_history` dan `known_issue_risks` memberikan konteks historis lintas waktu, sedangkan *field* `metrics_hourly` dan `alerts_recent` menyediakan konteks terkait kondisi aktual dari data monitoring. Adapun *field* `insight_context` digunakan untuk menegaskan mode analisis dan kelas risiko sehingga model tidak mencampur dengan perlakuan antara aset monitoring dan *non*-monitoring. Prosedur pengukuran token diuraikan pada Subbab 3.8.7.

3.7.2 Metode AI untuk Assign Recommendation

Pada fitur AI *Assign Recommendation*, konteks yang disiapkan mencakup profil aset kandidat, status aset, jenis aset, serta kebutuhan atau karakteristik penggunaan yang menjadi dasar pertimbangan. LLM kemudian diminta menghasilkan rekomendasi aset yang paling sesuai beserta alasan singkat yang dapat membantu operator memahami dasar rekomendasi. Hasil yang diberikan AI tidak langsung dieksekusi sebagai keputusan final, tetapi digunakan sebagai dukungan keputusan bagi operator.

Template *prompt* pada Subbab 3.7.2 sampai 3.8.4 ditampilkan dalam bentuk ringkas dan skematik untuk menjelaskan struktur instruksi. *Prompt* aktual pada sistem dapat memuat instruksi tambahan yang bersifat teknis dan konteks-spesifik, sepanjang tidak mengubah kontrak keluaran utama. Struktur *template prompt assign* disajikan pada Gambar 3.13.

```
System:
Anda adalah analis alokasi aset TI. Gunakan hanya data yang diberikan.
Berikan rekomendasi objektif, singkat, dan dapat diaudit.

User:
Tujuan: merekomendasikan maksimal {max_recommendations} aset untuk pegawai.
employee_profile_toon:
{employee_profile_toon}

current_devices_toon:
{current_devices_toon}

available_devices_toon:
{available_devices_toon}

Aturan keluaran:
1) Hanya pilih dari available_devices_toon.
2) Prioritas 1 adalah rekomendasi terbaik.
3) Keluarkan dalam format:
recommendations[n]{asset uuid,asset type,asset model,priority,reason}
```

Gambar 3.13 *Template prompt AI Assign*.

3.7.3 Metode AI untuk *Asset Insight*

Fitur AI *Asset Insight* bertujuan menghasilkan narasi analitis yang menjelaskan kondisi aset secara ringkas namun informatif, mencakup ringkasan kondisi terkini, prediksi pemeliharaan, dan rekomendasi tindakan. Narasi ini dirancang untuk membantu operator memahami keadaan aset, faktor yang perlu diperhatikan, serta kemungkinan tindak lanjut awal.

Konteks analisis pada fitur ini disesuaikan dengan ketersediaan data monitoring. Untuk aset yang dilengkapi data monitoring, sistem memanfaatkan metrik operasional terkini sebagai basis analisis. Sebaliknya, untuk aset tanpa data monitoring langsung, analisis lebih bertumpu pada data identitas aset, histori isu, dan riwayat pemeliharaan. Pendekatan adaptif ini memungkinkan sistem menghasilkan *insight* yang relevan baik untuk aset dengan maupun tanpa kemampuan monitoring.

Keluaran *insight* disusun dalam struktur terstruktur yang mencakup tiga komponen utama yaitu ringkasan kondisi terkini pada *field* `current_condition`, prediksi pemeliharaan `predictive_maintenance`, dan serangkaian rekomendasi tindakan `recommendations`. Untuk aset tanpa data monitoring, sistem tidak menghasilkan referensi metrik numerik seperti penggunaan CPU atau memori, melainkan berfokus pada informasi historis dan kontekstual yang tersedia. Struktur template prompt *insight* ditampilkan pada Gambar 3.14.

```
System:
Anda adalah asisten analisis kondisi aset TI.
Susun ringkasan kondisi saat ini, prediksi pemeliharaan, dan rekomendasi
tindakan.

User:
mode: {monitoring|non_monitoring}
asset_uuid: {asset_uuid}
context_toon:
{context_toon}

Aturan keluaran:
insight:
  current_condition: <ringkasan utama>
  predictive_maintenance: <prediksi / boleh kosong jika data kurang>
recommendations[n]{text}:
  "<aksi 1>"
  "<aksi 2>"
  ...

Constraint:
- Jika mode=non_monitoring dan tidak ada metrics, jangan menyebut
CPU/RAM/Disk/Net sebagai fakta numerik.
```

Gambar 3.14 *Template prompt AI Insight.*

3.7.4 Metode AI untuk *Health Evaluation*

Fitur AI *Health Evaluation* bertujuan mengevaluasi dan memperbarui status kesehatan aset secara berkala melalui *pipeline* internal. Berbeda dengan AI *Assign* dan AI *Insight* yang

dipicu oleh permintaan pengguna, fitur ini beroperasi sebagai proses otomatis yang menghasilkan *snapshot* kesehatan aset untuk mendukung pemantauan kondisi secara proaktif.

Pipeline evaluasi kesehatan menggabungkan dua pendekatan komplementer untuk menangkap dimensi kondisi aset yang berbeda. Evaluasi operasional berbasis aturan membaca *matriks* terkini CPU, RAM, *disk*, dan *alert* untuk mendeteksi anomali jangka pendek dan menghasilkan status kesehatan secara langsung. Evaluasi berbasis LLM menganalisis konteks siklus hidup *asset* meliputi histori isu, pemeliharaan, spesifikasi, dan usia komponen untuk menghasilkan skor numerik yang kemudian dikonversi ke status. Pendekatan ganda ini dipilih karena metrik operasional saja tidak cukup untuk menangkap degradasi *gradual*, sedangkan analisis LLM saja tidak cukup responsif terhadap lonjakan beban mendadak.

Sistem menerapkan perlakuan berbeda berdasarkan ketersediaan data metrik. Aset risiko tinggi dilengkapi data metrik operasional real-time sehingga evaluasi operasional dapat dijalankan sepenuhnya. Sebaliknya, aset risiko rendah tidak memiliki data metrik sehingga hanya evaluasi berbasis LLM yang aktif. Stratifikasi ini mencerminkan prinsip manajemen aset berbasis risiko yang berakar pada perbedaan konsekuensi operasional antara aset CAPEX dan OPEX sebagaimana diuraikan pada Subbab 2.1.3 bahwa aset bernilai strategis dengan dampak operasional lebih besar mendapat evaluasi lebih ketat dan berlapis. Pada saat yang sama, stratifikasi ini merupakan penerapan konkret dari spektrum *human-in-the-loop* yang dibahas pada Subbab 2.7.1 bahwa aset kelas risiko tinggi ditempatkan pada ujung spektrum yang lebih ketat melalui *guardrail override*, sedangkan aset kelas risiko rendah diberi ruang interpretasi model yang lebih besar dengan pengawasan berkala.

Untuk aset kelas risiko tinggi, evaluasi operasional membaca rata-rata metrik per jam dari histori tujuh hari terakhir dan menerapkan aturan ambang sebagaimana dirangkum pada Tabel 3.11. Evaluasi ini menghasilkan status kesehatan secara langsung bukan skor numerik berdasarkan *threshold* metrik dan kondisi *alert.*

Tabel 3.11 Ketenuan ambang kesehatan operasional berbasis metri

Metrik	**Ambang *WARNING***	**Ambang *CRITICAL***
CPU rata-rata per jam	≥ 75%	≥ 90%
RAM rata-rata per jam	≥ 75%	≥ 90%
Disk rata-rata per jam	≥ 85%	≥ 95%

Selain ambang metrik, dua aturan menjaga stabilitas status operasional. Mekanisme *anti-flapping* mencegah eskalasi akibat lonjakan sesaat, status `*CRITICAL*` hanya ditetapkan apabila dua sampel terakhir secara berturut-turut melampaui ambang kritis, sehingga anomali satu jam tidak langsung memicu status tersebut. Eskalasi berbasis alert menaikkan status sesuai durasi dan *severity*, jika status *alert* yang masih *ongoing* berdurasi minimal satu jam memicu *critical*, sedangkan *alert* berdurasi kurang dari satu jam maupun *alert warning* hanya memicu status `*WARNING*`.

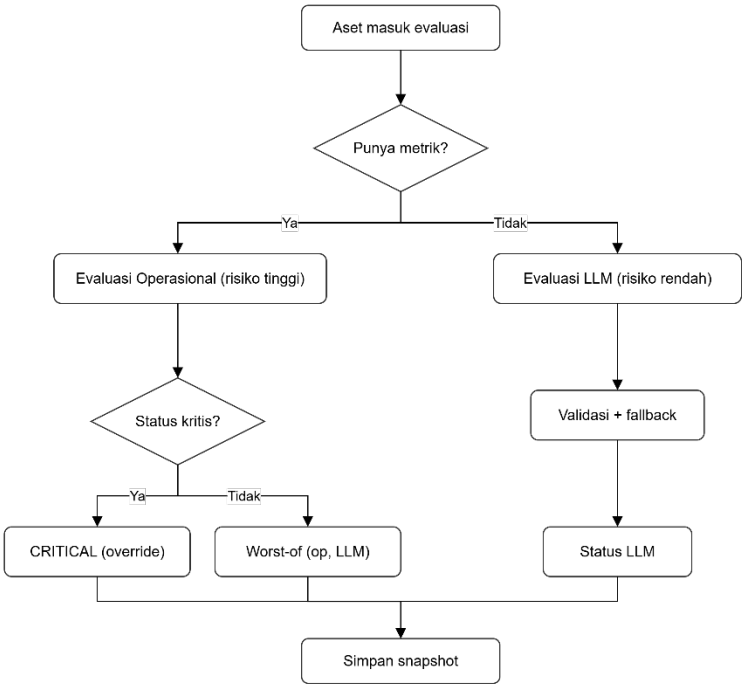
Evaluasi berbasis LLM menghasilkan *lifecycle score* dalam rentang 0-100 melalui penilaian empat dimensi risiko 'AlertRisk (bobot 40%)', 'TrendRisk (30%)', 'ReliabilityEventRisk (20%)', dan 'AssetContextRisk (10%)'.

Skor dihitung dengan formula:

$$\text{lifecycle_score} = 100 - \text{GlobalRisk} \quad (3.1)$$

Lalu dikonversi ke status kesehatan. Evaluasi LLM aktif untuk kedua kelas risiko, namun perannya lebih dominan pada aset kelas risiko rendah karena tidak ada evaluasi operasional yang memvalidasi.

Logika penggabungan dan pengendalian pipeline dirangkum pada Gambar 3.15. Pada aset kelas risiko tinggi, kondisi operasional *critical* mengaktifkan *hard override* sehingga status akhir langsung ditetapkan `*CRITICAL*` dan keluaran LLM diabaikan, sehingga anomali yang terdeteksi secara deterministik tidak dapat ditutupi oleh penilaian model. Apabila *override* tidak terpicu, status akhir ditentukan dari tingkat keparahan tertinggi antara evaluasi operasional dan konversi skor LLM. Pada aset risiko rendah, status keluaran LLM divalidasi terhadap referensi status domain pada masterdata sistem dan status yang tidak dikenali akan ditolak. Apabila LLM mengembalikan teks naratif dan bukan status terstruktur, sistem memetakan teks tersebut ke status melalui pemindaian kata kunci. Seluruh mekanisme pengendalian bersifat deterministik dengan prioritas tetap, yaitu aturan memegang keputusan akhir pada kondisi kritis dan LLM hanya dapat mendorong status ke arah yang lebih waspada, bukan menurunkannya.



Gambar 3.15 Alur keputusan evaluasi kesehatan aset dengan *guardrail* berbasis aturan

Ketika layanan LLM tidak tersedia setelah mekanisme retry, pipeline menjalankan fallback lokal agar evaluasi tidak terhenti. Pada aset kelas risiko tinggi, fallback terutama menggunakan evaluasi operasional berbasis metrik dan alert. Pada aset kelas risiko rendah, fallback membaca lifecycle_status atau lifecycle_summary pada payload jika tersedia; apabila kedua konteks tersebut kosong, sistem menggunakan hasil evaluasi operasional yang tersedia secara konservatif. Hasil fallback dinormalisasi ke status domain yang sah, disimpan sebagai snapshot baru dengan nilai confidence 0, dan diberi faktor penanda fallback agar sumber hasil dapat ditelusuri. Karena konteks alternatif pada aset non-monitoring lebih terbatas, hasil fallback tetap diposisikan sebagai indikasi awal yang memerlukan verifikasi operator.

Pipeline menerima konteks dalam format TOON yang mencakup data identitas aset, histori isu, histori pemeliharaan, dan data metrik terkini. Keluaran pipeline berupa *snapshot* kesehatan yang disimpan ke basis data dan digunakan untuk memperbarui status aset pada sistem inti. *Template prompt* pada Gambar 3.16 mengilustrasikan struktur input dan output untuk kedua kelas risiko.

Template prompt health (struktur):

```
System:
Anda adalah evaluator kesehatan aset TI.
Gunakan konteks yang tersedia dan keluarkan hasil terstruktur.

User:
type: {high|low}
asset_uuid: {asset_uuid}
context_toon:
{context_toon}

Aturan keluaran:
- Jika type=high:
  {
    "lifecycle_score": <0..100>,
    "confidence": <0..100>,
    "top_factors": ["...", "...", "..."]
  }
- Jika type=low:
  {
    "final_health_status": "HEALTHY|WARNING|CRITICAL",
    "final_health_score": <0..100>,
    "confidence": <0..100>,
    "top_factors": ["...", "...", "..."]
  }
```

Gambar 3.16 *Template prompt AI Health.*

3.7.5 Mekanisme Pengendalian: *Retry, Fallback*, dan Validasi Output

Model pengambilan keputusan pada penelitian ini difokuskan pada AI *Health Evaluation* karena fitur tersebut menghasilkan status kesehatan aset yang digunakan sebagai dasar tindak lanjut operasional. Model ini tidak dimaksudkan sebagai pengambil keputusan otomatis, melainkan sebagai mekanisme pendukung keputusan yang mengolah data aset, histori operasional, data monitoring, keluaran LLM, dan aturan domain untuk menghasilkan status awal yang dapat diverifikasi operator.

Tabel 3. 12 Komponen model pengambilan keputusan AI *Health Evaluation*

Komponen	Penjabaran
Tujuan keputusan	Menentukan status kesehatan aset sebagai dasar prioritas tindak lanjut operasional
Alternatif keputusan	*HEALTHY, WARNING*, dan *CRITICAL*

Komponen	Penjabaran
Input keputusan	Identitas aset, kelas risiko, histori *issue*, histori *maintenance*, metrik monitoring, *alert*, dan keluaran LLM
Kriteria keputusan	Ambang metrik, *severity alert*, *lifecycle score*, histori gangguan, dan validitas status pada masterdata
Metode	Kombinasi LLM untuk interpretasi konteks *lifecycle* dan *rule-based guardrail* untuk kondisi operasional kritis
Output sistem	*operational_health, lifecycle_health, final_health, confidence*, faktor dominan, *trigger* evaluasi, waktu pembentukan, dan *health snapshot*
Pengambil keputusan akhir	Operator atau personel domain yang memverifikasi hasil sebelum tindakan lapangan dilakukan

Keluaran model pada jalur LLM dapat memuat skor antara seperti *lifecycle_score* atau *final_health_score* untuk membantu pemetaan status. Namun, data yang disimpan sebagai *snapshot* kesehatan pada *backend* adalah status operasional, status *lifecycle*, status akhir, *confidence,* faktor dominan, *trigger* evaluasi, dan waktu pembentukan. Dengan demikian, skor LLM diperlakukan sebagai nilai antara dalam *pipeline*, bukan sebagai satu-satunya dasar keputusan akhir.

Aturan keputusan utama pada model ini disusun untuk menjaga agar keluaran LLM tidak menurunkan tingkat kewaspadaan pada kondisi yang sudah terdeteksi kritis secara deterministik. Ringkasan aturan keputusan disajikan pada Tabel 3. 13.

Tabel 3. 13 Aturan keputusan AI *Health Evaluation*

Kondisi	Perlakuan keputusan
Aset *high-risk* dengan metrik atau *alert* kritis	*Guardrail* menetapkan status akhir *CRITICAL*
Aset *high-risk* tanpa kondisi kritis	Status akhir mengikuti tingkat keparahan tertinggi antara *operational health* dan *lifecycle health*
Aset *low-risk* dengan keluaran LLM valid	Status akhir mengikuti keluaran LLM setelah divalidasi terhadap masterdata
Aset *low-risk* dengan keluaran LLM tidak valid atau *service* LLM gagal	Sistem menjalankan fallback lokal, lalu menormalisasi hasil ke referensi status domain yang sah
Seluruh kelas aset	Output sistem digunakan sebagai dukungan keputusan dan tetap diverifikasi operator

Prioritas tingkat keparahan status digunakan agar penggabungan hasil evaluasi dapat dijalankan secara konsisten. Pada penelitian ini, status kesehatan dipetakan ke peringkat *severity* sebagai berikut:

$$\text{rank(HEALTHY)} = 1, \text{rank(WARNING)} = 2, \text{rank(CRITICAL)} = 3 \quad (3.2)$$

Berdasarkan pemeringkatan tersebut, untuk aset *high-risk* yang tidak terkena *hard override*, status akhir ditentukan menggunakan tingkat keparahan tertinggi antara hasil evaluasi operasional dan hasil evaluasi *lifecycle*:

$$\text{final_health} = \text{max_severity}(\text{operational_health}, \text{lifecycle_health}) \quad (3.3)$$

Untuk aset *low-risk*, status akhir mengikuti *lifecycle_health* setelah nilai keluaran dinormalisasi ke status yang dikenali dan divalidasi terhadap referensi status pada masterdata sistem. Jika pemanggilan LLM gagal, sistem menjalankan *fallback* lokal terbatas berdasarkan status atau ringkasan *lifecycle* yang tersedia pada payload. Apabila *payload* tersebut tidak tersedia, *fallback* lokal menggunakan evaluasi operasional yang pada aset *low-risk* umumnya tidak memiliki metrik monitoring, sehingga hasilnya cenderung konservatif dan tetap perlu diverifikasi operator.

Dengan model tersebut, alur pengambilan keputusan bersifat proporsional terhadap kelas risiko aset. Aset *high-risk* memperoleh kontrol lebih ketat karena memiliki sinyal monitoring dan konsekuensi gangguan yang lebih besar, sedangkan aset *low-risk* tetap memperoleh evaluasi berbasis histori dengan validasi status domain. Posisi ini menjaga karakter sistem sebagai *Decision Support System* berbasis *human-in-the-loop*, bukan sistem otomasi keputusan penuh.

3.7.6 Mekanisme Pengendalian: *Retry*, *Fallback*, dan Validasi Output

Integrasi dengan service LLM eksternal menghadapi masalah ketidakpastian operasional yang perlu dikelola secara eksplisit. Keterlambatan respons, kegagalan sementara, dan keluaran yang tidak sesuai format dapat mengganggu alur bisnis jika tidak ditangani dengan mekanisme pengendalian yang tepat. Penelitian ini menerapkan tiga layer mekanisme pengendalian retry dengan *backoff* eksponensial, strategi *fallback* yang disesuaikan per-fitur, dan validasi outpu struktur. Pola-pola ketahanan *service* seperti *retry, backoff, dan fallback* dijelaskan dalam

literatur sebagai mekanisme penting untuk menjaga keandalan sistem yang bergantung pada *service* eksternal (Punithavathy & Priya, 2024).

Mekanisme *retry* diterapkan pada komunikasi antara *backend* dan LangChain *service* untuk menangani kegagalan sementara seperti *rate limiting* atau *error server*. Sistem melakukan percobaan ulang dengan jeda yang meningkat secara eksponensial untuk menghindari pembebanan berlebih pada layanan yang sedang bermasalah. *Retry* hanya dilakukan untuk kondisi yang dianggap dapat pulih, seperti *error* jaringan atau status HTTP yang menunjukkan masalah sementara. Selain mekanisme *retry*, sistem juga menerapkan *timeout* pada setiap permintaan HTTP untuk mencegah permintaan yang menggantung tanpa respons.

Strategi *fallback* dirancang berbeda untuk setiap fitur AI karena karakteristik dan kebutuhan operasional yang berbeda. Untuk fitur AI *Asset Insight*, ketika pemanggilan model gagal, sistem mengembalikan pesan *fallback* yang informatif agar operator tetap memperoleh panduan sementara. Untuk fitur AI *Health Evaluation*, *fallback* dilakukan dengan menggunakan evaluasi operasional berbasis aturan sehingga *pipeline* evaluasi tidak terhenti meskipun model tidak tersedia. Untuk fitur AI *Assign Recommendation*, sistem tidak menyediakan *fallback* karena rekomendasi alokasi memerlukan analisis kontekstual yang tidak dapat digantikan oleh aturan sederhana.

Validasi keluaran diterapkan untuk memastikan bahwa respons dari LLM memenuhi struktur yang diharapkan sebelum digunakan oleh sistem. *Field* yang diperlukan diperiksa keberadaannya, tipe data divalidasi, dan nilai dinormalisasi jika diperlukan. Sebagai contoh, untuk keluaran health *evaluation*, status kesehatan dinormalisasi ke salah satu dari tiga nilai yang valid. Jika status yang dihasilkan model tidak sesuai, sistem menggunakan *mapping* dari skor numerik untuk menentukan status yang tepat. Untuk keluaran yang tidak dapat diparsing atau tidak memenuhi skema validasi, sistem menerapkan strategi *fallback* yang telah dijelaskan sebelumnya.

3.7.7 Pemilihan Model *Large Language Model*

Penelitian ini menggunakan arsitektur pemanggilan model tidak langsung untuk menjaga pemisahan tanggung jawab antara logika bisnis dan inferensi model bahasa. *Backend* utama tidak berkomunikasi langsung dengan provider model eksternal, melainkan melalui LangChain AI *service* internal yang bertindak sebagai perantara. Arsitektur ini memungkinkan pengaturan

akses, autentikasi, *routing*, dan pengendalian permintaan dipusatkan pada satu layanan, sehingga backend inti tetap independen terhadap perubahan provider atau konfigurasi model.

Model yang digunakan adalah OpenAI GPT-5.5, pemilihan model ini didasarkan pada pertimbangan operasional, bukan untuk tujuan komparasi antarmodel. Model ini berfungsi sebagai komponen inferensi utama yang digunakan secara konsisten selama implementasi dan evaluasi artefak.

Pertimbangan pemilihan model mencakup kemampuan *reasoning* kontekstual untuk membaca kombinasi data aset, histori operasional, dan konteks monitoring. Kemampuan menghasilkan keluaran yang dapat diarahkan ke struktur respons sistem melalui instruksi dan validasi output. Serta integrasi yang memungkinkan pengendalian pola pemanggilan melalui mekanisme *retry, throttling, timeout*, dan *fallback*.

Tabel 3.14 Konfigurasi integrasi LLM pada penelitian

Parameter	**Nilai**
Proxy LLM	9Router (kompatibel format OpenAI)
Model	cx/gpt-5.5
Temperature	0.2
Top-p	0.8
Format konteks	TOON
Mekanisme pengendalian	*Retry, timeout, throttling, fallback*

Konfigurasi parameter *sampling* seperti *temperature* dan top-p ditetapkan untuk menjaga keseimbangan antara konsistensi struktur keluaran dan fleksibilitas naratif. Mekanisme pengendalian seperti *retry*, *timeout*, dan *throttling* diterapkan karena endpoint AI bergantung pada layanan model eksternal melalui *proxy*. Karena penelitian ini bukan komparasi antarmodel, evaluasi difokuskan pada kestabilan kontrak integrasi, validitas struktur output, mekanisme *fallback*, dan penilaian domain terhadap keluaran model.

Untuk menjaga konsistensi selama periode pengujian, seluruh permintaan model menggunakan alias konfigurasi yang sama pada 9Router. Konsistensi ini dipahami sebagai konfigurasi *request* yang dikirim oleh LangChain *service*, bukan sebagai jaminan bahwa versi model aktual pada provider di balik *proxy* selalu dapat diverifikasi secara langsung.

3.7.8 Validasi Keluaran AI dan Penanganan Keluaran Tidak Valid

Validasi keluaran AI dilakukan sebelum output dipakai pada alur operasional untuk menjaga agar output dapat diproses sistem, tidak melanggar *contract*, dan tidak langsung memengaruhi status operasional bila format tidak valid. Setiap fitur AI memiliki kriteria validasi yang disesuaikan dengan karakteristik keluarannya. AI *Assign* memeriksa keterbacaan struktur rekomendasi dan *field* utama seperti identitas aset dan prioritas. AI *Insight* mewajibkan keberadaan ringkasan kondisi terkini dan menormalisasi daftar rekomendasi. AI *Health* memeriksa status kesehatan, skor numerik, tingkat kepercayaan, dan faktor-faktor utama, serta memvalidasi kesesuaian status ke referensi domain.

Apabila keluaran AI tidak memenuhi ketentuan validasi, sistem menerapkan mekanisme penanganan bertahap. Sistem terlebih dahulu menjalankan *retry* terbatas sesuai konfigurasi. Jika tetap gagal, sistem menjalankan *fallback* sesuai jenis fitur. Untuk AI *Insight*, sistem mencoba *endpoint* spesifik terlebih dahulu dan melakukan *fallback* ke *endpoint legacy* hanya jika *endpoint* spesifik tidak tersedia. Untuk AI *Health*, sistem menggunakan *fallback* evaluasi lokal berbasis aturan agar pipeline tetap berjalan. Untuk AI *Assign*, tidak tersedia mekanisme *fallback* karena rekomendasi alokasi memerlukan analisis kontekstual yang tidak dapat digantikan oleh aturan sederhana. Jika keluaran tetap tidak valid dan tidak ada *fallback* yang sah, hasil ditolak dan tidak dipakai untuk memperbarui status operasional.

3.8 Metode Pengujian dan Evaluasi

Pengujian dan evaluasi dilakukan untuk mengetahui apakah sistem yang dibangun telah memenuhi fungsi yang diharapkan, mampu berintegrasi antarkomponen, dan andal untuk digunakan pada skenario operasional yang relevan. Pengujian dan evaluasi dilakukan melalui lima kelompok utama, yaitu pengujian fungsional, pengujian integrasi, pengujian performa dan keandalan, pengujian skenario bisnis, serta evaluasi kualitas output AI berbasis penilaian operator. Instrumen *feedback* operator disiapkan sebagai evaluasi pelengkap, dan hasil utama penelitian tidak bergantung pada ketersediaan data tersebut. Pada implementasi, pengujian dilakukan terutama melalui skenario manual terarah dan observasi hasil eksekusi sistem.

Data uji pada bab ini menggunakan kombinasi data operasional nyata pada lingkungan studi kasus dan data sintetis terbatas untuk skenario batas. Data sintetis digunakan terutama pada kondisi seperti aset tidak *borrowable*, *alert* tanpa *asset_uuid,* serta simulasi kegagalan *endpoint* AI untuk pengujian *fallback*.

3.8.1 Pengujian Fungsional

Pengujian fungsional dilakukan dengan pendekatan *black-box testing* (Ayuningtyas et al., 2023) pada modul-modul utama sistem. Pengujian ini berfokus pada kesesuaian antara masukan, proses, dan keluaran tanpa meninjau detail implementasi internal. Modul yang diuji meliputi pengelolaan aset, peminjaman dan pengembalian, pencatatan isu, pemeliharaan, monitoring, serta fitur AI. Pada praktiknya, pengujian dilakukan terutama melalui skenario manual yang terstruktur. Tujuan pengujian ini adalah memastikan bahwa setiap fungsi inti sistem berjalan sesuai kebutuhan yang telah dirumuskan.

3.8.2 Pengujian Integrasi

Pengujian integrasi dilakukan untuk menilai kelancaran komunikasi antarlayanan, khususnya antara *backend* dan LangChain *service*, karena sistem dibangun dalam komponen yang terpisah. Melalui pengujian ini, dapat diketahui apakah data konteks berbasis TOON terkirim dengan benar di dalam *payload* layanan, apakah LangChain *service* mengembalikan hasil sesuai format yang diharapkan, serta apakah *backend* mampu memproses hasil tersebut sebelum diteruskan ke pengguna atau disimpan untuk kebutuhan internal.

3.8.3 Pengujian Performa dan Keandalan

Pengujian performa dan keandalan merupakan rancangan evaluasi terstruktur terhadap perilaku sistem saat berinteraksi dengan layanan AI eksternal. Fokusnya bukan hanya kecepatan, tetapi juga ketahanan layanan pada kondisi gangguan.

Load test dengan konkurensi terbatas dirancang untuk mengamati perilaku tiap fitur AI secara terpisah pada tingkat beban yang berbeda. Metrik yang diamati meliputi latensi, *error rate*, dan *throughput*. Pengujian dijalankan pada lingkungan pengembangan dengan spesifikasi yang dicantumkan pada BAB IV sesuai kondisi aktual saat pelaksanaan.

Selain itu, *fault injection* dirancang untuk mensimulasikan skenario kegagalan layanan AI guna menguji pola *retry*, *exponential backoff*, dan *fallback* yang telah diimplementasikan pada sistem (Punithavathy & Priya, 2024). Skenario kegagalan yang disimulasikan mencakup *error server* dan *rate limiting* menggunakan *mock proxy* yang menggantikan *endpoint* 9Router. Setiap skenario dieksekusi pada tiga fitur AI dengan beberapa iterasi, menghasilkan total 45 uji. Verifikasi dilakukan melalui pencatatan jumlah panggilan *mock proxy* sebagai indikator aktivasi *retry*, status respons, strategi *fallback*, serta ketiadaan *fatal error* pada proses utama. Hasil eksekusi dilaporkan pada BAB IV Subbab 4.5.1.

Untuk pengukuran waktu seleksi pada fitur AI *Assign Recommendation*, eksperimen dirancang sebagai *paired comparison* dengan dua kondisi, kondisi *baseline* tanpa rekomendasi AI dan kondisi AI-*assisted* dengan tambahan prioritas dan alasan dari AI. *Dataset* kandidat aset dibuat identik pada kedua kondisi untuk memastikan perbandingan yang adil. Namun, karena keterbatasan kondisi operasional dan akses operator pada periode penelitian, pengukuran terstruktur ini belum dieksekusi dalam lingkup penelitian ini.

3.8.4 Pengujian Skenario Bisnis

Selain pengujian teknis, penelitian ini menggunakan pengujian berbasis skenario bisnis (*Use Case*). Skenario bisnis digunakan untuk memastikan bahwa sistem tidak hanya berjalan secara teknis, tetapi juga relevan dengan proses kerja pengelolaan aset TI pascaakuisisi. Skenario yang diuji mencakup modul inti aset dan penggunaan, AI *Assign*, AI *Insight*, AI *Health*, serta integrasi monitoring dan *alert*.

Di samping skenario bisnis *end-to-end*, penelitian ini menetapkan 26 kasus evaluasi kualitas keluaran AI. Kasus tersebut disusun menggunakan *purposive sampling* (Nyimbili & Nyimbili, 2024) berdasarkan variasi domain fitur, jenis aset, kondisi data, dan peran evaluator. Desain ini tidak diarahkan untuk generalisasi statistik populasi, melainkan untuk mengevaluasi artefak pada konteks operasional yang representatif sesuai pendekatan *Design Science Research* (DSR). Rincian 26 kasus tersebut menjadi dasar evaluasi kualitas keluaran AI yang dijelaskan pada Subbab 3.8.5. Rancangan ini menggunakan seluruh aset dan persona final, yaitu 4 aset *monitorable*, 6 aset *non-monitorable*, dan 6 persona pengguna.

Tabel 3.15 Komposisi kasus evaluasi kualitas keluaran AI

Group	Pasangan evaluator	Fitur	Jumlah kasus	Konteks utama
A	AMP dan NDC	AI *Health monitoring*	4	*Router, firewall,* server dengan data monitoring CPU/RAM/*network*
A	AMP dan NDC	AI *Insight monitoring*	4	*Insight* aset *monitorable* berbasis *metric* dan *alert*
B	AMP dan Inventaris	AI *Health non-monitoring*	6	Laptop berdasarkan riwayat *issue, maintenance*, dan kondisi inventaris
B	AMP dan Inventaris	AI *Insight non-monitoring*	6	*Insight lifecycle* laptop tanpa metrik monitoring
B	AMP dan Inventaris	AI *Assign*	6	Persona pengguna dan *pool* laptop tersedia

Group	Pasangan evaluator	Fitur	Jumlah kasus	Konteks utama
Total			26	

3.8.5 Evaluasi Keluaran AI Berbasis *Domain-Based Expert Evaluation*

Evaluasi kualitas keluaran AI menggunakan pendekatan *Domain-Based Expert Evaluation*, yaitu penilaian keluaran AI oleh operator yang memiliki relevansi domain terhadap fitur yang dinilai. Pendekatan ini dipilih karena sistem berfungsi sebagai pendukung keputusan (*decision support*), sehingga kualitas keluaran tidak cukup dinilai dari format atau keberhasilan teknis saja, tetapi juga dari kesesuaiannya terhadap konteks operasional.

Tiga evaluator dipilih berdasarkan peran operasional di BSI UII, dengan profil masing-masing disajikan pada Tabel 3.16.

Tabel 3.16 Profil evaluator penelitian

No	Peran	Perspektif operasional	Relevansi utama
1	Kepala Tim AMP	Kebijakan aset, prioritas organisasi, alokasi perangkat	Semua fitur sebagai perspektif pengelolaan aset
2	Staff NDC	Monitoring, *alert*, kondisi jaringan dan infrastruktur	AI *Health/Insight* untuk aset monitorable
3	Staff inventaris	Inventaris fisik, peminjaman, pengembalian aset	AI *Assign* dan evaluasi aset non-monitoring

Berbeda dari rancangan awal yang mempertimbangkan seluruh rater menilai seluruh kasus, desain final tidak memaksa setiap evaluator menilai seluruh fitur karena tidak semua fitur berada dalam kapasitas domain setiap penilai. Sebagai contoh, Staff NDC tidak menjadi penilai untuk AI *Assign* karena peminjaman dan alokasi laptop bukan domain operasional NDC. Oleh karena itu, pasangan evaluator ditetapkan berdasarkan relevansi domain fitur. Pemetaan pasangan evaluator per fitur disajikan pada Tabel 3.17.

Tabel 3.17 Pemetaan evaluator berbasis domain

Group	Fitur	Kasus	Evaluator	Alasan pemilihan
A	AI *Health monitoring*	HLT-A1 s.d. HLT-A4	AMP dan NDC	NDC memahami monitoring/*alert,* AMP memahami kebijakan kondisi aset

Group	Fitur	Kasus	Evaluator	Alasan pemilihan
A	AI *Insight monitoring*	INS-A1 s.d. INS-A4	AMP dan NDC	*Insight* aset jaringan/infrastruktur membutuhkan perspektif monitoring dan pengelolaan aset
B	AI Health non-monitoring	HLT-B1 s.d. HLT-B6	AMP dan Inventaris	Kondisi laptop dinilai dari riwayat aset, *issue*, dan kesiapan peminjaman
B	AI I*nsight non-monitoring*	INS-B1 s.d. INS-B6	AMP dan Inventaris	*Insight lifecycle* laptop relevan dengan pengelolaan inventaris dan pemeliharaan
B	AI *Assign*	ASN-B1 s.d. ASN-B6	AMP dan Inventaris	Rekomendasi alokasi laptop membutuhkan perspektif kebijakan aset dan operasional peminjaman

Kategori penilaian yang digunakan adalah:

a. 2 – **Sesuai**. keluaran AI sesuai dengan kondisi kasus dan dapat diterima secara operasional.
b. 1 - **Sebagian sesuai**. keluaran AI masih relevan, tetapi memiliki kekurangan minor, kurang lengkap, atau perlu klarifikasi.
c. 0 - **Tidak sesuai**. keluaran AI tidak relevan, keliru, atau berpotensi menyesatkan keputusan.

Setiap kasus dinilai pada tiga aspek:

1. Sesuai kondisi kasus, yaitu apakah status*, insight*, atau rekomendasi AI sesuai dengan kondisi aset/pengguna yang diuji.
2. Alasan relevan, yaitu apakah alasan AI berhubungan langsung dengan data input, riwayat aset, metrik, *alert,* atau kebutuhan pengguna.
3. Keakuratan klaim, yaitu apakah AI menghindari klaim yang tidak didukung data, misalnya menyimpulkan kondisi *disk* ketika data *disk* tidak tersedia.

Kondisi kasus yang diberikan kepada evaluator ditulis sebagai kondisi *as-is* berdasarkan data input, seperti riwayat kerusakan, *issue* yang belum selesai, catatan *maintenance,* metrik, *alert* sintetis, dan kebutuhan pengguna. Label output AI seperti status kesehat tidak diperlakukan sebagai kondisi awal, melainkan sebagai keluaran AI atau *expected result* yang dinilai.

Perlu dinyatakan secara eksplisit bahwa sebagian evaluator domain yang dipilih pada subbab ini merupakan personel yang sama dengan tiga informan wawancara pada BAB I yang

digunakan untuk membangun urgensi penelitian. Posisi ganda ini berpotensi menimbulkan ancaman validitas internal berupa bias kesepakatan dan *social desirability*, karena evaluator sudah familiar dengan peneliti serta memiliki kepentingan terhadap keberhasilan proyek. Mitigasi yang diterapkan mencakup penyajian kondisi kasus secara *as-is* berdasarkan data input, kriteria penilaian yang berfokus pada kesesuaian operasional terhadap data dan bukan pada preferensi terhadap keluaran AI, kewajiban mencatat alasan pada pemberian skor 0 atau 1, serta penetapan pasangan evaluator berdasarkan relevansi domain. Meskipun demikian, bias ini tidak dapat dieliminasi sepenuhnya dan diakui secara jujur sebagai konsekuensi dari keterbatasan akses evaluator pada studi kasus tunggal di BSI UII.

Output AI representatif dikumpulkan melalui eksekusi API untuk 26 kasus final. Untuk AI *Insight* dan AI *Assign*, *endpoint* operasional sistem digunakan secara langsung. Untuk AI *Health*, karena *pipeline* kesehatan aset pada sistem normalnya dijalankan oleh job terjadwal, penelitian ini menggunakan *endpoint* testing internal khusus yang memanggil *pipeline health evaluation* yang sama dengan alur sistem.

Endpoint testing tersebut hanya dipakai untuk pengambilan output penelitian dan berfungsi sebagai *testing rig* untuk reproduksibilitas pengujian, bukan sebagai fitur operasional pengguna akhir. Mekanisme akses *endpoint testing* dikendalikan secara lokal selama pengambilan data dan tidak diekspos pada konfigurasi operasional.

Untuk mengukur kesepakatan antarpenilai, penelitian ini menggunakan *domain-based expert agreement evaluation* pada pasangan evaluator yang menilai himpunan kasus yang sama. Ukuran yang dilaporkan adalah *percent agreement* per fitur dan per aspek penilaian, distribusi skor, serta analisis naratif terhadap perbedaan penilaian. *Percent agreement* tidak digunakan sendirian, tetapi dibaca bersama distribusi skor dan catatan penilai agar sumber perbedaan lebih mudah ditelusuri. Hasil kesepakatan tidak dibaca sebagai reliabilitas populasi, melainkan sebagai indikator kesesuaian penilaian pada pasangan evaluator yang relevan secara domain dalam konteks studi kasus BSI UII.

Selain evaluasi *rater*, penelitian ini juga menetapkan metrik kuantitatif konsistensi output AI pada input identik. Metrik dibedakan per fitur karena bentuk keluaran berbeda:

a. AI *Assign Recommendation*. *Overlap Coefficient* pada himpunan UUID aset top-3 antareksekusi.
b. AI *Asset Insight*. Jaccard *Similarity* pada himpunan kata kunci insight setelah *preprocessing* sederhana, seperti *lowercasing*, penghapusan *stopword* umum, dan normalisasi istilah teknis

(Jalal et al., 2022). Konsep Jaccard Similarity dan rasional pemilihannya untuk narasi LLM dijelaskan pada Subbab 2.8.1.

c. AI *Health Evaluation*. proporsi status final *health* yang identik antareksekusi, serta selisih absolut untuk skor numerik.

Pengujian konsistensi dilakukan pada 26 kasus final dengan tiga eksekusi ulang per kasus, sehingga menghasilkan 78 eksekusi. Eksekusi dilakukan secara berurutan dengan jeda antarpemanggilan untuk mengurangi risiko *rate limiting* pada layanan model dan efek *cache* yang dapat memengaruhi variabilitas keluaran. Hasil pelaksanaan dilaporkan pada BAB IV.

Dengan desain *domain-based expert agreement evaluation*, hasil evaluasi merupakan validasi operasional terbatas terhadap keluaran AI. Hasil ini tidak diklaim sebagai reliabilitas tingkat populasi karena jumlah evaluator, jumlah kasus, dan konteks organisasi masih terbatas pada studi kasus BSI UII.

3.8.6 Evaluasi DSS Berbasis *Feedback* Operator

Untuk memastikan bahwa evaluasi tidak berhenti pada aspek fungsional teknis, penelitian ini menambahkan evaluasi dukungan keputusan melalui umpan balik operator menggunakan skala *Likert*. Konsep skala *Likert* dan posisinya sebagai instrumen pelengkap evaluasi DSS dijelaskan pada Subbab 2.8. Instrumen ini dipakai untuk menilai apakah sistem membantu keputusan operasional secara nyata pada aspek efisiensi, keterpahaman, relevansi rekomendasi, kejelasan alasan, prioritas tindakan, dan prinsip *human-in-the-loop*.

Instrumen *feedback* operator ditelaah dari sisi keterbacaan butir, relevansi dimensi DSS, dan kesesuaian konteks operasional. Dengan jumlah responden terbatas, apabila *feedback* dikumpulkan, hasil skor *Likert* dimaknai sebagai indikasi persepsi awal, bukan inferensi statistik populasi.

Pelaksanaan evaluasi DSS dilakukan setelah operator menjalankan skenario bisnis utama, sehingga penilaian diberikan berdasarkan pengalaman penggunaan aktual pada modul inti dan fitur AI. Instrumen *feedback* DSS berbeda dari *form* evaluasi kualitas keluaran AI, *feedback* DSS menilai persepsi kegunaan sistem secara umum, sedangkan evaluasi kualitas AI menilai output AI per kasus.

Feedback operator dirancang sebagai evaluasi pelengkap apabila responden telah mencoba alur sistem secara langsung. Apabila data *feedback* belum tersedia pada saat pelaporan akhir, hasil utama evaluasi dukungan keputusan dibatasi pada pengujian skenario

bisnis, validasi keluaran AI oleh evaluator domain, konsistensi output, dan pengujian teknis. Instrumen *feedback* operator yang digunakan pada penelitian ini disajikan pada Tabel 3.18.

Tabel 3.18 Instrumen *feedback* operator (Likert 1-5) untuk evaluasi DSS

Pernyataan	**Aspek**
Sistem membantu operator menemukan informasi aset yang dibutuhkan	Efisiensi
AI *Assign* membantu prioritas kandidat aset	DSS
AI *Insight* membantu memahami kondisi aset	Interpretasi
AI *Health* membantu prioritas tindak lanjut	Evaluasi risiko
Alasan rekomendasi dapat diverifikasi	*Auditability*
Keputusan akhir tetap pada operator	*Human-in-the-loop*
Sistem relevan terhadap kebutuhan operasional harian	Relevansi operasional

Tabel 3.19 Skenario uji bisnis yang dieksekusi

No.	**Skenario**	**Input uji utama**	**Kondisi batas yang diperiksa**	**Hasil yang diharapkan**
1	Modul inti aset dan penggunaan	Pembuatan aset, alokasi/pengembalian, pencatatan isu, pencatatan pemeliharaan	Aset tidak boleh dialokasikan saat status tidak *borrowable.* pengembalian menutup peminjaman aktif	Data transaksi tersimpan konsisten, histori terbentuk, dan status aset mengikuti aturan
2	AI *Assign Recommendation*	uuid_pengguna valid dan kandidat aset tersedia	Kandidat aset terbatas/tidak beragam	Sistem tetap menghasilkan minimal 1 rekomendasi terstruktur saat data valid
3	AI *Asset Insight*	*uuid_asset mode monitoring* dan *non-*monitoring	Data *metrics/alerts* kosong pada non-monitoring	Sistem mengembalikan narasi *insight* stabil tanpa *error*, dengan isi sesuai konteks tersedia
4	AI *Health Evaluation*	*Trigger pipeline health* untuk aset *high-risk* dan *low-risk*	*Endpoint* AI gagal sementara (simulasi error *transient*)	*Snapshot health* tetap terbentuk melalui *fallback*/*guardrail* dan sinkronisasi status tetap valid
5	Integrasi monitoring dan *alert*	Pengambilan metrik terjadwal dan *alert webhook* (*firing/resolved*)	*Alert* tanpa ‘asset_uuid’	*Alert* valid memicu *refresh health*, *alert* tanpa identitas aset diabaikan tanpa mematikan *pipeline*

3.8.7 Metode Pengukuran Eksploratoris Efisiensi Token

Berbekal format TOON yang telah diperkenalkan pada Subbab 3.7.1, bagian ini menyusun prosedur pengukuran komparatif eksploratoris token antara TOON dan JSON mentah.

Pengukuran dilakukan menggunakan metode perbandingan berpasangan (*paired comparison*) pada lima skenario konteks yang dipilih secara *purposive sampling* (Nyimbili & Nyimbili, 2024) untuk mewakili variasi skenario penggunaan sistem, rekomendasi alokasi perangkat (*Assign*), analisis wawasan dengan dan tanpa konteks monitoring (*Insight*), serta evaluasi kesehatan pada perangkat berisiko tinggi dan rendah (*Health Evaluation*). Setiap skenario diukur dengan tiga varian data untuk mereduksi bias kebetulan, sehingga total terdapat 15 pasang sampel yang dibandingkan. Ringkasan skenario pengukuran token disajikan pada Tabel 3.20.

Tabel 3.20 Ringkasan skenario pengukuran token

No	Skenario	Karakter data	Volume data simulasi
1	*Assign*	Profil pengguna dan kandidat aset	-
2	*Insight monitoring*	Metrik, *alert*, dan konteks aset termonitor	168 baris (resolusi per jam, 7 hari) dan 60 baris (resolusi per hari, 60 hari)
3	*Insight non-monitoring*	*Histori issue, maintenance,* dan konteks inventaris	-
4	*Health high-risk*	Metrik, *alert*, histori aset, dan faktor risiko	168 baris (resolusi per jam, 7 hari) dan 60 baris (resolusi per hari, 60 hari)
5	*Health low-risk*	Histori aset dan konteks non-monitoring	-

Penghitungan jumlah token pada masing-masing format dilakukan menggunakan *tokenizer*v standar industri ‘cl100k_base’ via pustaka tiktoken (Schmidt et al., 2024). Penggunaan *tokenizer* ini merupakan aproksimasi komparatif, dengan angka yang diukur difokuskan pada persentase reduksi relatif (*relative reduction*), bukan representasi jumlah tagihan token absolut pada model LLM tertentu. Perbandingan dibatasi pada JSON mentah dan TOON karena JSON merupakan format kontrak data utama pada pertukaran antarlayanan sistem.

Prosedur pengukuran mencakup persiapan pasangan konteks identik dalam dua format, pengukuran jumlah token pada masing-masing format, dan perhitungan persentase pengurangan token.

3.9 Indikator Keberhasilan

Indikator keberhasilan penelitian ditetapkan untuk menilai apakah sistem yang dibangun telah memenuhi tujuan penelitian. Indikator pertama adalah keterpenuhan kebutuhan sistem, yaitu ketika fungsi-fungsi utama yang dirancang dapat berjalan sesuai kebutuhan yang telah diidentifikasi pada tahap analisis. Indikator kedua adalah keberhasilan integrasi, yaitu ketika *backend*, basis data, layanan AI, dan sumber data monitoring dapat saling berkomunikasi sesuai alur yang dirancang.

Indikator ketiga adalah kesesuaian operasional keluaran AI. Pada indikator ini, keberhasilan dinilai dari kemampuan sistem menghasilkan rekomendasi, *insight*, dan evaluasi kesehatan yang sesuai struktur keluaran, dapat diproses dengan benar oleh sistem, dan dapat digunakan sebagai dukungan keputusan terbatas sesuai konteks studi kasus. Indikator keempat adalah keandalan sistem, yaitu ketika proses utama tetap dapat berjalan atau menurun secara terkendali saat layanan AI mengalami gangguan. Indikator kelima adalah kesesuaian sistem terhadap skenario bisnis yang diuji. Seluruh indikator beserta kriteria dan *threshold*-nya dirangkum pada Tabel 3.21.

Tabel 3.21 Indikator keberhasilan penelitian

Indikator	**Kriteria keberhasilan**	***Threshold***
Keterpenuhan fungsi utama	Modul inti dan modul AI berjalan sesuai kebutuhan	5/5 skenario bisnis pada **Tabel 3.19**
Integrasi sistem	Pertukaran data antarkomponen berjalan baik	Keberhasilan *request* integrasi lintas *service* ≥ 95% pada eksekusi uji
Kesesuaian operasional output AI	Hasil AI sesuai format, dapat diproses sistem, dan dapat digunakan sebagai dukungan keputusan terbatas	Struktur output valid pada ≥ 90% eksekusi uji berulang
Konsistensi output AI	Keluaran AI stabil pada input identik sesuai karakter tiap fitur	Metrik konsistensi (*Overlap Assign, Jaccard Insight*, proporsi status *Health*) dilaporkan per kasus uji ulang

Indikator	Kriteria keberhasilan	*Threshold*
Konsistensi status *health*	Status akhir tidak bertentangan dengan referensi sistem	100% hasil final_*health*_status terpetakan ke master data.status domain *asset_health*
Keandalan layanan	Sistem tetap operasional saat terjadi gangguan layanan AI	100% skenario *error* validasi memicu respons *error* yang tertangani (tidak menyebabkan *crash*), 45 uji *fault injection* 500/429/504 terverifikasi tanpa fatal *error* dengan *retry* dan *fallbac*k dua lapis aktif
Kesesuaian operasional dan dukungan keputusan	Skenario bisnis utama berjalan baik dan output AI mendukung proses keputusan operator	Skenario bisnis utama lulus, struktur output AI valid, *percent agreement* dan metrik konsistensi terdokumentasi, f*eedback* operator bersifat pelengkap/lanjutan apabila data tersedia

Verifikasi indikator ini dilakukan melalui validasi respons *error* terbatas serta *fault injection* terbatas terhadap gangguan layanan AI eksternal menggunakan *mock proxy*. Hasil *fault injecton* dilaporkan pada Tabel 4.14 dan Tabel 4.15. Verifikasi penuh pada beban produksi riil dengan variasi trafik dan gangguan berkepanjangan tetap ditunda untuk penelitian lanjutan. Keterlacakan rumusan masalah terhadap metode evaluasi dan indikator disajikan pada Tabel 3.22.

Tabel 3.22 Keterlacakan rumusan masalah, metode, dan indikator

Rumusan masalah	Metode evaluasi	Indikator	Bukti pada BAB IV
RM1 Integrasi proses pascaakuisisi	Pengujian fungsional dan skenario bisnis	5/5 skenario bisnis utama lulus	Hasil pengujian fungsional dan skenario bisnis
RM2 Dukungan keputusan LLM	*Expert agreement*, *percent agreement*, analisis *disagreemen t*, dan konsistensi output AI	Output AI valid, mayoritas skor rater berada pada kategori sesuai/sebagian sesuai, *percent agreement* terlapor, dan konsistensi output terdokumentasi	Evaluasi kualitas keluaran AI berbasis domain dan konsistensi output
RM3 Keandalan integrasi AI	Observasi *runtime* dan validasi *error* terbatas	Sistem menangani *error input*/*auth*/*endpoint* tanpa *crash*, *retry-backoff* dan fallback dua lapis terverifikasi pada 45 uji *fault injection* tanpa fatal *error*	Evaluasi performa dan keandalan layanan AI

3.10 Etika Penelitian

Penelitian ini dilaksanakan setelah memperoleh izin dari Kepala BSI UII sebagai penanggung jawab unit studi kasus. Izin tersebut mencakup penggunaan data operasional untuk kebutuhan pengembangan artefak, pengujian terbatas, dan pelaporan ilmiah dalam ruang lingkup penelitian skripsi.

Seluruh data operasional yang digunakan dalam pengujian dan publikasi dianonimkan. Identifier seperti UUID aset, nama perangkat, identitas personal, dan elemen yang dapat mengarah ke individu atau aset tertentu diganti dengan kode pada naskah dan tabel hasil. Surat keterangan penggunaan data dari BSI UII dicantumkan pada Lampiran.

Data operasional hanya digunakan untuk kebutuhan penelitian dan tidak dipublikasikan dalam bentuk yang memungkinkan identifikasi aset, pengguna, lokasi detail, alamat IP, kredensial, atau informasi sensitif organisasi. Akses terhadap data mentah dibatasi pada peneliti dan pihak yang berwenang di BSI UII.

3.11 Keterbatasan Penelitian

3.11.1 Daftar Keterbatasan

Penelitian ini memiliki beberapa keterbatasan yang perlu diperhatikan:

1. Ketersediaan data historis monitoring pada lingkungan studi kasus mungkin belum panjang atau belum merata pada seluruh aset, sehingga kualitas konteks analisis dapat berbeda antaraset.
2. Observasi monitoring NDC per jenis aset masih terbatas pada sampel aset tertentu, terutama *server* dan *access point*, sehingga belum merepresentasikan seluruh kelas aset TI.
3. Fitur AI pada sistem memiliki ketergantungan pada layanan model eksternal, sehingga performa dan stabilitas analisis dapat dipengaruhi oleh latensi, batas permintaan, atau kebijakan layanan tersebut.
4. Penelitian dilakukan pada konteks operasional yang terbatas dengan tiga evaluator pada ruang lingkup organisasi tertentu, sehingga hasil implementasi belum secara langsung mewakili kondisi organisasi lain yang memiliki struktur dan skala berbeda.
5. Cakupan dibatasi karena penelitian tidak membahas RBAC granular, *dashboard* analitik *enterprise*, arsitektur *multitenant*, maupun komparasi mendalam antaralternatif terhadap model LLM, karena fokus ditujukan pada pembangunan dan evaluasi artefak

pada konteks studi kasus. Pembatasan ini merupakan keputusan ruang lingkup, bukan kelemahan metodologis.

6. Otomatisasi pengujian pada implementasi saat ini masih terbatas, sehingga sebagian besar evaluasi tetap bertumpu pada skenario uji manual dan observasi perilaku sistem.
7. Keterbatasan lingkungan dan cakupan pengujian teknis. Pengujian performa, *fault injection*, dan unit test dilakukan pada lingkungan pengembangan atau uji terbatas, bukan pada beban produksi riil organisasi. Cakupan unit test otomatis saat ini terbatas pada sebagian logika evaluasi kesehatan aset, sehingga belum mencakup seluruh modul sistem.
8. *Guardrail* pada *pipeline health evaluation* saat ini menetapkan status `*CRITICAL*` untuk setiap *alert* `*CRITICAL*` yang masih berlangsung tanpa mempertimbangkan durasi. Kondisi ini membuat aturan *anti-flapping* berbasis durasi pada jalur evaluasi operasional efektif ditimpa oleh *guardrail* untuk aset kelas risiko tinggi. Penguatan *guardrail* dengan filter durasi dapat menjadi pekerjaan lanjutan guna memperhalus sensitivitas eskalasi status.
9. Pengukuran kuantitatif waktu seleksi pada fitur AI *Assign Recommendation* (Subbab 3.8.3) dirancang sebagai *paired comparison* dengan n=10 per kondisi, namun belum dieksekusi dalam lingkup penelitian ini karena keterbatasan akses operator pada periode penelitian. Konsekuensinya, efisiensi waktu seleksi dilaporkan sebagai indikasi kualitatif berbasis wawancara informal, bukan angka kuantitatif terukur. Verifikasi kuantitatif ditunda untuk penelitian lanjutan.

3.11.2 Ancaman Validitas dan Mitigasi

Untuk memetakan potensi ancaman terhadap validitas hasil secara lebih sistematis, Tabel 3.23 berikut merangkum identifikasi ancaman dan langkah mitigasi yang dilakukan dalam penelitian ini (Hastjarjo, 2011).

Tabel 3.23 Ancaman terhadap validitas dan mitigasi

Jenis validitas	***Threats***	**Mitigasi**
Internal	Bias penilai terhadap sistem peneliti	Evaluator berbasis domain, form penilaian terstruktur, pencatatan alasan, dan analisis *disagreement*
Internal	*Learning effect* dan *fatigue effect* pada	Pengukuran dirancang dengan urutan pengerjaan antarkondisi yang dikendalikan secara terbatas dan dataset

Jenis validitas	*Threats*	Mitigasi
	paired comparison waktu seleksi	kandidat disetarakan, namun belum dieksekusi dalam lingkup penelitian ini karena keterbatasan akses operator
Internal	Evaluator merupakan informan wawancara pada BAB I	Potensi bias kesepakatan dan *social desirability* akibat familiaritas terhadap peneliti dan kepentingan terhadap proyek. Mitigasi: kondisi kasus disusun *as-is* berdasarkan data input, kriteria penilaian berbasis kesesuaian operasional (bukan preferensi terhadap AI), pencatatan alasan wajib pada skor 0/1, dan pasangan evaluator ditentukan berdasarkan relevansi domain. Keterbatasan ini diakui sebagai konsekuensi akses terbatas pada studi kasus tunggal, bukan dieliminasi sepenuhnya.
Construct	Definisi "kesesuaian" bersifat subjektif	Skala 0/1/2 dengan aspek eksplisit seperti sesuai kondisi kasus, alasan relevan, dan keakuratan klaim
External	Studi kasus tunggal (BSI UII)	Hasil merupakan temuan kontekstual, bukan generalisasi lintas organisasi
Reliability	Variabilitas keluaran LLM antareksekusi	*Temperature* rendah (0.2), top-p 0.8, validasi struktur keluaran, dan pengukuran konsistensi pada 26 kasus x 3 eksekusi
Statistical conclusion	Sampel terbatas dan berbasis studi kasus tunggal	*Percent agreement* dilaporkan per pasangan domain bersama distribusi skor, keterbatasan diakui secara eksplisit

Dengan memahami keterbatasan tersebut, hasil penelitian tetap dapat ditempatkan secara proporsional sebagai pengembangan artefak sistem pada konteks studi kasus tertentu, sekaligus menjadi dasar bagi penelitian dan pengembangan berikutnya.

3.12 Reproduksibilitas Penelitian

Untuk mendukung reproduksibilitas penelitian, beberapa elemen metodologis dan teknis didokumentasikan secara eksplisit. Pertama, konfigurasi pemanggilan model, termasuk *proxy* LLM, alias model, parameter *temperature*, *top-p*, *timeout, retry*, *throttle*, output format, validasi output, dan *fallback*, dirangkum pada Tabel 3.14. Kedua, penerapan format TOON, prosedur penyusunan konteks, dan metode pengukuran token dijelaskan pada Subbab 3.7.1 dan Subbab 3.8.7 agar proses pembentukan konteks dapat direplikasi pada data sejenis. Ketiga, kasus evaluasi kualitas keluaran AI disusun dalam 26 kasus terstruktur sebagaimana dijelaskan pada Tabel 3.15, dengan aspek penilaian, pasangan evaluator, dan skala skor yang dijelaskan pada Subbab 3.8.5.

Reproduksibilitas juga didukung melalui penggunaan *endpoint testing internal* untuk AI *Health Evaluation*. *Endpoint* tersebut digunakan sebagai *testing rig* agar *pipeline health* yang biasanya dijalankan oleh job terjadwal dapat dieksekusi ulang secara terkendali selama pengambilan output penelitian. Namun, reproduksibilitas penuh tetap memiliki keterbatasan karena sistem bergantung pada layanan model eksternal melalui *proxy* 9Router. Oleh karena itu, replikasi dengan konfigurasi yang sama masih dapat menghasilkan variasi keluaran sesuai karakteristik probabilistik LLM dan keterbatasan verifikasi versi model provider di balik *proxy*. Dengan demikian, reproduksibilitas penelitian ini terutama dijaga pada tingkat konfigurasi sistem, struktur kasus uji, format konteks, dan prosedur evaluasi, bukan pada jaminan *determinisme* penuh keluaran model.

BAB IV
HASIL PENELITIAN, PENGUJIAN, DAN PEMBAHASAN

4.1 Gambaran Umum Hasil Penelitian

Bab ini menyajikan hasil penelitian pengembangan Sistem Pendukung Keputusan Manajemen Aset TI Berbasis LLM pada studi kasus BSI UII, dengan fokus pada tiga kontribusi utama, yaitu AI *Assign Recommendation*, AI *Asset Insight*, dan AI *Health Evaluation*, serta integrasi monitoring dan *alert* sebagai penyedia konteks operasional. Pembahasan menekankan makna hasil penelitian, bukan dokumentasi teknis implementasi, sehingga tiap subbab disusun untuk menjawab tiga pertanyaan utama yaitu apa yang dilakukan sistem, apa hasil yang diperoleh, dan mengapa hasil tersebut penting bagi operasional.

Seluruh hasil yang dilaporkan pada bab ini diperoleh pada ruang uji terbatas, yaitu lingkungan pengembangan dengan volume data, jumlah pengguna, dan beban trafik yang terbatas. Karena itu, temuan dibaca sebagai indikasi awal pada konteks studi kasus BSI UII, bukan sebagai bukti kinerja pada skala produksi. Kualifikasi *scope* ini berlaku untuk seluruh subbab dan tidak diulang pada setiap pernyataan agar pembahasan hasil tetap ringkas dan tegas.

Secara arsitektural, sistem berjalan pada *service core* dan LangChain *service* yang saling terintegrasi. Pendekatan ini memungkinkan analisis aset dilakukan secara adaptif, namun tetap dikendalikan oleh aturan domain agar keluaran tetap konsisten dan dapat ditelusuri. Pada ruang lingkup implementasi ini, SAP tetap berperan sebagai sistem akuisisi dan pencatatan awal aset pada level universitas, sehingga pembahasan pada Bab ini difokuskan pada nilai tambah sistem di fase pascaakuisisi BSI UII. Dengan posisi tersebut, sistem yang dikembangkan bersifat pelengkap operasional SAP, bukan pengganti SAP.

Bukti visual pada Gambar 4.1 sampai Gambar 4.9 digunakan sebagai penguat analisis dan ditempatkan langsung pada subbab hasil terkait agar keterkaitan narasi dan temuan tetap jelas.

Tabel 4.1 *Pair matching* desain BAB III terhadap bukti BAB IV

Desain pada BAB III	Bukti pada BAB IV	Status
Frontend	Antarmuka modul aset dan AI pada Gambar 4.1, Gambar 4.3, Gambar 4.5, dan Gambar 4.7	Berjalan
Backend	API aset, *issue*, *maintenance*, dan *health* tervalidasi pada pengujian fungsional/integrasi	Berjalan
Database	Entitas *asset, issue, maintenance*, dan *health snapshot* tersimpan konsisten	Berjalan
LangChain AI *service*	Fitur AI *Assign*, AI *Insight*, dan AI *Health* menghasilkan keluaran terstruktur	Berjalan
Monitoring-*alert*	*Alert* valid memicu pembaruan evaluasi kesehatan (*health refresh*) dan sinkronisasi	Berjalan terbatas
Guardrail/fallback	Validasi respons *error* dasar menunjukkan sistem merespons kondisi tidak valid secara terkendali	Tervalidasi terbatas

4.2 Hasil Fitur Utama Penelitian

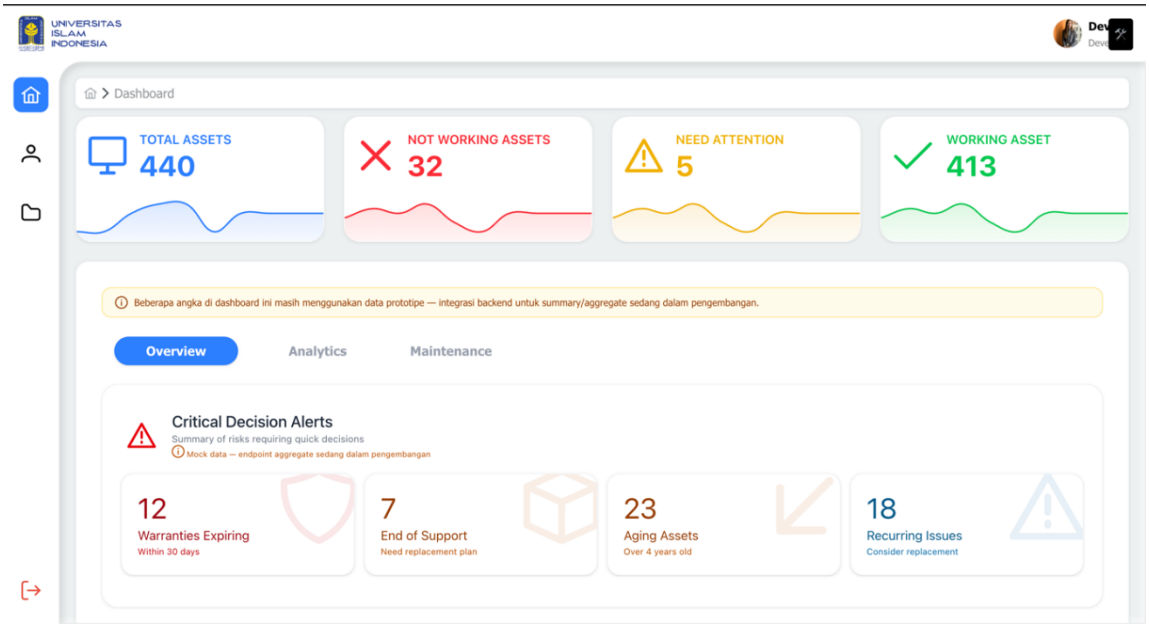
4.2.1 Ringkasan Domain yang Relevan

Domain yang dibahas pada subbab ini adalah domain yang berkontribusi langsung terhadap tujuan penelitian. Domain administratif umum, termasuk domain pengadaan/akuisisi pada SAP, tidak dijabarkan rinci karena berperan sebagai prasyarat sistem, bukan inti temuan penelitian. Domain yang berkontribusi langsung terhadap tujuan penelitian dirangkum pada Tabel 4.2.

Tabel 4.2 Ringkasan domain pendukung kontribusi penelitian

Domain	*Entity*	Peran pada penelitian
Operasional aset	`asset',`asset_pengguna`, `asset_issue`, `asset_ Maintenance'	Menyediakan data kondisi dan histori aset
AI *Insight*	`asset_ai_insight_cache`	Menyimpan hasil *insight* untuk kesinambungan akses
AI *Health*	`asset_health_snapshot`,`asset.id_status_health`	Menyimpan hasil evaluasi dan sinkronisasi status kesehatan
Monitoring dan *alert*	`asset_metric_hourly`, `asset_alerts`	Menyediakan konteks kondisi aktual untuk analisis

Temuan ini mengindikasikan bahwa pemisahan domain operasional, domain analitis, dan domain keputusan telah berjalan dengan baik. Pemisahan tersebut penting karena mendukung ketertelusuran alasan perubahan status aset.



Gambar 4.1 *Dashboard system.*

4.2.2 Hasil AI *Assign Recommendation*

a. Deskripsi Fitur

AI *Assign Recommendation* merupakan modul dukungan keputusan awal untuk alokasi aset, bukan modul eksekusi otomatis penugasan aset. Tujuan utamanya adalah mereduksi waktu seleksi kandidat perangkat serta menstandarkan alasan teknis yang digunakan operator saat menentukan aset yang akan dialokasikan.

Fitur ini memproses tiga kelompok input utama, yaitu profil pengguna (posisi dan kebutuhan kerja), data perangkat aktif yang sudah digunakan pengguna, serta kandidat aset yang berstatus tersedia. Ketiga input tersebut disusun menjadi konteks terstruktur sebelum dikirim ke komponen AI untuk menghasilkan urutan prioritas rekomendasi.

Output yang dihasilkan berupa daftar prioritas aset beserta alasan rekomendasi per kandidat. Ruang lingkup keputusan dibatasi pada faktor teknis dan ketersediaan data internal, keputusan final penugasan tetap dilakukan operator agar kebijakan non-teknis organisasi tetap terjaga.

b. **Ringkasan Alur**

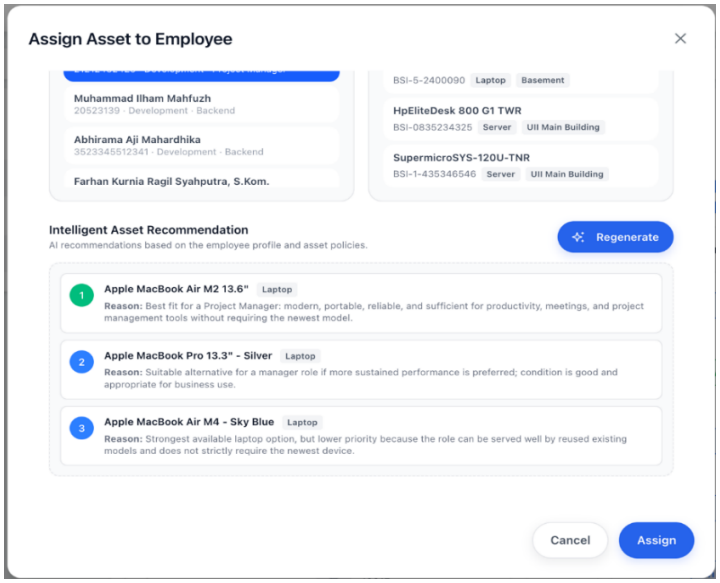
Karena alur teknis implementasinya telah diuraikan pada Subbab 3.5.5, bagian ini menyoroti hasil keluaran dan makna operasional fitur AI *Assign Recommendation.*

c. Cuplikan Hasil *Response* API


```
{
  "recommendations": [
    {
      "uuid_asset": "71ad5cac-c63c-47de-87e8-2fb68d3b621a",
      "asset_type": "desktop",
      "asset_model": "EliteDesk 800 G1 TWR",
      "priority": 1,
      "reason": "Kesesuaian spesifikasi komputasi dan ketersediaan status
borrowable"
    },
    {
      "uuid_asset": "5d58b6a4-56b2-4f5e-9b60-7bb14f50def0",
      "asset_type": "peripheral",
      "asset_model": "MX Master 4 Wireless",
      "priority": 2,
      "reason": "Alternatif kompatibel terhadap kebutuhan kerja harian
pengguna"
    }
  ]
}
```


Gambar 4.2 *Response* AI *Assign Recommendation*

Cuplikan pada Gambar 4.2 memperlihatkan bahwa keluaran sistem tidak hanya menampilkan kandidat aset, tetapi juga alasan prioritas yang dapat dipertanggungjawabkan. UUID pada seluruh cuplikan merupakan identifier ilustratif yang telah dianonimkan untuk kebutuhan publikasi dokumen.



-

Gambar 4.3 Tampilan hasil AI *Assign Recommendation*

d. Analisis

Berdasarkan observasi kualitatif, fitur ini berpotensi memperpendek waktu seleksi awal aset dibanding proses manual. Pada proses manual, operator umumnya memeriksa data profil pengguna, status aset, dan histori penggunaan secara terpisah. Pada implementasi AI *Assign*, prioritas dan alasan disajikan dalam satu keluaran sehingga titik awal pengambilan keputusan menjadi lebih terarah.

Untuk meminimalkan kesan subjektif, pengukuran waktu seleksi awal dirancang mengikuti protokol pada Tabel 4.3 Karena keterbatasan kondisi serta akses operator pada periode penelitian, pengukuran terstruktur ini belum dieksekusi dalam lingkup penelitian ini, status pelaksanaannya dirangkum pada Tabel 4.4.

Tabel 4.3 Protokol pengukuran waktu seleksi awal

Aspek	Definisi operasional
Objek ukur	Waktu seleksi awal kandidat aset oleh operator
Titik mulai	Operator mulai menelusuri kandidat aset untuk satu permintaan
Titik selesai	Operator menetapkan *shortlist* kandidat untuk keputusan assign
Mode pembanding	Mode manual (tanpa AI) vs *mode* AI *Assign* (dengan rekomendasi)

Aspek	Definisi operasional
Sampel	20 pengamatan (10 manual, 10 AI *Assign*) pada operator yang sama
Kontrol	Dataset kandidat setara (jumlah kandidat 3-8 aset), tanpa memasukkan waktu *approval* manajerial

Tabel 4.4 Status pelaksanaan pengukuran waktu seleksi awal

Aspek	Status
Protokol pengukuran	Dirancang (Tabel 4.3)
Pelaksanaan pengukuran terstruktur	Belum dieksekusi dalam lingkup penelitian ini
Analisis kuantitatif (median, IQR, *strip plot*)	Belum dihasilkan
Alasan penundaan	Keterbatasan akses operator pada periode penelitian
Tindak lanjut	Verifikasi kuantitatif efisiensi waktu ditunda untuk penelitian lanjutan

Karena pengukuran terstruktur belum dieksekusi, efisiensi waktu seleksi tidak dilaporkan dalam bentuk angka kuantitatif pada penelitian ini. Berdasarkan wawancara informal dengan 3 personel BSI UII, proses seleksi kandidat aset secara manual umumnya memerlukan waktu lebih dari lima menit per permintaan karena operator perlu merekonsiliasi data dari beberapa sumber terpisah, sedangkan AI *Assign* menyajikan prioritas dan alasan dalam satu keluaran sehingga titik awal pengambilan keputusan menjadi lebih terarah. Potensi efisiensi waktu ini bersifat indikasi kualitatif yang belum dikonfirmasi secara kuantitatif. Fokus utama evaluasi AI *Assign* tetap berada pada keterbacaan struktur rekomendasi, relevansi alasan, dan kesesuaian hasil menurut evaluator domain. Verifikasi kuantitatif waktu seleksi pada ukuran sampel dan variasi operator yang lebih besar ditunda untuk penelitian lanjutan.

Kinerja AI *Assign* bekerja paling optimal ketika profil pengguna terisi lengkap (posisi, kebutuhan kerja, dan perangkat aktif), data ketersediaan aset mutakhir, serta kandidat aset cukup beragam. Kinerja menjadi kurang optimal ketika data profil pengguna minim, stok aset sangat terbatas, atau status aset belum diperbarui tepat waktu, karena model kehilangan konteks pembeda antarkandidat.

Risiko utama fitur ini adalah bias rekomendasi akibat ketergantungan pada kualitas data pengguna dan histori keputusan sebelumnya. Selain itu, rekomendasi belum mempertimbangkan faktor non-teknis seperti preferensi personal, kebijakan departemen non-standar, atau urgensi lintas tim yang tidak terekam di data. Untuk kandidat dengan skor yang

berdekatan, urutan prioritas AI dapat bervariasi antareksekusi, sehingga verifikasi operator tetap diperlukan agar keputusan akhir konsisten. Fitur AI *Assign* menunjukkan potensi mendukung keputusan alokasi awal secara lebih terstruktur, dengan catatan evaluasi lanjutan pada skala pengguna dan data yang lebih besar tetap diperlukan.

4.2.3 Hasil AI *Asset Insight*

a. Deskripsi Fitur

AI *Asset Insight* berperan sebagai modul interpretasi data yang mengubah data operasional heterogen menjadi narasi kondisi aset yang mudah ditindaklanjuti. Tujuan fitur ini adalah mengurangi beban operator dalam membaca data mentah yang tersebar pada beberapa sumber.

Input fitur mencakup histori *issue*, histori *maintenance,* data metrik dan *alert* (untuk aset monitoring), serta konteks identitas dan klasifikasi aset. Sistem memilih mode analisis sesuai karakteristik aset agar *insight* yang dihasilkan relevan dengan kondisi lapangan.

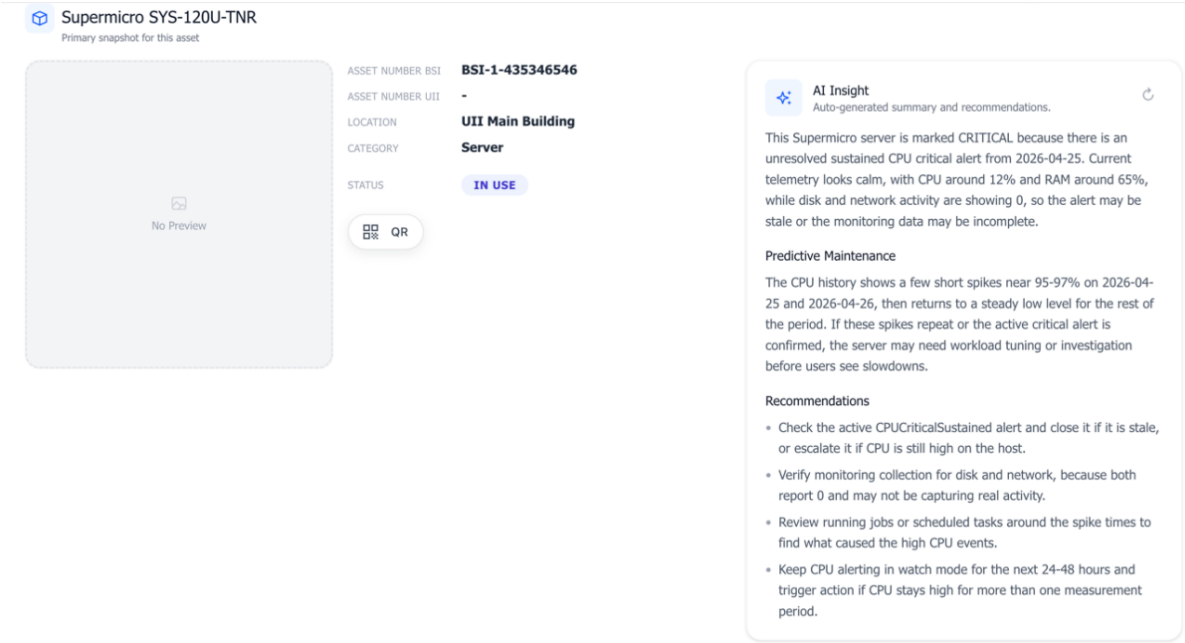
Output *insight* meliputi ringkasan kondisi saat ini, indikasi *preventive* atau *predictive maintenance*, dan rekomendasi tindakan prioritas. Ruang lingkup fitur dibatasi sebagai penyedia interpretasi operasional, fitur tidak mengeksekusi tindakan perbaikan otomatis tanpa validasi manusia.

b. Hasil *Response* API


```
{
  "current_condition": "Terdapat gejala degradasi performa periodik pada aset
monitoring dengan alert warning berulang.",
  "predictive_maintenance": "Disarankan preventive maintenance kapasitas dan
penyesuaian threshold alert dalam horizon mingguan.",
  "recommendations": [
    "Prioritaskan investigasi beban pada periode puncak trafik.",
    "Lakukan tuning ambang CPU/RAM bertingkat sebelum critical."
  ]
}
```


Gambar 4.4 Respons AI *Asset Insight*

Output pada Gambar 4.4 ini mengindikasikan bahwa data mentah berhasil ditransformasikan menjadi informasi yang lebih mudah digunakan untuk tindakan operasional.



Gambar 4.5 Tampilan hasil AI *Asset Insight*

c. Analisis

Hasil pengujian memperlihatkan bahwa pembedaan mode monitoring dan non-monitoring berjalan konsisten. Temuan ini menjadi relevan karena kebutuhan informasi kedua kelompok aset tersebut memang berbeda. Dibandingkan pembacaan data mentah secara manual, *insight* naratif menunjukkan kecenderungan membantu identifikasi prioritas tindakan dan membuat ringkasan kondisi lebih mudah dibaca operator.

Fitur *insight* bekerja optimal ketika histori *issue*, histori *maintenance*, dan data metrik/*alert* tersedia dengan timestamp yang konsisten. Pada kondisi data parsial atau tidak sinkron, kualitas narasi dapat menurun karena sistem hanya menyimpulkan dari konteks yang tersedia. Risiko utamanya adalah misinterpretasi narasi oleh pengguna, terutama jika ringkasan dibaca tanpa melihat indikator sumber seperti jenis *alert* aktif atau pola gangguan berulang.

Dari perspektif permasalahan Bab I, fitur ini menjawab isu data operasional yang tidak terstruktur dengan mengubah data mentah menjadi narasi yang lebih operasional. Walaupun demikian, output LLM tetap bersifat probabilistik sehingga frasa dan penekanan *insight* dapat sedikit berbeda pada eksekusi berulang (Huang et al., 2025). Fitur ini menunjukkan

kecenderungan membantu operator memperjelas prioritas tindak lanjut harian karena analisis tidak lagi selalu dimulai dari data mentah yang tersebar.

4.2.4 Hasil AI *Health Evaluation*

Berbeda dari dua fitur sebelumnya yang menekankan rekomendasi dan interpretasi, AI *Health Evaluation* difokuskan pada konsistensi penetapan status kesehatan aset.

a. **Deskripsi Fitur**

AI *Health Evaluation* merupakan modul evaluasi status kesehatan aset yang menggabungkan pendekatan AI dan *rule based*. Tujuan utamanya adalah menghasilkan status kesehatan yang konsisten, *auditabel*, dan dapat disinkronkan ke sistem inti sebagai dasar tindak lanjut operasional.

Input evaluasi terdiri atas data metrik, *alert*, histori *issue*, histori *maintenance,* serta konteks siklus hidup aset. Pada aset *high-risk*, status final dibentuk dengan skema gabungan yaitu sinyal operasional dievaluasi dengan aturan domain dan konteks *lifecycle* diperkaya AI. Pada aset *low-risk*, status final dari AI tetap divalidasi terhadap referensi status kesehatan yang sah pada master data.

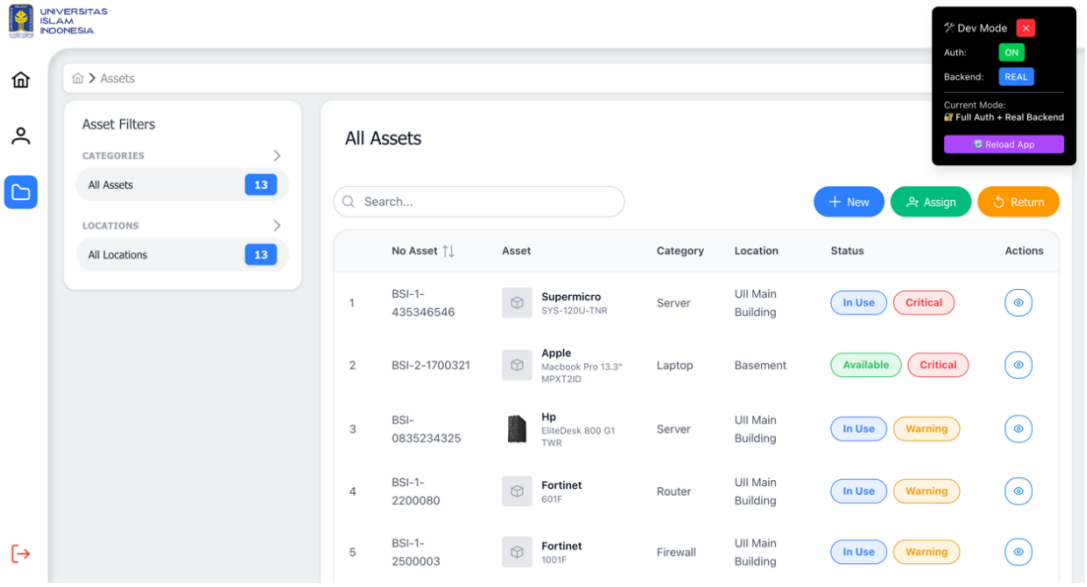
Output fitur mencakup ‘operational_health’, ‘lifecycle_health’, ‘final_health_status’, ‘final_health_score’, tingkat kepercayaan, dan faktor dominan yang memengaruhi hasil. Ruang lingkup fitur dibatasi sebagai mesin evaluasi dan sinkronisasi status, keputusan aksi lanjutan di lapangan tetap menjadi kewenangan operator atau unit terkait.

b. Hasil *Response* API


```
{
  "operational_health": "CRITICAL",
  "lifecycle_health": "WARNING",
  "final_health_status": "CRITICAL",
  "final_health_score": 35,
  "confidence": 81.2,
  "top_factors": [
    "Alert kritis masih aktif",
    "CPU tinggi berkelanjutan",
    "Riwayat issue performa belum stabil"
  ]
}
```


Gambar 4.6 Respons AI *Health Evaluation*

Temuan pada Gambar 4.6 tersebut menunjukkan bahwa keputusan kesehatan final diturunkan dari kombinasi indikator operasional, histori aset, dan aturan risiko.



Gambar 4.7 Tampilan status kesehatan aset

c. Analisis

Secara analitis, temuan utama fitur ini terletak pada keseimbangan antara adaptivitas AI dan kontrol aturan domain. Pendekatan tersebut mengurangi risiko keputusan yang terlalu bergantung pada model, sekaligus tetap memanfaatkan kemampuan AI dalam membaca konteks histori aset. Dampaknya, hasil evaluasi menjadi lebih adaptif namun tetap *auditabel*.

Kondisi kerja optimal tercapai ketika data metrik, *alert*, histori *maintenanc*e, dan histori *issue* tersedia serta referensi status kesehatan pada master data valid. Kinerja menurun saat data monitoring kosong, *alert* tidak memiliki identitas aset, atau pembaruan histori terlambat, karena evaluasi hanya dapat bergantung pada sebagian sinyal.

Keterbatasan utama fitur ini adalah potensi konflik antara keluaran AI dan aturan *guardrail*. Pada beberapa kasus, *guardrail* dapat menahan hasil AI yang terlalu optimistis. Di sisi lain, *guardrail* juga dapat membuat status akhir lebih konservatif daripada interpretasi model. Karena itu, evaluasi *health* perlu diperlakukan sebagai dukungan keputusan yang ditinjau periodik, bukan keputusan final otomatis (Durachman et al., 2026). Dalam konteks Bab I, fitur ini menjawab masalah ketidakjelasan *lifecycle* aset dengan menyediakan status kesehatan terstruktur dan faktor penyebab utama. Mekanismwe ini menunjukkan

kecenderungan membantu deteksi dini risiko operasional dan memudahkan sinkronisasi status antaroperator.

4.2.5 Hasil Integrasi Monitoring dan *Alert*

Sebagai pengikat antarfitur, integrasi monitoring dan alert dirancang untuk menyediakan konteks operasional bagi rekomendasi, insight, dan evaluasi health. Namun, penting untuk ditegaskan bahwa pada lingkungan studi kasus BSI UII, integrasi real-time dengan sistem monitoring Prometheus belum dapat diimplementasikan penuh karena memerlukan pemetaan identitas aset dan koordinasi lintas unit yang membutuhkan pembahasan mendalam serta jangka waktu yang tidak singkat. Oleh karena itu, evaluasi pada penelitian ini menggunakan data monitoring dari ekspor CSV yang diberikan oleh NDC yang merepresentasikan data operasional aktual pada periode tertentu, beserta *synthetic alerts* yang disinkronkan dengan metrik tersebut.

a. **Deskripsi Fitur**

Integrasi monitoring dan *alert* dirancang sebagai mekanisme penyedia konteks operasional untuk mendukung *insight* dan evaluasi *health*. Tujuan integrasi ini adalah mempersingkat jeda pembaruan penilaian ketika terjadi perubahan kondisi lapangan yang bermakna secara operasional.

Input utama integrasi berupa metrik periodik dan *event alert* dari sistem pemantauan. Setiap *event* melewati validasi identitas aset sebelum digunakan sebagai pemicu pembaruan *health*, sehingga risiko pembaruan status pada aset yang salah dapat ditekan.

Ruang lingkup integrasi difokuskan pada ingestion data kontekstual dan *trigger* evaluasi ulang, bukan pengambilan keputusan bisnis secara langsung. Dengan batasan ini, integrasi berfungsi sebagai penguat akurasi konteks bagi fitur AI, bukan pengganti proses verifikasi operasional.

b. *Response Event*

```
{
  "operational_health": "CRITICAL",
  "lifecycle_health": "WARNING",
  "final_health_status": "CRITICAL",
  "final_health_score": 35,
  "confidence": 81.2,
  "top_factors": [
    "Alert kritis masih aktif",
    "CPU tinggi berkelanjutan",
    "Riwayat issue performa belum stabil"
  ]
}
```


Gambar 4.8 *event alert* yang telah dianonimkan.

Pada Gambar 4.8 memperlihatkan bahwa perubahan kondisi operasional dapat direspons melalui evaluasi *health* dengan jeda yang lebih pendek.

c. Analisis

Integrasi monitoring dan *alert* menunjukkan potensi memperkuat kemampuan sistem dalam merespons perubahan kondisi aset dibanding pendekatan manual berkala. Implikasi operasionalnya adalah deteksi risiko layanan yang lebih dini dan tindak lanjut yang lebih terarah. Intergrasi ini juga menunjukkan kecenderungan menurunkan *blind spot* karena perubahan kondisi tidak menunggu evaluasi berkala manual.

Data monitoring pada evaluasi BAB IV menggunakan pendekatan *scenario-based synthetic dataset* yang dasarnya adalah data *real* NDC. *Alert* yang digunakan merupakan *synthetic alert* yang disinkronkan dengan injeksi metrik karena akses *Alertmanager live* tidak tersedia. Data *disk* dari NDC tidak tersedia dan diisi 0 sebagai *placeholder* untuk memenuhi kontrak JSON, sehingga tidak digunakan sebagai dasar evaluasi maupun klaim kondisi *disk*.

4.3 Hasil Pengujian Fungsional

Setelah hasil implementasi tiap fitur diuraikan, pembahasan dilanjutkan pada pengujian fungsional untuk memastikan setiap komponen beroperasi sesuai tujuan penelitian.

Tabel 4.5 Ringkasan hasil pengujian fungsional

No	Fitur	Hasil	Status	Makna hasil
1	Modul aset dan penggunaan	Proses data aset berjalan konsisten	Lulus	Fondasi data operasional memadai untuk fitur AI

No	Fitur	Hasil	Status	Makna hasil
2	AI *Assign Recommendation*	Rekomendasi berisi prioritas dan alasan	Lulus dengan catatan	Mendukung keputusan awal, namun masih bergantung pada layanan AI
3	AI *Asset Insight*	Narasi *insight* terbentuk sesuai *mode* aset	Lulus	Membantu interpretasi kondisi aset secara cepat
4	AI *Health Evaluation*	*Snapshot* dan sinkronisasi status berjalan	Lulus	Menjaga konsistensi status kesehatan antarmodul
5	Integrasi *alert* dan metrik	*Event* operasional memicu evaluasi *health*	Lulus dengan catatan	Responsif, tetapi data tanpa identitas aset tidak dieksekusi

Tabel 4.6 Detail pengujian modul inti

Kode uji	Skenario	Detail	Kondisi batas	Hasil aktual	Status
BRW-01	*Assign aset valid*	Input menggunakan uuid dari aset yang *borrowable*, uuid pengguna valid serta tanggal peminjaman terisi	Aset status ASSET_AVAILABLE/ASSET_NEW	*Record borrow* terbentuk, status aset berubah ke ASSET_IN_USE	Lulus
BRW-02	*Assign saat aset* tidak *borrowable*	*uuid_asset* berstatus *non-borrowable*	Status aset selain ASSET_AVAILABLE/ASSET_NEW	*Request* ditolak dengan pesan validasi status pinjam	Lulus
BRW-03	*Return* aset aktif	*uuid borrow* aktif	*Borrow* masih aktif (`flag_aktif` =1, tgl_akhir null)	`*flag*_aktif` menjadi 0, `tgl_akhir` terisi, `status` kembali `ASSET_AVAILABLE` bila tidak ada *borrow* aktif lain	Lulus
BRW-04	Return ulang pada *borrow* yang sudah ditutup	*uuid borrow* nonaktif	*Borrow* sudah memiliki tgl_akhir	*Request* ditolak dengan pesan "*already returned*"	Lulus

Kode uji	Skenario	Detail	Kondisi batas	Hasil aktual	Status
ISS-01	Tambah *issue* valid	*uuid_asset* valid dan data *issue*	*Asset* tersedia	*Issue* tersimpan dan menjadi bagian histori aset	Lulus
ISS-02	*Update issue* dengan UUID tidak ada	*uuid issue* tidak ditemukan	Referensi *issue invalid*	*Request* ditolak dengan pesan "*issue not found*"	Lulus
MNT-01	Tambah *maintenance* tanpa uuid_status	*Payload maintenance* tanpa status	*Field* wajib kosong	*Request* ditolak dengan pesan "*uuid_status is required*"	Lulus
MNT-02	Hapus *maintenance* (*soft delete*)	*uuid maintenance valid*	*Record* tersedia	*flag_aktif maintenance* menjadi 0 dan histori tetap terlacak	Lulus

Tabel 4.6, menampilkan representatif dari pengujian modul inti yang relevan langsung dengan ruang lingkup penelitian. Dokumentasi teknis yang lebih rinci disimpan pada berkas hasil pengujian dan lampiran pendukung.

Pada dasarnya, hasil pengujian fungsional memperlihatkan bahwa alur inti penelitian telah berjalan menyeluruh pada proses internal pascaakuisisi BSI UII. Hal ini mengindikasikan bahwa sistem tidak berhenti pada tahap prototipe, tetapi telah memenuhi fungsi operasional dasar pada ruang lingkup studi kasus. Hasil evaluasi dukungan proses bisnis operasional disajikan pada Tabel 4.7.

4.3.1 Evaluasi Dukungan Proses Bisnis

Selain lulus secara fungsional, sistem juga dievaluasi terhadap dukungannya pada proses kerja operasional. Evaluasi ini penting agar hasil penelitian tidak terbatas pada konfirmasi keberfungsian teknis, melainkan juga mengkaji kontribusi fitur terhadap pengambilan keputusan harian. Hasil evaluasi dukungan proses bisnis operasional disajikan pada Tabel 4.7.

Tabel 4.7 Evaluasi dukungan proses bisnis operasional

Proses kerja operasional	Bukti hasil pada BAB IV	Interpretasi dukungan
Pencarian kondisi aset	*Dashboard* status dan ringkasan *health* (Gambar 4.1)	Informasi kondisi awal lebih sistematis ditelusuri

Proses kerja operasional	Bukti hasil pada BAB IV	Interpretasi dukungan
Seleksi kandidat aset	AI *Assign* dengan prioritas dan alasan (Gambar 4.3)	Penyaringan awal kandidat lebih terstruktur
Interpretasi *issue/maintenance*/monitoring	AI *Insight mode monitoring/non-monitoring* (Gambar 4.5)	Data mentah lebih mudah dipahami menjadi tindakan
Evaluasi *health* aset	AI *Health, guardrail,* sinkronisasi status (Gambar 4.7)	Penilaian risiko aset lebih konsisten
Respons *alert*	Integrasi *webhook dan refresh health*	Respons kejadian kritis lebih terarah pada ruang uji terbatas

Tabel tersebut menunjukkan bahwa sistem memberikan dukungan proses kerja pada lima titik keputusan utama, namun tetap ditempatkan sebagai dukungan keputusan berbasis *DSS human-in-the-loop* (Durachman et al., 2026).

4.4 Hasil Pengujian Integrasi

Selain verifikasi fungsional pada level modul, penelitian ini juga menilai kesinambungan pertukaran data antarlayanan melalui pengujian integrasi serta validitas struktur keluarannya.

Tabel 4.8 Ringkasan hasil pengujian integrasi

Integrasi	Hasil utama	Status	Implikasi
Aplikasi inti ke AI *Assign*	Struktur rekomendasi tervalidasi	Berhasil	Keputusan alokasi dapat diterapkan pada antarmuka operasional
Aplikasi inti ke AI *Insight*	*Insight* tervalidasi, *fallback* tersedia saat *endpoint* transisi	Berhasil dengan catatan	Layanan tetap berlanjut saat terjadi perubahan *endpoint*
Aplikasi inti ke AI *Health Evaluation*	Evaluasi status kesehatan tervalidasi	Berhasil	Status kesehatan dapat disinkronkan konsisten
Alert webhook ke AI *Health Evaluation*	*Event* valid memicu evaluasi ulang	Berhasil dengan catatan	Menunjukkan potensi responsivitas yang lebih baik terhadap kondisi kritis
Job metrik ke AI *Health Evaluation*	*Snapshot* metrik masuk ke alur evaluasi	Berhasil	Menambah ketepatan konteks evaluasi operasional

Secara bertahap, temuan ini menegaskan bahwa integrasi antarlayanan tidak hanya berjalan, tetapi juga memiliki ketahanan dasar terhadap gangguan kompatibilitas terbatas.

4.5 Evaluasi Kuantitatif Performa *Endpoint* dan Efisiensi Data

Untuk melengkapi hasil pengujian fungsional dan integrasi, evaluasi kuantitatif pada subbab ini difokuskan pada performa layanan AI dan keandalan mekanisme pemulihan ketika terjadi gangguan.

Tabel 4.9 Indikator konfigurasi keandalan yang diterapkan

Indikator	Nilai	Interpretasi
Retry maksimum	2	Membatasi kegagalan berulang agar tidak membebani layanan
Jeda minimum antarpermintaan	3000 ms	Menjaga jarak antarpermintaan untuk mereduksi lonjakan trafik
Jeda awal r*etry*	1500 ms	Menentukan jeda awal pemulihan saat gagal
Jeda maksimum *retry*	8000 ms	Membatasi jeda maksimum agar layanan tetap responsif
Cooldown trigger EVENT health	10 menit	Mencegah evaluasi berulang berlebihan pada *event* berdekatan
Cooldown trigger MANUAL health	10 menit	Menjaga stabilitas saat ada intervensi operator
Interval refresh low-risk	30 hari	Menyesuaikan frekuensi evaluasi dengan profil risiko rendah
Unit test domain/ai/usecase	6/6 lulus	Menunjukkan konsistensi logika inti pada ruang uji terbatas

Limited-concurrency load test dilakukan pada lingkungan pengembangan menggunakan skenario 1, 5, dan 10 *concurrent requests* untuk *endpoint* AI *Assign*, AI *Insight*, dan AI *Health.* Titik pengukuran berada pada *request* dari *client* pengujian menuju *endpoint backend,* sehingga nilai latensi pada Tabel 4.10 dan Tabel 4.11 merepresentasikan waktu respons *endpoint* pada jalur uji, bukan latensi inferensi LLM *end-to-end* yang mencakup komunikasi ke LangChain *service*, proxy 9Router, dan provider LLM. Nilai tersebut dapat dipengaruhi oleh *cache, fallback,* kondisi jaringan lokal, dan jalur uji yang tidak selalu memanggil model eksternal pada setiap permintaan. Pengujian ini diperlakukan sebagai pengukuran terbatas pada ruang uji, bukan tolok ukur performa produksi.

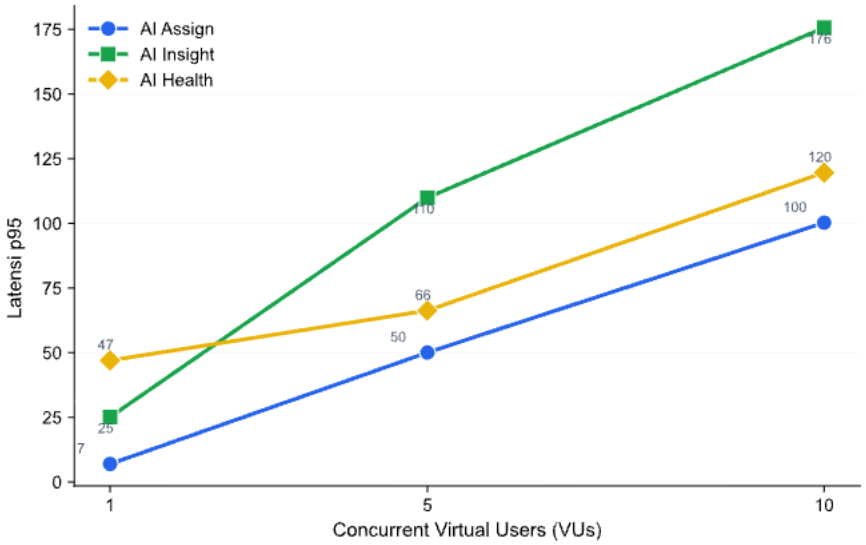
Tabel 4.10 Hasil observasi latensi *backend* pada *limited-concurrency load test*

Endpoint	**Min observasi (s)**	**Max observasi (s)**	**Rata-rata *latency* (s)**	**Rata-rata p50 (s)**	**Rata-rata p95 (s)**	***Error Rate* (%)**
AI *Assign*	6,956	100,267	31,401	31,314	52,426	0,0
AI *Insight*	25,112	175,667	88,912	88,851	103,566	0,0
AI *Health*	40,895	119,638	51,996	52,636	77,638	0,0

Tabel 4.10, menunjukkan bahwa seluruh *endpoint* AI tetap mengembalikan respons berhasil pada skenario 1, 5, dan 10 *concurrent requests*. *Endpoint* AI *Insight* memiliki latensi tertinggi pada jalur uji karena memproses keluaran naratif yang lebih panjang. AI *Assign* relatif lebih rendah pada *baseline* 1 VU, tetapi tetap meningkat pada 5 dan 10 VUs.

Tabel 4.11 Rincian p50/p95 *backend* pada *limited-concurrency load test*

Endpoint	**VUs**	**p50 (s)**	**p95 (s)**	***Error Rate* (%)**
AI *Assign*	1	6,956	6,956	0,0
AI *Assign*	5	28,156	50,056	0,0
AI *Assign*	10	58,830	100,267	0,0
AI *Insight*	1	25,112	25,112	0,0
AI *Insight*	5	88,517	109,919	0,0
AI *Insight*	10	152,923	175,667	0,0
AI *Health*	1	46,997	46,997	0,0
AI *Health*	5	41,836	66,281	0,0
AI *Health*	10	69,075	119,638	0,0



Gambar 4.9 Plot latensi p95 pada skenario 1, 5, dan 10 VUs untuk tiga *endpoint*

Berdasarkan Tabel 4.11 dan Gambar 4.9, peningkatan beban menyebabkan kenaikan p95 yang jelas pada ketiga *endpoint* AI. Pola ini menunjukkan bahwa jalur *backend*/uji tetap sensitif terhadap peningkatan *concurrency*. Hasil ini menunjukkan bahwa sistem dapat mempertahankan keberhasilan respons pada beban ringan terbatas, namun pengukuran tambahan terhadap latensi lintas layanan dan optimasi antrean, *caching*, atau pembatasan *concurrency* tetap diperlukan sebelum digunakan sebagai beban produksi. Penurunan p50 pada AI *Health* dari 1 VU (46,997 detik) ke 5 VU (41,836 detik) disebabkan oleh mekanisme *cooldown health evaluation* (10 menit untuk *trigger manual*) yang berlaku pada aset yang sama. Pada skenario 5 VU, seluruh permintaan konkuren diarahkan ke aset yang sama, hanya permintaan pertama yang menjalankan *pipeline* LLM penuh, sedangkan permintaan berikutnya dikembalikan dari *snapshot* yang sudah tersimpan sehingga latensinya jauh lebih rendah. Perilaku ini merupakan desain *anti-flapping* yang bekerja sebagaimana mestinya, bukan anomali kinerja.

Untuk menjaga keterlacakan kualitas format keluaran, validitas struktur respons AI representatif dicatat dari 26 kasus evaluasi final.

Tabel 4.12 Validitas struktur output AI representatif

Kelompok	Fitur	Jumlah kasus	Struktur valid	Struktur tidak valid
A	AI *Health monitoring*	4	4	0
A	AI *Insight monitoring*	4	4	0
B	AI *Health non-monitoring*	6	6	0
B	AI *Insight non-monitoring*	6	6	0
B	AI *Assign*	6	6	0
Total		26	26	0

Validitas struktur pada Tabel 4.12 dibaca sebagai keberhasilan output melewati validator format yang telah ditetapkan, bukan sebagai bukti bahwa substansi seluruh keluaran sudah benar secara domain. Validator memeriksa keberadaan *field* wajib, kesesuaian bentuk data, dan kemampuan *backend* memproses respons tanpa *error*. Hal itu menunjukan, capaian 26/26 valid menunjukkan bahwa *pipeline* final mampu menghasilkan keluaran yang dapat diproses sistem pada kasus evaluasi, sedangkan kualitas substansi tetap dinilai melalui evaluasi *rater* pada Subbab 4.7.

Selain *limited-concurrency load test*, dilakukan validasi respons *error* terkendali untuk memverifikasi bahwa sistem memberikan respons yang aman ketika menerima input atau kondisi akses yang tidak valid. Pengujian ini difokuskan pada kesalahan yang dapat dipicu tanpa mengubah konfigurasi produksi layanan, yaitu UUID tidak valid, *payload* kosong, *endpoint* tidak ditemukan, dan autentikasi kosong. Dengan demikian, Tabel 4.13 dibaca sebagai *negative testing* terbatas, bukan *fault injection* penuh terhadap kegagalan layanan AI eksternal. Selain *error rate* 0%, keberhasilan seluruh respons pada load test juga menyiratkan bahwa struktur output AI yang dikembalikan valid secara format, karena kegagalan validasi struktur pada sisi *backend* akan tercermin sebagai peningkatan *error rate*. Maka dari itu, indikator validitas struktur output pada *limited-concurrency load test* terpenuhi secara implisit melalui *error rate* 0%.

Tabel 4.13 Hasil validasi respons *error* terkendali

Skenario	Jumlah uji	Berhasil	Gagal	Proporsi	Analisis
FB-01 *Invalid asset* UUID detail	5	5	0	100%	Sistem mengembalikan HTTP 400 secara terkendali, bukan fatal *error.*

Skenario	Jumlah uji	Berhasil	Gagal	Proporsi	Analisis
FB-02 AI *Health payload* kosong	5	5	0	100%	Sistem menolak *payload* kosong dengan HTTP 400.
FB-03 *Endpoint* tidak ditemukan	5	5	0	100%	Sistem mengembalikan HTTP 404 untuk *route* tidak tersedia.
FB-04 *Auth* kosong pada AI *Assign*	5	5	0	100%	Sistem menolak *request* tanpa autentikasi dengan HTTP 400.

Analisis pada Tabel 4.13 dibaca sebagai verifikasi bahwa mekanisme validasi dan respons *error* berjalan terkendali pada skenario kesalahan dasar. Skenario kegagalan layanan AI eksternal seperti HTTP 500, HTTP 429, dan *timeout* yang dirancang pada BAB III telah dieksekusi secara terbatas melalui *fault injection* terhadap *endpoint* 9Router menggunakan *mock proxy*. Hasil eksekusi tersebut dilaporkan pada Subbab 4.5.1.

4.5.1 Hasil *Fault Injection* terhadap Mekanisme *Retry* dan *Fallback*

Pengujian dilakukan dengan menempatkan *mock proxy* lokal sebagai pengganti 9Router. *Mock proxy* menerima panggilan `/v1/chat/completions` dan dapat dikonfigurasi untuk mengembalikan status HTTP 500, 429, atau 504 secara dinamis. Pendekatan ini memungkinkan verifikasi mekanisme pemulihan secara menyeluruh, *retry* pada Go *backend* maupun pada LangChain AI *service*, keduanya dengan konfigurasi *retry* maksimum 2 kali, jeda awal 1.500 ms, dan jeda maksimum 8.000 ms (Tabel 4.9). Istilah *retry* maksimum 2 kali di sini berarti terdapat paling banyak dua percobaan ulang setelah percobaan pertama, sehingga total upaya per komponen adalah 3 (1 percobaan awal dan 2 *retry*).

Tiga skenario kegagalan diuji pada tiga fitur AI (*Assign, Insight, Health Evaluation*) dengan lima iterasi per kombinasi, menghasilkan 45 uji. Parameter jeda *retry* dipertahankan pada nilai produksi agar durasi pengujian mencerminkan perilaku nominal mekanisme pemulihan. Hasil agregat disajikan pada Tabel 4.14.

Tabel 4.14 Hasil *faulit injection* per skenario kegagalan dan fitur AI (agregat lima iterasi)

Skenario	**Fitur**	**Jumlah uji**	**Respons sistem**	**HTTP status**	***Fallback* terverifikasi**	**Durasi rata-rata (s)**
HTTP 500	AI *Assign*	5	Penolakan input	400	Reject terkontrol setelah *retry exhausted*	19,5
HTTP 500	AI *Insight*	5	Pesan *fallback graceful*	200	LangChain AI *Service graceful fallback*	12,1
HTTP 500	AI *Health*	5	Evaluasi berbasis aturan lokal	200	*Go local rule-based fallback*	4,6
HTTP 429	AI *Assign*	5	Penolakan input	400	*Reject terkontrol setelah retry exhausted*	19,6
HTTP 429	AI I*nsight*	5	Pesan *fallback graceful*	200	LangChain AI *Service graceful fallback*	12,1
HTTP 429	AI *Health*	5	Evaluasi berbasis aturan lokal	200	*Go local rule-based fallback*	4,5
HTTP 504	AI *Assign*	5	Penolakan input	400	*Reject* terkontrol *setelah retry exhausted*	20,4
HTTP 504	AI *Insight*	5	Pesan *fallback graceful*	200	LangChain AI *Service graceful fallback*	12,7
HTTP 504	AI *Health*	5	Evaluasi berbasis aturan lokal	200	*Go local rule-based fallback*	4,9

Nilai durasi rata-rata dihitung dari iterasi ke-2 hingga ke-5 iterasi pertama dikecualikan karena durasi lebih tinggi akibat *warm-up*, sebagai contoh 36.0 detik pada AI *Assign* skenario HTTP 500. Seluruh 45 uji berlangsung tanpa *failure* pada *Go backend* maupun LangChain AI *service*.

Tabel 4.14, ketiga skenario kegagalan menghasilkan pola *response* yang konsisten per fitur. AI *Assign* menolak *request* secara aman (HTTP 400), sedangkan AI *Insight* dan AI *Health Evaluation* mengembalikan *response success* (HTTP 200) melalui mekanisme *fallback* yang berbeda. Konsistensi pola lintas skenario 500, 429, dan 504 menunjukkan bahwa mekanisme pemulihan tidak sensitif terhadap jenis status kegagalan, melainkan di-*trigger* oleh *failure*

retryable (429 atau ≥500) sebagaimana yang telah dirancang pada *layer* LangChain *service* dan *layer backend* Golang.

Untuk memverifikasi jumlah upaya *retry* yang dilakukan pada setiap *layer*, jumlah panggilan mock *proxy* direkam per uji. Karena satu uji dapat memicu beberapa sub-*pipeline* (misalkan AI *Insight* dengan parameter 'force = true' memanggil *pipeline health* dan *pipeline insight* secara berurutan), jumlah panggilan *mock* dibaca sebagai indikator aktivasi *retry*, bukan jumlah permintaan pengguna.

Tabel 4.15 Verifikasi aktivasi *retry* per lapis dan strategi *fallback* per fitur (agregat 15 uji)

Fitur	**Panggilan *mock* per uji (stabil)**	***Retry* LangChain (*Layer* 2)**	***Retry* Go (*Layer* 1)**	**Strategi *fallback***	**Non-fatal**
AI *Assign*	9	3× terverifikasi	3× terverifikasi	*Reject* terkontrol HTTP 400	Ya
AI *Insight*	6	3×2 *sub-pipeline* terverifikasi	Tidak terpicu	LangChain *graceful fallback* HTTP 200	Ya
AI *Health*	3	3× terverifikasi	Tidak terpicu	*Go local rule-based* HTTP 200	Ya

Pada AI *Assign*, satu uji memicu 9 panggilan *mock* - dengan 3 kali *call* di LangChain dan 3 kali *call* di Go. Setelah seluruh *retry* di LangChain habis, LangChain *service* mengembalikan HTTP 500 ke Go *backend*, yang kemudian memicu Go untuk mengulangi permintaan yang sama. Pola ini memvalidasi mekanisme pemulihan dua arah sesuai rancangan pada BAB III.

Situasi berbeda terlihat pada AI *Insight* dan AI *Health Evaluation*. Pada AI *Insight*, LangChain *service* menangkap kegagalan LLM dan mengembalikan pesan *fallback* yang valid secara TOON, sehingga Go menerima respons HTTP 200 dan tidak perlu melakukan *retry*. Pada AI *Health Evaluation*, Go *backend* langsung beralih ke evaluasi berbasis aturan lokal setelah menerima laporan kegagalan dari LangChain, sehingga *retry* di Go juga tidak diperlukan. Kedua perilaku ini konsisten dengan prinsip desain yang mendasari arsitektur dua *layer*, *retry* di Go hanya diaktifkan ketika LangChain mengembalikan *failure code* (HTTP 5xx), dan tidak diaktifkan Ketika *fallback* di LangChain telah menghasilkan *response* valid. Secara keseluruhan, hasil *fault injection* menunjukkan bahwa mekanisme *retry* dan *fallback* bekerja sesuai rancangan pada seluruh fitur yang diuji.

Terdapat tiga catatan atas batas hasil ini. *Pertama*, *fault injection* dilakukan pada lingkungan pengembangan menggunakan *mock proxy*, perilaku pada gangguan produksi nyata dapat berbeda, terutama terkait pola *timeout* asinkron dan *backpressure* jaringan. *Kedua*, meskipun parameter *backoff* menggunakan nilai nominal produksi (1.500/8.000 ms), beban konkurensi uji rendah (satu uji berurutan) sehingga interaksi dengan konkurensi tinggi belum dievaluasi. *Ketiga*, tidak terpicunya *retry Go* pada AI *Insight* dan AI *Health* bukan menunjukkan kelemahan mekanisme, melainkan konsekuensi langsung dari keberhasilan *fallback* di LangChain sehingga verifikasi *retry Go* pada kedua fitur ini tidak dapat dipisahkan dari strategi *fallback* yang diterapkan.

4.5.2 Evaluasi Efisiensi Representasi Konteks (TOON vs JSON)

Sistem pendukung keputusan yang diusulkan mengandalkan format representasi data *Token-Oriented Object Notation* (TOON) (Schopplich, 2025) untuk mengompresi *payload* konteks sebelum dikirimkan ke LLM. Tabel 4.16 merangkum hasil pengukuran komparatif antara JSON mentah dan TOON pada lima skenario sistem.

Tabel 4.16 Hasil Pengukuran Pengurangan Token Konteks (TOON vs JSON)

Skenario Pengujian	Jumlah Token (JSON)	Jumlah Token (TOON)	Selisih Token	Reduksi
S1 – Rekomendasi Alokasi (*Assign*)	463	256	207	43,8%
S2 – *Insight* (Mode Monitoring)	15.518	6.578	8.940	57,6%
S3 – *Insight* (Mode Non-Monitoring)	1.182	662	520	43,9%
S4 – *Health Evaluation* (*High-Risk*)	11.804	5.042	6.762	57,3%
S5 – *Health Evaluation* (*Low-Risk*)	1.178	632	546	46,3%
Rata-rata Reduksi	6.029	2.634		49,8%

Persentase 49,8% pada baris terakhir dihitung sebagai rata-rata aritmetik dari persentase reduksi per skenario (S1 - S5), bukan dari rasio rata-rata token JSON terhadap rata-rata token TOON. Setiap skenario diukur dengan tiga varian data ($n = 3$), kolom token JSON dan TOON menampilkan rata-rata ketiga varian. Pengukuran menggunakan tokenizer cl100k_base

(tiktoken) sebagai aproksimasi. Kolom token rata-rata (6.029 dan 2.634) hanya disertakan sebagai konteks deskriptif beban *payload absolut*, bukan sebagai dasar penghitungan persentase agregat.

Berdasarkan Tabel 4.16, penggunaan format TOON secara konsisten mengurangi konsumsi token dibandingkan dengan struktur JSON mentah pada kelima skenario yang diuji, dengan rata-rata reduksi sebesar 49,8%.

Penghematan yang paling signifikan terjadi pada Skenario 2 dan Skenario 4, di mana sistem harus melampirkan riwayat metrik historis yang padat seperti data seri waktu per jam selama 7 hari dan per hari selama 60 hari. Pada kedua skenario tersebut, TOON mampu memangkas lebih dari 57% jumlah token-setara dengan penghematan lebih dari 8.900 token per permintaan tunggal pada Skenario 2. Hal ini menunjukkan bahwa arsitektur TOON yang menghilangkan sintaks berulang pada struktur *array tabular* (seperti tanda kutip, kurung kurawal, dan pengulangan nama atribut) berbanding lurus dengan peningkatan efisiensi muatan *prompt* LLM. Dengan efisiensi ini, risiko pemotongan konteks (*context truncation*) pada instruksi *predictive maintenance* dapat diminimalisasi tanpa mengurangi *resource* yang dikirimkan.

Untuk memperjelas batas validitas konstruk pengukuran, sebagai catatan penting, karakteristik struktur data pada kelima skenario. Skenario S1, S3, dan S5 menyajikan konteks berupa objek atribut aset yang disertai daftar ringkas seperti histori isu, histori pemeliharaan, atau kandidat alokasi, sedangkan Skenario S2 dan S4 menyajikan konteks dengan *array tabular* padat berupa deret waktu metrik per jam selama 7 hari dan per hari selama 60 hari. Dengan kata lain, seluruh skenario yang diuji merupakan skenario operasional ITAM penelitian ini yang memang didominasi struktur data tabular atau daftar uniform. Pada struktur data semacam ini, TOON unggul karena menghilangkan redundansi sintaksis pada *array tabular*. Sebaliknya, untuk data yang sangat bersarang (*deeply nested*) atau heterogen dengan banyak objek anonim bersarang, JSON dapat menjadi lebih efisien karena TOON mengandalkan urutan dan penamaan *field* yang tetap. Oleh karena itu, klaim reduksi 49,8% berlaku pada konteks struktur data ITAM penelitian ini dan tidak diklaim sebagai efisiensi universal TOON lintas seluruh struktur data. Keterbatasan ini selaras dengan keterbatasan pengukuran TOON yang dinyatakan pada BAB V.

4.6 Perbandingan Nilai Sistem

Bertolak dari hasil pengujian pada subbab sebelumnya, bagian ini menegaskan nilai tambah sistem penelitian dibanding pendekatan operasional sebelumnya. Perbandingan kondisi sebelum dan sesudah implementasi dirangkum pada Tabel 4.17.

Tabel 4.17 Perbandingan sebelum dan sesudah implementasi

Aspek	Sebelum	Sesudah implementasi
Rekomendasi alokasi aset	Seleksi manual berbasis pengalaman operator	Rekomendasi terurut disertai alasan
Pembacaan kondisi aset	Analisis data mentah per kasus	Narasi *insight* yang ringkas dan operasional
Evaluasi status kesehatan	Tidak terintegrasi menyeluruh	Evaluasi *health* terstruktur dengan *guardrail*
Respons terhadap *event* lapangan	Cenderung reaktif dan manual	Pemicu otomatis dari isu, pemeliharaan, metrik, dan *alert*
Konsistensi status	Berpotensi berbeda antaroperator	Sinkronisasi status kesehatan lebih konsisten

Tabel 4.18 Perbandingan pendekatan *rule-only* dan kombinasi AI + *rule-based*

Kriteria	*Rule-only*	Kombinasi AI + *rule-based*
Konsistensi keputusan	Tinggi	Tinggi
Kemampuan membaca konteks naratif	Rendah	Tinggi
Auditability	Tinggi	Tinggi
Adaptasi terhadap variasi kasus	Terbatas	Lebih baik
Risiko output tidak stabil	Rendah	Terkendali melalui *guardrail* dan validasi

Melalui perbandingan tersebut, terlihat bahwa pendekatan kombinasi menunjukkan potensi memperkuat kualitas interpretasi tanpa mengurangi kontrol keputusan operasional. Perbandingan ini bersifat analitis-konseptual berdasarkan karakteristik pendekatan dan hasil implementasi sistem, bukan hasil eksperimen kuantitatif terpisah terhadap sistem *rule-only*.

4.7 Evaluasi Kualitas Keluaran AI terhadap Keputusan Operasional

4.7.1 Tujuan Evaluasi

Pada tahap ini, evaluasi bertujuan menilai kesesuaian keluaran AI (LLM) terhadap keputusan operasional pada konteks studi kasus BSI UII. Fokus evaluasi tidak diarahkan pada

akurasi statistik model semata, melainkan pada kesesuaian praktis keluaran AI terhadap kebutuhan keputusan harian operator.

Dalam konteks penelitian ini, sistem berfungsi sebagai *Decision Support System* (DSS) dengan pola *DSS human-in-the-loop* (Durachman et al., 2026). Karena itu, kualitas keluaran dinilai sebagai dukungan keputusan, bukan keputusan otomatis.

4.7.2 Metode Evaluasi

Metode yang digunakan Adalah *domain-based expert evaluation*, yaitu evaluasi keluaran AI oleh evaluator yang memiliki relevansi operasional terhadap fitur dan jenis aset yang dinilai. Desain ini digunakan karena tiga fitur AI memiliki domain keputusan yang berbeda. Staff NDC relevan untuk aset *monitorable* dan data monitoring, sedangkan Staff Inventaris relevan untuk aset non-monitoring dan peminjaman. Kepala Tim AMP menjadi evaluator pengelolaan aset yang menjembatani kedua kelompok.

Sebagai hasilnya, penelitian ini tidak memakai desain tiga evaluator untuk seluruh kasus. Kesepakatan antarpenilai dianalisis menggunakan *domain-based expert agreement evaluation* pada pasangan evaluator yang menilai himpunan kasus yang sama. Ukuran utama yang dilaporkan Adalah *percent agreement* per fitur dan aspek penilaian, distribusi skor, serta analisis naratif *disagreement*.

4.7.3 Data Uji dan Prosedur

Evaluasi dilakukan pada 26 kasus yang menggunakan seluruh data final penelitian 4 aset *monitorable*, 6 aset *non-monitorable*, dan 6 persona pengguna.

Komposisi 26 kasus evaluasi mengikuti rancangan pada BAB III Tabel 3.21, yaitu 8 kasus Group A (AMP dan NDC) dan 18 kasus Group B (AMP dan Inventaris). Dari komposisi tersebut, 4 kasus Group A (aset monitoring) menggunakan kombinasi data metrik real dari NDC sebagai *anchor* dengan injeksi sintetis terkontrol untuk skenario batas (CPU *spike*, CPU *critical* berkelanjutan, *memory warning*, dan anomali jaringan), sedangkan 22 kasus Group B (non-monitoring dan *assign*) menggunakan data operasional nyata berupa riwayat *issue*, *maintenance*, dan *pool* laptop tanpa injeksi sintetis. Impikasinya, sebagian besar kasus evaluasi (85%) bersumber dari data operasional nyata, dan data sintetis terbatas pada skenario batas yang tidak tersedia pada data operasional asli.

Prosedur evaluasi:

1. Sistem dijalankan untuk menghasilkan output AI representatif pada 26 kasus.
2. Output AI diringkas ke dalam form *rater* sesuai kondisi *as-is* kasus.
3. Setiap kasus diberikan kepada pasangan evaluator yang relevan secara domain.
4. Evaluator memberi skor 0/1/2 pada tiga aspek penilaian.
5. Skor ditabulasi dan dianalisis menggunakan *percent agreement*, distribusi skor, dan analisis *disagreement*.

Output AI representatif dikumpulkan melalui *endpoint* operasional untuk AI *Assign* dan AI *Insight*. Untuk AI *Health*, penelitian memakai *endpoint* testing internal khusus yang memanggil *pipeline health evaluation* yang sama dengan alur sistem. *Endpoint* ini hanya digunakan sebagai *testing rig* untuk pengambilan output penelitian, bukan fitur operasional pengguna akhir.

4.7.4 Kriteria Penilaian

Kategori penilaian yang digunakan:

a. 2 – Sesuai. Output AI sesuai dengan kondisi kasus dan dapat diterima secara operasional.
b. 1 - Sebagian sesuai. Output AI masih relevan, tetapi memiliki kekurangan minor, kurang lengkap, atau perlu klarifikasi.
c. 0 - Tidak sesuai. Output AI tidak relevan, keliru, atau berpotensi menyesatkan keputusan.

Setiap kasus dinilai pada tiga aspek:

1. Sesuai kondisi kasus, yaitu kesesuaian status, *insight*, atau rekomendasi AI terhadap kondisi aset/pengguna yang diuji.
2. Alasan relevan, yaitu keterhubungan alasan AI dengan data input, riwayat aset, metrik, *alert,* atau kebutuhan pengguna.
3. Keakuratan klaim, yaitu kehati-hatian AI dalam menghindari klaim yang tidak didukung data.

Kondisi yang ditampilkan pada form *rater* adalah kondisi actual apa adanya berdasarkan data input, bukan label hasil AI. Label seperti `*HEALTHY*`, `*WARNING*`, dan `*CRITICAL*` diperlakukan sebagai output AI atau *expected result* yang dinilai.

4.7.5 Output AI Representatif

Output AI untuk 26 kasus berhasil dikumpulkan dan valid secara struktur pada tahap pengambilan data.

Tabel 4.19 Ringkasan output AI representatif

Group	Fitur	Jumlah kasus	Status output	Catatan
A	AI *Health monitoring*	4	4/4 valid	Output *health snapshot* berhasil terbentuk
A	AI *Insight monitoring*	4	4/4 valid	Output *insight* berisi kondisi dan rekomendasi
B	AI *Health non-monitoring*	6	6/6 valid	Output *health* untuk laptop non-monitoring berhasil terbentuk
B	AI *Insight non-monitoring*	6	6/6 valid	Output *insight lifecycle* berhasil terbentuk
B	AI *Assign*	6	6/6 valid	Output rekomendasi aset per persona berhasil terbentuk
Total		26	26/26 valid	

4.7.6 Hasil Evaluasi *Rater* Berbasis *Agreement*

Hasil penilaian rater telah dikumpulkan dari tiga evaluator domain (AMP, NDC, dan Inventaris), dengan setiap kasus dinilai oleh pasangan evaluator yang relevan secara domain. Data penilaian ditabulasi menjadi 156 baris penilaian, sesuai komposisi 26 kasus × 2 evaluator per kasus × 3 aspek. Setiap kombinasi fitur dan aspek kemudian dianalisis menggunakan *percent agreement*, distribusi skor, dan analisis *disagreement* pada pasangan evaluator yang menilai himpunan kasus yang sama. Rekapitulasi data penilaian disimpan sebagai lampiran pendukung untuk menjaga keterlacakan hasil.

Cohen's Kappa sempat dipertimbangkan pada tahap awal, tetapi tidak digunakan sebagai metrik utama karena distribusi skor aktual sangat homogen sehingga nilai Kappa mendekati nol atau tidak terdefinisi meskipun *percent agreement* tinggi fenomena yang dikenal sebagai *Kappa paradox*. Kondisi ini membuat Kappa tidak informatif sebagai ukuran kesepakatan pada data penelitian ini. Oleh karena itu, hasil evaluasi difokuskan pada *percent agreement*, distribusi skor, dan analisis naratif *disagreement* yang lebih langsung menangkap pola kesepakatan evaluator domain.

Tabel 4.20 Hasil *domain-based expert agreement evaluation*

Fitur	**Pasangan evaluator**	**Aspek**	**N kasus**	***Percent agreement***	***Disagreement item***	**Interpretasi**
AI *Health monitoring*	AMP & NDC	Sesuai kondisi kasus	4	1,000	-	Kesepakatan penuh
AI *Health monitoring*	AMP & NDC	Alasan relevan	4	1,000	-	Kesepakatan penuh
AI *Health monitoring*	AMP & NDC	Keakuratan klaim	4	1,000	-	Kesepakatan penuh
AI *Insight monitoring*	AMP & NDC	Sesuai kondisi kasus	4	1,000	-	Kesepakatan penuh
AI *Insight monitoring*	AMP & NDC	Alasan relevan	4	1,000	-	Kesepakatan penuh
AI *Insight monitoring*	AMP & NDC	Keakuratan klaim	4	1,000	-	Kesepakatan penuh
AI *Health non-monitoring*	AMP & Inventaris	Sesuai kondisi kasus	6	1,000	-	Kesepakatan penuh
AI *Health non-monitoring*	AMP & Inventaris	Alasan relevan	6	1,000	-	Kesepakatan penuh
AI *Health non-monitoring*	AMP & Inventaris	Keakuratan klaim	6	0,667	2 kasus laptop dengan riwayat kerusakan fisik	Kesepakatan sebagian
AI *Insight non-monitoring*	AMP & Inventaris	Sesuai kondisi kasus	6	1,000	-	Kesepakatan penuh
AI *Insight non-monitoring*	AMP & Inventaris	Alasan relevan	6	1,000	-	Kesepakatan penuh
AI *Insight non-monitoring*	AMP & Inventaris	Keakuratan klaim	6	1,000	-	Kesepakatan penuh
AI *Assign*	AMP & Inventaris	Sesuai kondisi kasus	6	1,000	-	Kesepakatan penuh
AI *Assign*	AMP & Inventaris	Alasan relevan	6	1,000	-	Kesepakatan penuh

Fitur	Pasangan evaluator	Aspek	N kasus	*Percent agreement*	*Disagreement item*	Interpretasi
AI *Assign*	AMP & Inventaris	Keakuratan klaim	6	1,000	-	Kesepakatan penuh

Secara garis besar, hasil evaluasi menunjukkan penerimaan yang tinggi terhadap keluaran AI oleh evaluator domain. Pasangan AMP dan NDC mencapai *percent agreement* 1,000 pada seluruh aspek monitoring, baik pada AI *Health* maupun AI *Insight*. Pasangan AMP dan Inventaris juga mencapai *percent agreement* 1,000 pada AI Assign, AI *Insight* non-monitoring, serta dua aspek AI *Health non*-monitoring. Perbedaan penilaian hanya muncul pada aspek keakuratan klaim untuk AI *Health non*-monitoring, dengan *percent agreement* 0,667 pada dua kasus laptop dengan riwayat kerusakan fisik dan penggantian komponen.

Nilai *percent agreement* yang tinggi perlu dibaca bersama potensi ceiling effect. Hal ini dapat terjadi karena jumlah kasus per fitur hanya 4 - 6 kasus, kasus yang dipilih merupakan kasus final yang relatif jelas untuk kebutuhan evaluasi domain, dan skala penilaian 0/1/2 bersifat sederhana sehingga perbedaan halus antarpenilai tidak selalu terlihat sebagai *disagreement*. Selain itu, evaluator menilai kasus sesuai domain keahliannya, sehingga peluang kesepakatan pada aspek yang jelas secara operasional menjadi lebih besar. Distribusi skor aktual pada penelitian ini juga sangat homogen dengan dominasi skor 2, sehingga peluang kesepakatan dasar (*base rate*) sudah tinggi tanpa mengacu pada kesepakatan substantif. Kondisi ini merupakan alasan Cohen's Kappa tidak informatif sekaligus keterbatasan yang melekat pada penggunaan percent *agreement* murni, karena metrik ini berisiko melebih-lebihkan reliabilitas sebenarnya ketika distribusi skor sangat skewed. Oleh karena itu, nilai *percent agreement* 1,000 pada 14 dari 15 kombinasi tidak dibaca sebagai bukti reliabilitas yang kuat tanpa kualifikasi, melainkan dibaca bersama distribusi skor dan analisis *disagreement*, bukan sebagai angka reliabilitas tunggal.

Berdasarkan hal tersebut, hasil ini kuat sebagai validasi operasional terbatas terhadap kasus penelitian, tetapi tidak diklaim sebagai generalisasi reliabilitas populasi.

4.7.7 Konsistensi Output AI pada Eksekusi Ulang

Selain evaluasi rater, konsistensi output AI dianalisis melalui eksekusi ulang pada 26 kasus evaluasi final. Pengujian dilakukan sebanyak tiga kali untuk setiap kasus, sehingga total terdapat 78 eksekusi. Pengujian ini menggunakan *prompt template* dan jalur *pipeline* final yang sama dengan keluaran yang disiapkan untuk penilaian rater. Konsekuensinya, uji konsistensi

ini berbeda dari uji eksploratif awal dan digunakan untuk membaca stabilitas keluaran sistem pada input identik.

Metrik konsistensi disesuaikan dengan bentuk keluaran masing-masing fitur. Pada AI *Assign*, konsistensi diukur melalui dua metrik *top-1 consistency* yaitu proporsi eksekusi yang menghasilkan aset rekomendasi peringkat pertama sama dengan keluaran rujukan, dan *top-3 exact consistency* yaitu proporsi eksekusi yang menghasilkan urutan tiga rekomendasi teratas identik dengan rujukan. Pada AI *Health*, konsistensi dibaca dari *status consistency* yaitu proporsi eksekusi yang menghasilkan status kesehatan dominan, serta simpangan baku skor kesehatan antar-eksekusi. Pada AI *Insight*, keluaran berbentuk narasi sehingga konsistensi diukur menggunakan kemiripan teks berbasis *Jaccard Similarity* terhadap gabungan bagian kondisi saat ini, prediksi atau perawatan, dan rekomendasi (Jalal et al., 2022). Konsep *Jaccard Similarity* dan keterbatasannya dalam mengukur kemiripan leksikal narasi LLM dijelaskan pada Subbab 2.8.1.

Tabel 4.21 Ringkasan konsistensi output AI pada eksekusi ulang final

Fitur	Metrik konsistensi utama	Jumlah kasus	Total eksekusi	Valid output	Hasil konsistensi
AI *Assign*	Stabilitas rekomendasi aset peringkat pertama + kestabilan top-3	6	18	18/18 (100%)	Top-1 *consistency* = 1,000, top-3 exact *consistency* = 0,611
AI *Health monitoring*	Proporsi status *health* dominan + simpangan baku skor	4	12	12/12 (100%)	Status *consistency* rata-rata = 0,917, s*core* SD rata-rata = 1,532
AI *Health non-monitoring*	Proporsi status *health* dominan + simpangan baku skor	6	18	18/18 (100%)	Status *consistency* rata-rata = 1,000, *score* SD rata-rata = 2,633
AI I*nsight monitoring*	Kemiripan narasi *insight* berbasis Jaccard	4	12	12/12 (100%)	*Jaccard similarity* rata-rata = 0,321
AI *Insight non-monitoring*	Kemiripan narasi *insight* berbasis Jaccard	6	18	18/18 (100%)	*Jaccard similarity* rata-rata = 0,444

Seluruh 78 eksekusi menghasilkan struktur keluaran valid (Tabel 4.21). Pada AI *Assign*, rekomendasi aset peringkat pertama konsisten pada seluruh kasus uji dengan *top-1 consistency*

mencapai 1,000. Urutan tiga rekomendasi teratas menunjukkan variasi terbatas (*top-3 exact consistency* rata-rata 0,611), yang mencerminkan fleksibilitas model dalam menyusun prioritas alternatif tanpa mengubah rekomendasi utama. Bagi operator, stabilitas peringkat pertama ini memberikan keyakinan bahwa rekomendasi alokasi dapat diandalkan sebagai masukan keputusan awal, sedangkan variasi pada peringkat berikutnya tetap memerlukan pertimbangan kontekstual.

Pada AI *Health*, kasus *non*-monitoring menunjukkan stabilitas status penuh (*status consistency* 1,000) dengan simpangan baku skor rata-rata 2,633. Kasus monitoring juga menunjukkan stabilitas relatif tinggi (*status consistency* rata-rata 0,917), dengan satu variasi yang teridentifikasi pada kasus router dengan anomali jaringan. Pada kasus ini, status kesehatan berubah antara `*WARNING*` dan `*HEALTHY*` antar-eksekusi karena pola *traffic* jaringan yang berada pada batas kategori. Variasi ini tidak memicu *override* dari *rule-based guardrail* ke status `*CRITICAL*` karena anomali yang diuji tidak melampaui ambang *critical* maupun *alert* dengan *severity critical* yang belum *resolved*. Dengan demikian, variasi status sepenuhnya ditentukan oleh evaluasi LLM terhadap konteks siklus hidup, bukan oleh mekanisme *guardrail* berbasis aturan. Temuan ini menunjukkan bahwa kondisi batas pada data monitoring dapat memengaruhi klasifikasi status, sehingga hasil evaluasi kesehatan tetap perlu dibaca bersama penilaian domain dari operator.

Pada AI *Insight*, nilai *Jaccard* rata-rata lebih rendah dibanding fitur lain karena keluaran yang dihasilkan berbentuk narasi rekomendasi. Nilai *Jaccard* 0,321 pada mode monitoring dan 0,444 pada *mode non*-monitoring mencerminkan variasi leksikal yang dapat dijelaskan dari karakter generatif narasi LLM, di mana model dapat merumuskan ulang kalimat dengan pilihan kata berbeda meskipun arah rekomendasinya tetap serupa. Variasi ini lebih tampak pada mode monitoring karena model cenderung menggunakan frasa berbeda saat menginterpretasikan nilai metrik numerik yang sama. Oleh karena itu, *insight* naratif tidak dapat diperlakukan sebagai teks pelaporan formal tanpa verifikasi operator, dan hasil uji konsistensi ini tidak dibaca sebagai pengganti penilaian domain melainkan sebagai bukti tambahan bahwa *pipeline* AI final menghasilkan keluaran valid dengan substansi yang perlu tetap diverifikasi manusia.

Secara keseluruhan, variasi yang teridentifikasi pada ketiga fitur tidak menunjukkan inkonsistensi substansial melainkan refleksi dari sifat generatif LLM dan kerentanan kasus batas (*boundary cases*) terhadap perbedaan penekanan *reasoning* model. Implikasi operasionalnya adalah keluaran AI pada penelitian ini berfungsi sebagai masukan keputusan

yang tetap memerlukan tinjauan operator, khususnya pada kasus dengan kondisi batas atau keluaran naratif yang memerlukan interpretasi kontekstual.

Apabila hasil ini dibaca terhadap ambang dapat diterima yang ditetapkan pada Tabel 3.18, ketiga fitur memenuhi ambang operasionalnya. AI *Assign* mencapai top-1 *consistency* 1,000 (ambang ≥ 0,80), AI *Health* mencapai status *consistency* 0,917-1,000 (ambang ≥ 0,90), dan AI *Insight* mencapai Jaccard 0,321–0,444 (ambang ≥ 0,30). Perlu ditegaskan bahwa pemenuhan ambang ini dibaca sebagai indikasi stabilitas operasional, bukan sebagai jaminan reliabilitas populasi. AI *Health* monitoring berada tepat di atas ambang dengan satu kasus router yang berfluktuasi pada kondisi batas, sehingga margin terhadap ambang tergolong tipis dan menegaskan perlunya verifikasi operator pada kasus semacam ini.

4.7.8 Analisis *Disagreement Rater*

Analisis *disagreement* dilakukan untuk menelusuri perbedaan penilaian antarevaluator. Hasil konversi skor menunjukkan bahwa perbedaan penilaian hanya terjadi pada fitur AI *Health non*-monitoring, aspek keakuratan klaim, yaitu pada dua kasus laptop dengan riwayat kerusakan fisik dan penggantian komponen. Tidak ditemukan *disagreement* pada fitur AI Assign, AI *Insight monitoring*, AI *Insight non-monitoring*, maupun AI *Health monitoring*. Rincian pola *disagreement* dan implikasinya disajikan pada Tabel 4.22.

Tabel 4.22 Analisis *disagreement rater*

Fitur	Kasus/aspek	Pola *disagreement*	Kemungkinan penyebab	Implikasi perbaikan
AI *Health monitoring*	Tidak ada	AMP dan NDC memberi skor sama pada seluruh kasus/aspek	Output monitoring dianggap selaras oleh perspektif pengelolaan aset dan monitoring	Pertahankan struktur konteks monitoring dan validasi output
AI *Insight monitoring*	Tidak ada	AMP dan NDC memberi skor sama pada seluruh kasus/aspek	Narasi *insight* monitoring dianggap relevan dan tidak memuat klaim yang bertentangan dengan data	Pertahankan pemisahan *mode monitoring* dan batasan klaim metrik
AI *Health non-monitoring*	Laptop dengan riwayat kerusakan / Keakuratan klaim	AMP = 2, Inventaris = 1	Evaluator Inventaris menilai ada ketidakakuratan temporal pada tanggal laporan kerusakan dan	Perbaiki penyusunan konteks histori agar tanggal *issue*, tanggal perbaikan,

Fitur	Kasus/aspek	Pola *disagreement*	Kemungkinan penyebab	Implikasi perbaikan
			pembaruan data perbaikan, AMP menilai klaim kondisi masih sesuai secara operasional	dan tanggal pembaruan data dibedakan eksplisit
AI *Health non-monitoring*	Laptop dengan riwayat penggantian baterai / Keakuratan klaim	AMP = 2, Inventaris = 1	Evaluator Inventaris menilai tanggal penggantian baterai tidak akurat karena servis dan penggantian baterai terjadi pada Oktober 2024	Tambahkan *guardrail* agar AI tidak menyimpulkan tanggal tindakan teknis jika *timestamp* pada histori tidak eksplisit atau ambigu
AI *Insight non-monitoring*	Tidak ada	AMP dan Inventaris memberi skor sama pada seluruh kasus/aspek	*Insight lifecycle* laptop dianggap sesuai dengan riwayat *issue/maintenance* dan konteks inventaris	Pertahankan format ringkasan as-is dan batasan klaim non-monitoring
AI *Assign*	Tidak ada	AMP dan Inventaris memberi skor sama pada seluruh kasus/aspek	Rekomendasi alokasi dan alasan dianggap sesuai dengan kebijakan aset serta kesiapan operasional peminjaman	Pertahankan penyajian alasan rekomendasi dan validasi kandidat *borrowable*

Perbedaan penilaian pada dua kasus laptop tersebut menunjukkan bahwa evaluator Inventaris lebih sensitif terhadap akurasi detail histori fisik aset, khususnya tanggal laporan kerusakan, tanggal servis, dan tanggal penggantian komponen. Sebaliknya, evaluator AMP cenderung menilai dari kesesuaian keputusan pengelolaan aset secara umum. Temuan ini menjadi masukan perbaikan untuk *pipeline* AI *Health non-monitoring* agar konteks tanggal dan urutan kejadian ditulis lebih eksplisit serta tidak ditafsirkan berlebihan oleh model.

4.7.9 Hasil *Feedback* Operator terhadap Dukungan Keputusan

Feedback operator dikumpulkan melalui sesi demo terpandu dengan Kepala Tim AMP pada 25 Juni 2026. Sesi mencakup *walkthrough* lima skenario bisnis Tabel 3.20 menggunakan *frontend* untuk modul inti aset serta fitur AI *Assign*, AI *Insight*, dan AI *Health*, dan *endpoint* API untuk integrasi *monitoring-alert*. *Dashboard* ringkasan operasional belum tersedia pada

saat sesi, namun tidak menjadi bagian dari lima skenario bisnis yang dievaluasi sehingga tidak memengaruhi penilaian. Setelah sesi, AMP mengisi instrumen Likert 1-5.

Instrumen *feedback* operator berbeda dari form evaluasi kualitas keluaran AI, *feedback* operator menilai persepsi kegunaan sistem secara umum, sedangkan evaluasi kualitas AI menilai output AI per kasus. Ketujuh pernyataan Likert difokuskan pada persepsi dukungan keputusan AI, bukan antarmuka pengguna, sehingga penilaian tertuju pada substansi fitur AI yang telah berfungsi.

Dengan n = 1, hasil *feedback* dibaca sebagai indikasi persepsi awal pengambil keputusan strategis (Kepala Tim AMP), bukan generalisasi terhadap populasi operator.

Tabel 4.23 Hasil *feedback* operator

Pernyataan	Aspek	Skor
Sistem membantu operator menemukan informasi aset yang dibutuhkan	Efisiensi	5
AI *Assign* membantu prioritas kandidat aset	DSS	4
AI *Insight* membantu memahami kondisi aset	Interpretasi	4
AI *Health* membantu prioritas tindak lanjut	Evaluasi risiko	4
Alasan rekomendasi dapat diverifikasi	*Auditability*	5
Keputusan akhir tetap pada operator	*Human-in-the-loop*	5
Sistem relevan terhadap kebutuhan operasional harian	Relevansi operasional	4
Rata-rata		4,4

Berdasarkan Tabel 4.23, Kepala Tim AMP memberikan skor rata-rata 4,4 dari 5 pada tujuh aspek evaluasi DSS. Tiga aspek memperoleh skor tertinggi yaitu efisiensi pencarian informasi aset, verifiabilitas alasan rekomendasi, dan kendali keputusan operator (*human-in-the-loop*).

Skor 5 pada aspek verifiabilitas alasan selaras dengan capaian *expert agreement* yang menunjukkan bahwa alasan AI dapat ditelusuri ke data input. Empat aspek lainnya dukungan AI *Assign*, interpretasi AI *Insight*, prioritas tindakan AI *Health*, dan relevansi operasional memperoleh skor 4, mengindikasikan persepsi positif dengan ruang perbaikan. Tidak ada aspek yang memperoleh skor di bawah 4, menunjukkan bahwa menurut persepsi responden ini, tidak ada domain yang dinilai kurang atau tidak membantu.

Hasil ini tidak dimaksudkan sebagai pengganti *expert agreement* maupun metrik konsistensi, melainkan sebagai dimensi evaluasi pelengkap yang melaporkan persepsi penggunaan sistem dari sudut pandang pengambil keputusan strategis.

4.7.10 Keterbatasan LLM pada Kesimpulan Evaluasi

LLM memiliki keterbatasan berupa potensi *hallucination* dan variasi keluaran akibat sifat probabilistik model (Huang et al., 2025). Variasi yang muncul terutama berada pada frasa naratif dan urutan kandidat setara, bukan pada perpindahan keputusan ke kategori tidak sesuai.

Dengan pendekatan *human-in-the-loop* dan *guardrail* berbasis aturan domain, keluaran AI berfungsi sebagai masukan keputusan operasional yang tetap diverifikasi operator (Durachman et al., 2026). Hasil evaluasi *rater* menunjukkan bahwa keluaran AI diterima oleh evaluator domain pada mayoritas kasus, dengan *percent agreement* 1,000 pada 14 dari 15 kombinasi fitur-aspek. Satu kombinasi yang tidak mencapai kesepakatan penuh adalah AI *Health non-monitoring* pada aspek keakuratan klaim, sehingga perbaikan utama diarahkan pada akurasi detail temporal histori aset. Hasil subbab ini menunjukkan potensi dukungan keputusan operasional, dengan catatan diperlukan evaluasi lanjutan pada skala data, pengguna, dan periode pengamatan yang lebih besar.

4.8 Rekam Jejak Penyempurnaan Artefak berdasarkan Siklus DSR

Sesuai dengan paradigma *Design Science Research* (Peffers et al., 2007), pengembangan artefak pada penelitian ini dijalankan melalui siklus *design–evaluate–refine* yang bersifat iteratif. Artefak yang dihasilkan bukan merupakan keluaran desain tunggal, melainkan produk dari serangkaian evaluasi teknis yang mendorong penyempurnaan bertahap. Pada penelitian ini, terdapat tiga keputusan desain utama yang mencerminkan iterasi artefak secara substantif.

4.8.1 Iterasi 1: Evolusi Format JSON ke TOON

Pada tahap awal perancangan, konteks operasional aset yang dikirimkan ke komponen LLM disusun menggunakan format JSON. Pilihan ini dipandang sebagai pendekatan yang paling langsung untuk implementasi awal karena format JSON merupakan format pertukaran data yang umum digunakan di seluruh lapisan sistem.

Pada fase evaluasi awal, ditemukan bahwa struktur JSON bersifat deklaratif dan menghasilkan *overhead* token yang tidak proporsional terhadap kandungan informasi yang dibawa. Fenomena ini terutama terlihat pada konteks yang memuat data *time-series* dari

‘asset_metric_hourly’, di mana setiap baris data mengulang *key repetition* dan tanda baca struktural secara redundan. Pada konteks *insight* dan *health evaluation* yang memuat riwayat metrik tujuh hari (±168 baris), beban token yang dihasilkan jauh melebihi kebutuhan informasional yang setara.

Temuan ini mendorong pencarian format representasi konteks yang lebih efisien untuk data tabular. Setelah meninjau beberapa alternatif, penelitian ini mengadopsi *Token-Oriented Object Notation* (TOON) (Schopplich, 2025), yaitu format *open-source* yang dirancang untuk mengeliminasi redundansi sintaksis pada struktur data tabular tanpa mengurangi kelengkapan informasi yang diterima LLM. Hasil evaluasi komparatif menunjukkan rata-rata pengurangan konsumsi token sebesar 49,8% di seluruh skenario pengujian, dengan reduksi tertinggi mencapai 57,6% pada konteks *insight* mode monitoring (Tabel 4.16). Penyempurnaan ini tidak hanya berdampak pada efisiensi komputasi, tetapi juga memperluas kapasitas konteks yang tersedia untuk instruksi analitis dalam prompt tanpa risiko pemotongan konteks (*context truncation*). Implikasi metodologis dari iterasi ini adalah bahwa pemilihan format representasi konteks bukan sekadar keputusan teknis implementasi, melainkan bagian dari desain sistem pendukung keputusan berbasis LLM yang memengaruhi efisiensi, keterlacakan, dan kualitas masukan model.

4.8.2 Iterasi 2: Stratifikasi Kelas Risiko *Health Evaluation*

Pada rancangan awal pipeline AI *Health Evaluation*, mekanisme evaluasi kesehatan dirancang dengan satu skema evaluasi yang berlaku seragam untuk seluruh aset. Pada fase perancangan yang lebih mendalam, identifikasi kelemahan struktural dilakukan melalui pembedaan karakteristik risiko operasional antara aset infrastruktur kritis (seperti server dan perangkat jaringan inti) dan aset umum (seperti workstation dan periferal) memiliki konsekuensi kegagalan yang tidak setara. Menerapkan satu skema evaluasi yang seragam berarti memberikan toleransi penilaian yang sama pada dua konteks dengan dampak kegagalan yang berbeda secara fundamental.

Pertimbangan tersebut mendorong desain ulang pipeline menggunakan pendekatan stratifikasi kelas risiko, yaitu *high-risk* dan *low-risk*, yang ditetapkan berdasarkan karakteristik aset. Stratifikasi ini mengikuti rancangan klasifikasi risiko pada BAB III, yaitu aset *high-risk* direpresentasikan oleh aset dengan penanda monitoring aktif dan konsekuensi gangguan lebih luas, sedangkan aset *low-risk* direpresentasikan oleh aset operasional pengguna akhir yang tidak memiliki monitoring langsung. Dengan rancangan tersebut, pemisahan kelas risiko tidak

berdiri sebagai label administratif, tetapi menjadi dasar pemilihan perlakuan evaluasi pada pipeline AI *Health Evaluation*.

Pada aset *high-risk*, evaluasi dirancang melalui skema *hybrid* yaitu komponen LLM mengevaluasi faktor siklus hidup dan konteks historis, sementara lapisan *rule-based guardrail* beroperasi secara deterministik untuk memvalidasi dan, jika diperlukan, menimpa (*override*) status kesehatan final berdasarkan kondisi operasional yang tidak dapat dikompromikan. Pada aset *low-risk*, LLM berperan lebih dominan dalam menyusun status kesehatan, meskipun keluarannya tetap divalidasi terhadap referensi status domain pada masterdata sistem. Pola ini selaras dengan model pengambilan keputusan pada BAB III, yaitu kondisi *critical* pada aset *high-risk* tidak boleh diturunkan oleh keluaran LLM, sedangkan aset *low-risk* tetap memerlukan validasi status agar hasil model tidak keluar dari domain sistem.

Implikasi operasional dari iterasi ini adalah bahwa evaluasi kesehatan aset tidak seharusnya menggunakan tingkat kontrol yang sama untuk seluruh kelas aset, karena konsekuensi kesalahan penilaian pada aset kritis berbeda dari aset operasional umum. Stratifikasi risiko membuat desain sistem lebih selaras dengan prinsip manajemen aset berbasis risiko (Albalghouni et al., 2018). Pembedaan antara aset infrastruktur kritis dan aset umum ini sejalan dengan distinksi CAPEX dan OPEX yang diuraikan pada Subbab 2.1.3, di mana aset bernilai investasi tinggi dengan konsekuensi gangguan luas memerlukan kontrol evaluasi yang lebih ketat dibanding aset operasional rutin berdampak terbatas.

4.8.3 Iterasi 3: Mekanisme *Caching* untuk efisiensi

Pada perancangan awal fitur AI *Asset Insight,* setiap permintaan *insight* oleh pengguna memicu pemanggilan *inference* ke komponen LLM secara langsung. Pendekatan ini menjamin aktualitas hasil, namun menimbulkan dilema desain. Disatu sisi, pemanggilan *inference* berulang pada aset yang kondisi operasionalnya tidak berubah secara signifikan mengakibatkan biaya latensi dan komputasi yang tidak proporsional, di sisi lain, mekanisme *rate-limiting* dari penyedia layanan model eksternal menjadi hambatan nyata ketika frekuensi akses meningkat.

Untuk menyelesaikan dilema ini, desain artefak dievolusikan dengan mengintegrasikan mekanisme *caching* hasil *insight* pada entitas ‘asset_ai_insight_cache’. Setiap hasil *insight* yang berhasil di-*generate* disimpan dan dijadikan sumber respons pada akses berikutnya, selama nilai ‘expires_at’ (TTL berbasis waktu) belum terlampaui dan ‘flag_aktif’ masih bernilai aktif. Hasilnya, sistem dapat menyajikan *insight* dengan latensi yang jauh lebih rendah pada kondisi normal, sambil tetap memperbarui konteks melalui *inference* baru ketika *cache*

sudah kedaluwarsa atau diinvalidasi secara eksplisit. Desain ini mencerminkan prinsip keseimbangan antara aktualitas informasi (*freshness*) dan efisiensi sumber daya (*resource efficiency*) dalam rancangan *Decision Support System* yang beroperasi dengan ketergantungan pada layanan eksternal. Implikasi desain dari iterasi ini adalah bahwa sistem pendukung keputusan berbasis LLM perlu menyeimbangkan aktualitas informasi, efisiensi pemanggilan model, dan ketersediaan layanan. *Caching* dengan TTL eksplisit menjadi mekanisme pragmatis untuk menjaga ketiga kebutuhan tersebut secara proporsional.

Ketiga iterasi tersebut mengilustrasikan bahwa proses perancangan artefak pada penelitian ini tidak bersifat linear, melainkan bersifat reflektif terhadap temuan teknis dan operasional yang muncul sepanjang siklus implementasi. Melalui rekam jejak penyempurnaan ini, artefak dapat dipandang bukan sekadar sebagai prototipe fungsional, melainkan sebagai artefak DSR yang dikalibrasi secara metodologis untuk memenuhi kriteria efisiensi representasi konteks, konsistensi evaluasi operasional, dan keseimbangan aktualitas dengan efisiensi komputasi.

4.9 Analisis Keterbatasan dan Risiko

Walaupun hasil evaluasi menunjukkan kecenderungan positif, pembacaan akademik yang utuh tetap perlu memuat batasan serta risiko implementasi yang teridentifikasi selama penelitian.

4.9.1 Kompatibilitas Integrasi *Endpoint*

Pada fase transisi layanan, *endpoint insight* mode baru sempat tidak tersedia sehingga diperlukan *fallback* ke *endpoint* sebelumnya.

- **Dampak terhadap validitas hasil**. hasil integrasi tidak dapat ditafsirkan sebagai kondisi *endpoint* stabil permanen karena dipengaruhi versi layanan saat pengujian.
- **Mitigasi yang sudah dilakukan.** *fallback* ke *endpoint legacy* diaktifkan dan diverifikasi melalui pengujian validasi respons *error* dasar (Tabel 4.13).
- **Rekomendasi penelitian lanjutan.** terapkan *version contract testing* dan *staging gate* sebelum *endpoint* baru dipromosikan ke lingkungan operasional.

4.9.2 Ketergantungan pada Layanan AI Eksternal

Ketergantungan pada layanan AI eksternal tetap menjadi risiko pada lonjakan permintaan.

- **Dampak terhadap validitas hasil**. performa latensi dan tingkat *error* sangat dipengaruhi kualitas koneksi dan stabilitas layanan pihak ketiga saat pengujian.
- **Mitigasi yang sudah dilakukan**. diterapkan interval *request*, *retry* terbatas, dan *exponential backoff* pada *backend* mekanisme *retry* dua *layer* (Langchain *service* dan Go *backend*) telah diverifikasi secara terbatas melalui *faulit injection* 500/429/504 pada 45 uji (Tabel 4.14 dan Tabel 4.15) tanpa kegagalan fatal.
- Tabel 4.15**Rekomendasi penelitian lanjutan**. tambahkan *circuit breaker* dan *degraded mode* terukur untuk mempertahankan kontinuitas layanan saat gangguan eksternal berkepanjangan.

4.9.3 Konsistensi Output AI pada Input Identik

Output AI berpotensi bervariasi walaupun konteks input identik. Pada penelitian ini setiap kasus evaluasi AI dijalankan ulang tiga kali pada 26 kasus final, sehingga menghasilkan 78 eksekusi total. Pengukuran ini digunakan untuk membaca variasi keluaran pada *prompt* dan *pipeline* final yang sama dengan pengambilan output *rater.*

- **Dampak terhadap validitas hasil**. keluaran AI tidak sepenuhnya deterministik, sehingga interpretasi hasil perlu mempertimbangkan variasi antareksekusi.
- **Mitigasi yang sudah dilakukan**. pengukuran konsistensi kuantitatif (*Overlap*/Jaccard/status identik) dan penerapan *guardrail* aturan domain pada AI *Health*.
- **Rekomendasi penelitian lanjutan**. lakukan uji konsistensi pada 100+ eksekusi per fitur serta perbandingan antarmodel LLM untuk melihat stabilitas lintas konfigurasi.

4.9.4 Keterbatasan Data *Event*

Event alert tanpa identitas aset tidak diproses ke evaluasi *health* untuk mencegah salah pembaruan status kesehatan.

- **Dampak terhadap validitas hasil.** sebagian kejadian lapangan dengan data label tidak lengkap tidak ikut terhitung pada pembaruan health otomatis.
- **Mitigasi yang sudah** dilakukan. *event invalid* diabaikan secara eksplisit dan dicatat di log agar tidak mengganggu *pipeline* utama.
- **Rekomendasi penelitian lanjutan**. tambah mekanisme *data enrichment* atau korelasi lintas sumber agar *event* tanpa ‘asset_uuid’ tetap dapat ditelusuri secara aman.

4.9.5 Batas Generalisasi Implementasi

Hasil implementasi sangat dipengaruhi oleh struktur data, alur kerja, dan tata kelola status pada organisasi studi kasus.

- Dampak terhadap validitas hasil. temuan tidak dapat digeneralisasi langsung ke organisasi lain tanpa adaptasi domain.
- Mitigasi yang sudah dilakukan hasil diperlakukan sebagai evaluasi operasional terbatas pada BSI UII, bukan klaim universal lintas organisasi.
- Rekomendasi penelitian lanjutan lakukan replikasi pada lebih dari satu unit/institusi, termasuk pengujian beban produksi riil, untuk memperkuat validitas eksternal.

Secara praktis, adopsi pada institusi lain akan lebih realistis jika memenuhi prasyarat minimal seperti data aset telah terdigitalisasi dengan histori operasional, tersedia infrastruktur monitoring yang setara (misalnya Prometheus/Grafana) untuk aset monitoring, dan alur keputusan operasional masih memungkinkan validasi manusia sebelum eksekusi final. Tanpa prasyarat tersebut, kualitas konteks AI dan konsistensi status kesehatan berisiko menurun. Sebagai penutup, Tabel 4.24 memetakan setiap indikator keberhasilan terhadap bukti hasil kuantitatif yang telah dilaporkan.

Tabel 4.24 Pemetaan indikator keberhasilan terhadap bukti hasil

Indikator keberhasilan	***Threshold target***	**Hasil terukur pada Bab IV**	**Status**
Fungsi utama berjalan	5/5 skenario bisnis lulus	5/5 skenario bisnis utama dari Tabel 3.20 tervalidasi melalui Subbab 4.2, Tabel 4.5, Tabel 4.6, dan Tabel 4.8.	Tercapai pada ruang uji terbatas
Integrasi antarlayanan berjalan	Keberhasilan integrasi ≥ 95%	5/5 integrasi utama pada Tabel 4.8 berhasil (100%)	Tervalidasi terbatas
Kesesuaian operasional output AI	Struktur output valid ≥ 90% eksekusi uji	Struktur output AI representatif valid pada 26/26 kasus evaluasi (Tabel 4.19), *percent agreement rater* 1,000 pada 14/15 kombinasi fitur-aspek dan 0,667 pada 1 kombinasi (Tabel 4.20)	Terpenuhi pada skenario uji terbatas

Indikator keberhasilan	*Threshold target*	Hasil terukur pada Bab IV	Status
Konsistensi output AI	Metrik dilaporkan per kasus uji ulang	Top-1 = 1,000 (*Assign*), status *consistency* 0,917-1,000 (*Health*), Jaccard 0,321-0,444 (*Insight*), lihat Tabel 4.21.	Terdokumentasi
Konsistensi status kesehatan	100% *mapping* 'final_health_status' ke referensi status	Tidak ditemukan konflik mapping pada skenario *health* yang diuji	Tercapai pada ruang uji terbatas
Keandalan mekanisme integrasi	100% skenario *fallback* terverifikasi tanpa fatal *error*	4/4 skenario validasi respons *error* dasar berjalan terkendali, *fault* layanan eksternal 500/*timeout*/429 belum dieksekusi penuh	Tervalidasi terbatas
Relevansi operasional	Bukti dukungan keputusan dan relevansi harian tersedia	Didukung melalui pengujian skenario bisnis, validitas output AI 26/26, konsistensi output 78/78 valid, evaluasi rater domain, dan feedback operator terbatas 1 responden (AMP, Tabel 4.23)	Terpenuhi terbatas tanpa *feedback Likert*

Bukti relevansi operasional didukung oleh *feedback* operator terbatas yaitu Kepala Tim AMP yang dilaporkan lengkap pada Subbab 4.7.9. Cakupan *interface* pada sesi demo meliputi modul inti aset dan ketiga fitur AI, *dashboard* ringkasan operasional belum tersedia pada saat sesi sehingga belum dievaluasi.

4.10 Pembahasan Hasil dan Keterkaitan Teori

Sebagai sintesis atas temuan empiris pada subbab sebelumnya, hasil penelitian menunjukkan bahwa sistem berpotensi mendukung pergeseran dari fungsi pencatatan menuju dukungan keputusan operasional. Berdasarkan evaluasi domain dan pengujian teknis yang tersedia, temuan ini berpotensi selaras dengan konsep IT *Asset Management* pada Bab II, yaitu

pengelolaan siklus hidup aset berbasis data dan ketertelusuran Keputusan (Maletič et al., 2020; Stone et al., 2018).

Pada perspektif *Decision Support System*, AI *Assign Recommendation* dan AI *Asset Insight* menunjukkan bahwa sistem berperan sebagai penyedia rekomendasi dan interpretasi, bukan sekadar penyimpanan data (Rashidi et al., 2018). Hal ini mengindikasikan pergeseran dari keputusan berbasis pengalaman individual ke keputusan yang lebih terstruktur (Rashidi et al., 2018).

Pada perspektif AI, hasil penelitian menegaskan bahwa model perlu ditempatkan dalam kerangka kontrol organisasi (Dong et al., 2024; Rashidi et al., 2018). AI *Health Evaluation* dengan *guardrail* memperlihatkan keseimbangan antara kemampuan analitik model dan batas keputusan operasional. Kondisi ini selaras dengan prinsip *condition-based monitoring* serta pendekatan *DSS human-in-the-loop*, sehingga keluaran AI tetap berperan sebagai rekomendasi dan bukan keputusan otomatis (Durachman et al., 2026).

Kontribusi penelitian ini tidak berhenti pada pembangunan fitur aplikasi, tetapi pada integrasi metode AI dan *rule-based* ke dalam alur operasional nyata dengan mekanisme kontrol yang jelas. *Guardrail* pada evaluasi *health*, validasi struktur output, *retry-backoff* terbatas, serta kebijakan skip pada *event* tanpa identitas aset menjadi bentuk kontrol agar sistem tetap adaptif namun tidak kehilangan akuntabilitas. Sebagai implikasinya, kontribusi utama penelitian terletak pada rancangan dan evaluasi artefak integrasi AI dan *rule-based* yang operasional, terukur pada ruang uji terbatas, dan dapat diadaptasi pada organisasi lain dengan penyesuaian struktur data serta kebijakan operasional (Peffers et al., 2007).

Analisis *trade-off* desain menegaskan alasan pemilihan pendekatan *hybrid* pada konteks implementasi penelitian ini. Pada konteks tersebut, pendekatan *rule-only* lebih unggul pada kebutuhan keputusan yang sangat stabil, deterministik, dan mudah diaudit, tetapi kurang adaptif saat konteks kasus berubah (Reddy & Fields, 2022; Tan, 2017). Pendekatan AI-*only* lebih unggul pada fleksibilitas interpretasi dan kemampuan membaca konteks naratif, namun berisiko pada konsistensi output dan kontrol keputusan. Pendekatan *hybrid* yang digunakan pada penelitian ini menambah kompleksitas desain dan integrasi. Meski demikian, pendekatan ini memberikan titik seimbang antara adaptivitas AI dan kepastian aturan operasional, sebagaimana tercermin pada hasil perbandingan di Tabel 4.18.

Secara keseluruhan, temuan pada subbab 4.2 sampai 4.7 mengindikasikan bahwa kombinasi AI dan *rule-based* pada konteks pengelolaan aset TI menunjukkan potensi manfaat praktis sekaligus menjaga aspek akuntabilitas keputusan. Keterkaitan dengan rumusan masalah

Bab I juga tampak eksplisit yaitu masalah data tidak terstruktur dijawab melalui AI *Insight*, keputusan manual dijawab melalui AI *Assign*, dan ketidakjelasan *lifecycle* aset dijawab melalui AI *Health Evaluation*.

Jika dikaitkan secara konseptual dengan prinsip ISO 55000 (*plan-do-check-act*) (Albalghouni et al., 2018), hasil penelitian ini menunjukkan pola siklus yang lebih utuh. Perencanaan dan penetapan aturan dilakukan pada desain domain (*plan*), eksekusi operasional berjalan melalui integrasi fitur inti (*do*), evaluasi kondisi aset berlangsung melalui *insight/health* dan pengujian berkala (*check*), serta tindak lanjut perbaikan didorong oleh rekomendasi operasional dan pembaruan status terstruktur (*act*). Keterkaitan ini bersifat konseptual dan tidak dimaksudkan sebagai hasil audit kepatuhan ISO 55000.

Dari perspektif AIOps, integrasi monitoring dan *alert* pada sistem ini menunjukkan bentuk awal otomatisasi deteksi kondisi operasional dan pemicu evaluasi berbasis data metrik (Notaro et al., 2021; Zhaoxue et al., 2021). Meskipun lingkupnya terbatas pada konteks manajemen aset pascaakuisisi dan belum mencakup spektrum penuh operasi TI seperti yang dijelaskan pada survei AIOps, mekanisme pemicu *health* dari *event* monitoring, *alert*, dan jadwal berkala mencerminkan prinsip dasar AIOps dalam mengurangi ketergantungan pada deteksi manual.

Temuan pada AI *Insight* dan AI *Health* juga memperkuat pembacaan *lifecycle* aset TI pada fase pascaakuisisi (Stone et al., 2018). Setelah aset dicatat, sistem tetap memantau riwayat isu (*issue*), pemeliharaan (*maintenance*), metrik, dan *alert* untuk menilai perubahan kondisi sepanjang umur pakai aset (Maletič et al., 2020). Akibatnya, pemantauan tidak berhenti pada inventarisasi, tetapi berlanjut pada evaluasi kesehatan aset secara periodik.

Dari perspektif nilai strategis organisasi, integrasi data monitoring dan histori kerusakan memberikan dasar untuk membaca tren gangguan berulang, menilai kecukupan cadangan aset (cukup/berlebih/kurang), dan menyusun prioritas perencanaan Rencana Kerja dan Anggaran Tahunan (RKAT) berbasis risiko layanan. Dalam konteks ini, aset TI tidak hanya dipandang sebagai objek operasional harian, tetapi juga sebagai komponen strategis yang berkaitan langsung dengan kontinuitas layanan organisasi (Gavrikova et al., 2020; Rahadi et al., 2024).

4.10.1 Implikasi Praktis untuk Operasional BSI UII

Implikasi praktis penelitian menunjukkan indikasi peningkatan keterarahan identifikasi masalah aset dan konsistensi pencatatan status kesehatan pada ruang uji terbatas, serta

penguatan dasar pengambilan keputusan harian. Dampak tersebut relevan untuk lingkungan operasional yang membutuhkan respons cepat dengan jejak keputusan yang dapat diaudit.

Manfaat ini tampak pada kecenderungan proses triase yang lebih terarah. Berdasarkan wawancara informal dengan personel operasional BSI UII, proses seleksi kandidat aset secara manual umumnya memerlukan waktu lebih dari lima menit per permintaan, sedangkan AI *Assign* menyajikan prioritas dan alasan dalam satu keluaran sehingga titik awal pengambilan keputusan menjadi lebih terarah, potensi pemendekan waktu ini bersifat indikasi kualitatif yang belum dikonfirmasi secara kuantitatif karena pengukuran terstruktur pada Tabel 4.3 belum dieksekusi dalam lingkup penelitian ini. Selain itu, sistem menunjukkan prioritas tindak lanjut yang lebih jelas serta kesiapan data untuk pembahasan kebutuhan pemeliharaan dan penggantian aset. Namun demikian, hasil ini belum merepresentasikan produksi dan perlu diverifikasi kembali pada volume data, variasi operator, dan variasi organisasi yang lebih besar.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berangkat dari kebutuhan BSI UII untuk memperkuat pengelolaan aset TI pascaakuisisi, khususnya pada aspek keterpaduan data operasional, dukungan keputusan alokasi, interpretasi kondisi aset, dan evaluasi kesehatan aset. Melalui pengembangan **Sistem Pendukung Keputusan Manajemen Aset TI Berbasis LLM**, penelitian ini menunjukkan bahwa integrasi antara data aset, histori operasional, monitoring-*alert*, LLM, dan aturan domain dapat membentuk artefak pendukung keputusan yang lebih terstruktur, meskipun tetap berada pada ruang uji terbatas dan tidak menggantikan keputusan manusia.

Berdasarkan perancangan, implementasi, pengujian, dan pembahasan pada BAB III dan BAB IV, kesimpulan penelitian disusun untuk menjawab rumusan masalah pada BAB I sebagai berikut.

1. **Integrasi proses pascaakuisisi aset TI**

   Sistem berhasil mengintegrasikan proses internal pascaakuisisi BSI UII melalui modul masterdata, aset, peminjaman dan pengembalian, pencatatan isu, pemeliharaan, integrasi monitoring-*alert*, serta fitur AI. SAP tetap berperan sebagai sistem akuisisi dan pencatatan awal aset pada level universitas, sedangkan sistem yang dikembangkan berfokus pada pengelolaan operasional pascaakuisisi di BSI UII. Pada penelitian ini, data referensi SAP digunakan melalui input manual berkas Excel sebagai *baseline* data, bukan melalui integrasi transaksi dua arah.

2. **Dukungan keputusan berbasis LLM**

   Fitur AI *Assign Recommendation*, AI *Asset Insight*, dan AI *Health Evaluation* berhasil menghasilkan keluaran terstruktur yang dapat digunakan sebagai dukungan keputusan operasional terbatas. Validitas struktur keluaran AI representatif mencapai 26/26 kasus. Evaluasi kualitas keluaran AI dilakukan menggunakan pendekatan *domain-based expert agreement evaluation* pada 26 kasus final, dengan pasangan evaluator sesuai domain AMP dan NDC untuk aset monitoring serta AMP dan Inventaris untuk aset non-monitoring dan skenario peminjaman. Hasil evaluasi menunjukkan *percent agreement* sebesar 1,000 pada 14 dari 15 kombinasi fitur-aspek. Satu kombinasi yang

tidak mencapai kesepakatan penuh adalah AI *Health non-monitoring* pada aspek keakuratan klaim, dengan *percent agreement* 0,667. Interpretasi dilakukan menggunakan *percent agreement,* distribusi skor, dan analisis *disagreement.* Konsistensi keluaran AI pada eksekusi ulang, sebagai salah satu indikator kualitas keluaran, diuraikan lebih rinci pada poin keandalan integrasi AI berikut . Temuan *disagreement* pada dua kasus laptop dengan riwayat kerusakan fisik dan penggantian komponen menunjukkan perlunya peningkatan akurasi detail temporal pada histori aset, terutama terkait tanggal laporan kerusakan, servis, dan penggantian komponen. Perlu ditegaskan bahwa perbandingan sebelum dan sesudah implementasi (Tabel 4.17) bersifat analitis-konseptual berdasarkan karakteristik pendekatan, bukan hasil pengukuran kuantitatif terhadap *baseline* manual. Verifikasi kuantitatif manfaat operasional, seperti pengukuran waktu seleksi manual, belum dieksekusi dalam lingkup penelitian ini dan ditunda untuk penelitian lanjutan.

3. **Keandalan integrasi AI**

   Sistem menunjukkan keandalan awal pada ruang uji terbatas melalui validasi struktur output, pengujian integrasi, *limited-concurrency load test,* pengukuran konsistensi output, dan validasi respons *error* terkendali. Uji konsistensi AI final dilakukan pada 26 kasus dengan tiga eksekusi per kasus, sehingga menghasilkan 78 eksekusi, dan seluruhnya menghasilkan output valid. AI *Assign* menunjukkan stabilitas rekomendasi peringkat pertama penuh, AI *Health* menunjukkan konsistensi status tinggi, dan AI *Insight* menunjukkan variasi narasi yang perlu tetap diverifikasi operator karena keluaran berbasis teks tidak sepenuhnya deterministik. *Limited-concurrency load test* pada jalur *backend*/uji menunjukkan *error rate* 0% pada skenario 1, 5, dan 10 *concurrent requests*. Waktu respons endpoint pada pengujian tersebut dilaporkan pada Tabel 4.10 dan Tabel 4.11, namun hasil latensi tidak ditafsirkan sebagai latensi inferensi LLM *end-to-end.* Validasi *error* terkendali berhasil pada 4/4 skenario dasar. Selain itu, *fail injection* terhadap *endpoint* 9Router telah dieksekusi secara terbatas pada 45 uji (3 skenario kegagalan × 3 fitur × 5 iterasi), memverifikasi mekanisme*retry* dan *fallback* dua *layer* tanpa kegagalan fatal (Tabel 4.14 dan Tabel 4.15). Namun, pengujian pada beban produksi riil dengan variasi *trafic* dan *issues* berkepanjangan belum dilakukan.

   Secara umum, penelitian ini menunjukkan bahwa sistem dapat dipandang sebagai artefak awal berbasis *Design Science Research* yang valid pada ruang uji terbatas untuk

mendukung pengelolaan aset TI pascaakuisisi di BSI UII. Sistem tidak menggantikan keputusan manusia, melainkan menyediakan rekomendasi, *insight*, dan evaluasi *health* yang tetap memerlukan verifikasi operator.

5.1.1 Kontribusi Penelitian

Kontribusi penelitian dirangkum sebagai berikut.

Tabel 5.1 Kontribusi penelitian

Jenis kontribusi	Isi kontribusi
Praktis	Sistem menyediakan dukungan operasional pascaakuisisi aset TI melalui *dashboard* status, pengelolaan aset, histori *issue/maintenance*, rekomendasi alokasi aset, *insight* kondisi aset, serta evaluasi *health* berbasis konteks monitoring dan histori aset.
Teknis	Penelitian menunjukkan integrasi *backend*, basis data, AI *service*, 9Router, model LLM, monitoring-*alert*, validasi output, *retry* terbatas, *fallback*, dan *guardrail* aturan domain dalam satu artefak sistem.
Akademik	Penelitian menerapkan domain-*based expert agreement evaluation* melalui pasangan evaluator relevan pada evaluasi keluaran AI dalam konteks IT *Asset Management*. Selain itu, penelitian mengadopsi TOON (Schopplich, 2025) sebagai representasi konteks ringkas untuk *payload* domain ITAM, dengan rata-rata reduksi token 49,8% dibanding JSON mentah pada lima skenario pengukuran (masing-masing dengan tiga varian data).

Temuan penelitian memperkuat argumen bahwa integrasi LLM dan aturan domain pada sistem pendukung keputusan dapat menyeimbangkan adaptivitas interpretasi model dengan kontrol operasional. Penggunaan LLM berfungsi sebagai komponen interpretasi dan rekomendasi, bukan sebagai pengambil keputusan otomatis. Hal ini sejalan dengan prinsip *human-in-the-loop* pada LLM-DSS, yaitu manusia tetap menjadi pengambil keputusan akhir.

5.1.2 Keterbatasan Hasil

Hasil penelitian perlu dipahami dalam keterbatasan berikut.

1. **Studi kasus tunggal.**

   Penelitian dilakukan pada konteks BSI UII, sehingga hasil tidak langsung dapat digeneralisasi ke organisasi lain tanpa adaptasi struktur data, aturan bisnis, dan proses operasional.

2. **Jumlah kasus evaluasi terbatas.**

Evaluasi kualitas keluaran AI menggunakan 26 kasus final, dengan jumlah kasus per fitur berkisar 4 - 6. Oleh karena itu, *percent agreement* dibaca bersama distribusi skor dan analisis naratif, bukan sebagai generalisasi reliabilitas populasi.

3. **Evaluasi ahli bersifat kontekstual.**
   Pasangan evaluator dibentuk berdasarkan relevansi domain pada studi kasus BSI UII, sehingga hasil *agreement* menggambarkan kesesuaian operasional pada konteks penelitian dan belum mewakili semua kemungkinan organisasi atau profil evaluator.
4. ***Feedback* operator belum menjadi hasil utama.**
   Instrumen *feedback* operator berbasis Likert telah disiapkan, tetapi belum digunakan sebagai hasil utama karena pengambilan *feedback* memerlukan responden yang telah mencoba alur sistem secara langsung, sedangkan *frontend* masih dalam tahap penyelesaian akhir. Dengan demikian, evaluasi dukungan keputusan difokuskan pada skenario bisnis, penilaian rater domain, konsistensi output, dan pengujian teknis.
5. **Lingkungan pengujian terbatas.**
   Limited-concurrency load test dan validasi *error* dilakukan pada lingkungan pengembangan, bukan beban produksi riil. Latensi yang dilaporkan merupakan latensi jalur *backend*/uji, bukan latensi inferensi LLM *end-to-end.*
6. ***Fault* layanan eksternal dieksekusi secara terbatas.**
   Mitigasi yang sudah dilakukan antara lain diterapkan interval *request*, *retry* terbatas, dan *exponential backoff* pada *backend*, mekanisme *retry* dua *layer* (Langchain *service* dan Go *backend*) telah diverifikasi secara terbatas melalui *fail injection test* 500/429/504 pada 45 uji (Tabel 4.14 dan Tabel 4.15) tanpa kegagalan fatal.
7. **Data monitoring tidak lengkap**
   Data *disk* NDC tidak tersedia dan hanya diisi 0 sebagai *placeholder* kontrak JSON. Oleh karena itu, *disk* tidak digunakan sebagai dasar evaluasi maupun klaim kondisi aset.
8. **Pengukuran TOON bersifat kontekstual**
   Efisiensi token TOON dibanding JSON diukur pada sampel konteks penelitian ini dan tidak diklaim berlaku universal untuk seluruh format representasi atau seluruh penyedia LLM.

5.1.3 Pelajaran dari Siklus *Design Science Research*

Berdasarkan proses perancangan, evaluasi, dan penyempurnaan artefak, terdapat tiga pelajaran utama dari penerapan *Design Science Research* pada penelitian ini.

1. **Representasi konteks memengaruhi kualitas artefak LLM.**
   Format input bukan sekadar detail teknis, tetapi bagian dari desain artefak. Perubahan dari JSON mentah ke TOON menunjukkan bahwa pengurangan redundansi token dapat meningkatkan efisiensi representasi konteks tanpa mengubah data substantif yang dikirimkan ke model.
2. **Pendekatan *hybrid* lebih sesuai untuk keputusan operasional berisiko.**
   AI memberikan fleksibilitas dalam membaca konteks historis dan naratif, tetapi aturan domain tetap diperlukan untuk menjaga akuntabilitas, terutama pada evaluasi *health* aset *high-risk*. *Guardrail* menjadi mekanisme penting agar keluaran model tidak langsung diterjemahkan menjadi keputusan final.
3. **Evaluasi artefak perlu membaca aspek teknis dan operasional secara bersamaan.**
   Validitas struktur output, konsistensi eksekusi ulang, *agreement* evaluator domain, dan analisis *disagreement* saling melengkapi. Kombinasi bukti tersebut lebih sesuai untuk artefak DSS berbasis LLM dibanding hanya mengandalkan satu metrik teknis atau satu penilaian subjektif.

Pelajaran tersebut menunjukkan bahwa nilai utama penelitian tidak hanya terletak pada keberhasilan implementasi fitur, tetapi juga pada proses kalibrasi desain agar artefak lebih efisien, terkendali, dan relevan dengan konteks operasional BSI UII.

5.2 Saran

Berdasarkan hasil penelitian dan keterbatasan yang ditemukan, saran pengembangan disusun sebagai berikut. Prioritas tinggi menunjukkan kebutuhan yang perlu dipenuhi sebelum operasionalisasi sistem secara lebih serius, prioritas sedang menunjukkan penyempurnaan evaluasi, kinerja, dan tata kelola, sedangkan prioritas rendah menunjukkan arah pengembangan dan penelitian lanjutan.

Tabel 5.2 Saran

Prioritas	Kategori	Saran
Tinggi	Operasional	Susun SOP penggunaan output AI yang memuat kriteria penerimaan rekomendasi, alur verifikasi operator, dan mekanisme eskalasi untuk kasus berisiko tinggi. Output AI perlu diperlakukan sebagai masukan keputusan, bukan keputusan final otomatis.

Prioritas	Kategori	Saran
Tinggi	Data	Perkuat kualitas histori *issue* dan *maintenance*, khususnya konsistensi tanggal laporan, tanggal servis, tanggal penggantian komponen, dan tanggal pembaruan data. Temuan *disagreement* pada dua kasus laptop menunjukkan bahwa ketidakjelasan temporal dapat memengaruhi penilaian keakuratan klaim AI.
Tinggi	AI dan *prompt*	Tambahkan *guardrail* agar model tidak menyimpulkan tanggal tindakan teknis jika *timestamp* pada histori tidak eksplisit. *Prompt* juga perlu menegaskan perbedaan antara tanggal *issue*, tanggal servis, tanggal penggantian komponen, dan tanggal *update* data.
Tinggi	Keandalan teknis	Tambahkan *circuit breaker* dan *degraded mode* terukur agar sistem tetap terkendali saat layanan model bermasalah berkepanjangan. Mekanisme *retry* dan *fallback* dua *layer* telah diverifikasi terbatas (Tabel 4.15), namun belum mencakup gangguan berkepanjangan dan beban konkurensi tinggi.
Sedang	Monitoring	Lanjutkan pemetaan metrik Grafana/Prometheus per jenis aset dan tingkatkan konsistensi label asset_uuid pada *alert.* Jika memungkinkan, lengkapi data *disk* agar evaluasi *health* dapat mencakup sinyal penyimpanan secara valid.
Sedang	Performa	Lakukan pengukuran latensi *end-to-end* yang benar-benar mencakup *backend*, AI *service*, proxy LLM, dan inferensi model. Hasil tersebut perlu dibedakan dari *limited-concurrency load test backend*/uji yang telah dilaporkan pada BAB IV.
Sedang	Evaluasi pengguna	Lakukan *feedback* operator berbasis Likert setelah *frontend* siap digunakan atau didemonstrasikan secara langsung. Minimal melibatkan AMP sebagai evaluator lintas fitur, lalu diperluas ke NDC dan Inventaris agar persepsi penggunaan sistem dapat dilaporkan secara lebih lengkap.
Sedang	Tata kelola	Pertahankan prinsip *human-in-the-loop*, audit log, dan pencatatan alasan rekomendasi. Setiap perubahan status *health* perlu tetap dapat ditelusuri ke data input, output AI, aturan *guardrail*, dan keputusan operator.
Rendah	Penelitian lanjutan	Perluas jumlah kasus per fitur, lakukan replikasi pada institusi lain, dan bandingkan konfigurasi model LLM yang berbeda untuk melihat stabilitas keluaran, biaya, dan kualitas rekomendasi.

Dengan saran tersebut, pengembangan berikutnya dapat diarahkan pada peningkatan kesiapan operasional, penguatan validitas eksternal, dan penyempurnaan kualitas data agar sistem pendukung keputusan manajemen aset TI berbasis LLM dapat digunakan secara lebih stabil dan akuntabel. Secara keseluruhan, penelitian ini menempatkan artefak yang dikembangkan sebagai kontribusi awal yang menghubungkan IT *Asset Management*, LLM, *guardrail* aturan domain, dan evaluasi berbasis domain dalam satu rancangan DSR yang dapat dikembangkan lebih lanjut.

DAFTAR PUSTAKA

Albalghouni, A., Alkatheeri, F., Elbaioumy, N., Albaiti, W., Shamsuzzaman, M., & Hamdan, S. (2018). *Challenges of implementing ISO 55000: A case study from an educational institute*.

Ayuningtyas, P. K., Atmodjo Wp, D., & Rachmadi, P. (2023). Performance And Functional Testing With The Black Box Testing Method. *International Journal of Progressive Sciences and Technologies*, *39*(2), 212. https://doi.org/10.52155/ijpsat.v39.2.5471

Dong, Y., Mu, R., Jin, G., Qi, Y., Hu, J., Zhao, X., Meng, J., Ruan, W., & Huang, X. (2024). *Building Guardrails for Large Language Models* (arXiv:2402.01822). arXiv. https://doi.org/10.48550/arXiv.2402.01822

Durachman, Y., Eva Khudzaeva, & Aulia, N. (2026). Large Language Models for Intelligent Decision Support in Inventory and Supply Chain Operations: A Systematic Literature Review. *Journal of Information Systems and Informatics*, *8*(2), 1984–2016. https://doi.org/10.63158/journalisi.v8i2.1540

Gavrikova, E., Volkova, I., & Burda, Y. (2020). Strategic Aspects of Asset Management: An Overview of Current Research. *Sustainability*, *12*(15), 5955. https://doi.org/10.3390/su12155955

Hastjarjo, T. D. (2011). *VALIDITAS EKSPERIMEN*. https://doi.org/10.22146/bpsi.11558

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2025). A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, *43*(2), 1–55. https://doi.org/10.1145/3703155

Jalal, A. A., Jasim, A. A., & Mahawish, A. A. (2022). A web content mining application for detecting relevant pages using Jaccard similarity. *International Journal of Electrical and Computer Engineering (IJECE)*, *12*(6), 6461. https://doi.org/10.11591/ijece.v12i6.pp6461-6471

Joshi, A., Kale, S., Chandel, S., & Pal, D. (2015). Likert Scale: Explored and Explained. *British Journal of Applied Science & Technology*, *7*(4), 396–403. https://doi.org/10.9734/BJAST/2015/14975

Kumar, M. A., Bollavathini, S. R., & Sisir, A. T. (2024). A ROADMAP TO SUCCESS: STRATEGIES AND CHALLENGES INADOPTING AIOPS FOR IT OPERATIONS.

International Journal of Interpreting Enigma Engineers, *01*(02), 28–35. https://doi.org/10.62674/ijiee.2024.v1i02.005

Lazaros, K., Vrahatis, A. G., & Kotsiantis, S. (2026). Human-in-the-Loop Artificial Intelligence: A Systematic Review of Concepts, Methods, and Applications. *Entropy*, *28*(4), 377. https://doi.org/10.3390/e28040377

Lin, Z., Trivedi, S., & Sun, J. (2024). *Generating with Confidence: Uncertainty Quantification for Black-box Large Language Models*. https://doi.org/10.48550/arXiv.2305.19187

Maletič, D., Maletič, M., Al-Najjar, B., & Gomišček, B. (2020). An Analysis of Physical Asset Management Core Practices and Their Influence on Operational Performance. *Sustainability*, *12*(21), 9097. https://doi.org/10.3390/su12219097

Mei, L., Yao, J., Ge, Y., Wang, Y., Bi, B., Cai, Y., Liu, J., Li, M., Li, Z.-Z., Zhang, D., Zhou, C., Mao, J., Xia, T., Guo, J., & Liu, S. (2025). *A Survey of Context Engineering for Large Language Models* (arXiv:2507.13334). arXiv. https://doi.org/10.48550/arXiv.2507.13334

Notaro, P., Cardoso, J., & Gerndt, M. (2021). A Survey of AIOps Methods for Failure Management. *ACM Transactions on Intelligent Systems and Technology*, *12*(6), 1–45. https://doi.org/10.1145/3483424

Nyimbili, F., & Nyimbili, L. (2024). Types of Purposive Sampling Techniques with Their Examples and Application in Qualitative Research Studies. *British Journal of Multidisciplinary and Advanced Studies*, *5*(1), 90–99. https://doi.org/10.37745/bjmas.2022.0419

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Punithavathy, E., & Priya, N. (2024). Auto retry circuit breaker for enhanced performance in microservice applications. *International Journal of Electrical and Computer Engineering (IJECE)*, *14*(2), 2274. https://doi.org/10.11591/ijece.v14i2.pp2274-2281

Rahadi, R. A., Indrayana, G. G., Afgani, K. F., Darmansyah, A., Anggoro, Y., Halim, R., Fitrianda, S., Purbayati, R., Astari, A. W., & Ayudiatri, S. (2024). Review of Asset Management Practice in Indonesian State-Owned Enterprise. *Journal Integration of Management Studies*, *2*(1), 61–70. https://doi.org/10.58229/jims.v2i1.138

Rashidi, M., Ghodrat, M., Samali, B., & Mohammadi, M. (2018). Decision Support Systems. In M. Pomffyova (Ed.), *Management of Information Systems*. InTech. https://doi.org/10.5772/intechopen.79390

Reddy, B., & Fields, R. (2022). From past to present: A comprehensive technical review of rule-based expert systems from 1980 -- 2021. *Proceedings of the ACM Southeast Conference*, 167–172. https://doi.org/10.1145/3476883.3520211

Salamun, S., Saka Wikarta, J., & Puspitasari, I. (2022). Penerapan Standar ISO 55000 Untuk Manajemen Aset Dengan Pendekatan Metode Topsis. *Jurnal Komputer Terapan*, *8*(2), 346–358. https://doi.org/10.35143/jkt.v8i2.5730

Schmidt, C. W., Reddy, V., Zhang, H., Alameddine, A., Uzan, O., Pinter, Y., & Tanner, C. (2024). *Tokenization Is More Than Compression* (arXiv:2402.18376). arXiv. https://doi.org/10.48550/arXiv.2402.18376

Schopplich, J. (2025). *TOON — Token-Oriented Object Notation*. Toonformat.Dev. https://toonformat.dev/

Setiawan, R., Kurniadi, D., Aulawi, H., & Kurniawati, R. (2019). Asset management information system for higher education. *Journal of Physics: Conference Series*, *1402*(2), 022083. https://doi.org/10.1088/1742-6596/1402/2/022083

Stone, M., Irrechukwu, C., Perper, H., Wynne, D., & Kauffman, L. (2018). *IT asset management: Financial services* (NIST SP 1800-5; p. NIST SP 1800-5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.1800-5

Tan, H. (2017). A brief history and technical review of the expert system research. *IOP Conference Series: Materials Science and Engineering*, *242*, 012111. https://doi.org/10.1088/1757-899X/242/1/012111

Teixeira, H. N., Lopes, I., & Braga, A. C. (2020). Condition-based maintenance implementation: A literature review. *Procedia Manufacturing*, *51*, 228–235. https://doi.org/10.1016/j.promfg.2020.10.033

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT* (arXiv:2302.11382). arXiv. https://doi.org/10.48550/arXiv.2302.11382

Yusof, I. J., & Abdul Rahim, F. E. (2024). Kappa Fleiss Analysis: Evidence Of Content Validity for Formative Assessment Literacy Test for Teachers of Fundamental Subjects in Vocational Special School/ Analisis Fleiss Kappa: Evidens Kesahan Kandungan Ujian Literasi Pentaksiran Formatif Guru-Guru Subjek Asas Sekolah Menengah Pendidikan Khas Vokasional. *Sains Humanika*, *16*(2), 9–16. https://doi.org/10.11113/sh.v16n2.2009

Zhaoxue, J., Tong, L., Zhenguo, Z., Jingguo, G., Junling, Y., & Liangxiong, L. (2021a). A Survey On Log Research Of AIOps: Methods and Trends. *Mobile Networks and Applications*, *26*(6), 2353–2364. https://doi.org/10.1007/s11036-021-01832-3

Zhaoxue, J., Tong, L., Zhenguo, Z., Jingguo, G., Junling, Y., & Liangxiong, L. (2021b). A Survey On Log Research Of AIOps: Methods and Trends. *Mobile Networks and Applications*, *26*(6), 2353–2364. https://doi.org/10.1007/s11036-021-01832-3

Zhou, N., Patel, D., Lin, S., & O'Donncha, F. (2024). *Towards Automated Solution Recipe Generation for Industrial Asset Management with LLM* (arXiv:2407.18992). arXiv. https://doi.org/10.48550/arXiv.2407.18992

LAMPIRAN

Lampiran 1. Daftar Kasus, Rubrik *Rater*, dan Data Mentah Skor *Rater*

1.1 Daftar Kasus Evaluasi

Tabel 6.1 Data Kasus

Kode	Fitur	Risiko	Aset	Ringkasan Konteks	Evaluator
HLT-A1	AI *Health*	*High-risk*	Juniper MX204 (router)	*Router monitorable* NDC; data CPU/RAM tersedia	AMP, NDC
HLT-A2	AI *Health*	*High-risk*	Fortinet 1001F (firewall)	*Firewall monitorable*; *expected* kondisi stabil	AMP, NDC
HLT-A3	AI *Health*	*High-risk*	Fortinet 601F (router)	*Router* dengan *synthetic network anomaly* + *alert* aktif	AMP, NDC
HLT-A4	AI *Health*	*High-risk*	Supermicro SYS-120U-TNR (server)	Server dengan *synthetic* CPU *spike/sustained critical alert*	AMP, NDC
HLT-B1	AI *Health*	*Low-risk*	MacBook Pro 13.3 MPXT2ID	*Used-Critical*; riwayat baterai drop, bau gosong, IC audio rusak	AMP, Inventaris
HLT-B2	AI *Health*	*Low-risk*	MacBook Pro 13.3 MPXT2ID	*Used-Warning;* gejala lemot, panas, ganti baterai	AMP, Inventaris
HLT-B3	AI *Health*	*Low-risk*	MacBook Pro 13.3 MPXT2ID	*Used-Healthy*; kondisi baik, bekas	AMP, Inventaris
HLT-B4	AI *Health*	*Low-risk*	MacBook Air M4 Sky Blue	*New-Healthy*; aset baru	AMP, Inventaris
HLT-B5	AI *Health*	*Low-risk*	Lenovo IdeaPad SLIM 5-14ABR8	*Used-Healthy*; Lenovo 2024	AMP, Inventaris
HLT-B6	AI *Health*	*Low-risk*	MacBook Air M2 13.6	*New-Healthy*; aset baru	AMP, Inventaris
INS-A1	AI *Insight*	*High-risk*	Juniper MX204 (router)	*Router monitorable* NDC	AMP, NDC
INS-A2	AI Insight	*High-risk*	Fortinet 1001F (firewall)	*Firewall monitorable*	AMP, NDC
INS-A3	AI *Insight*	*High-risk*	Fortinet 601F (router)	*Router* dengan *synthetic network anomaly* + *alert* aktif	AMP, NDC
INS-A4	AI *Insight*	*High-risk*	Supermicro SYS-120U-TNR (server)	Server dengan *synthetic* CPU *spike/sustained alert*	AMP, NDC

Kode	Fitur	Risiko	Aset	Ringkasan Konteks	Evaluator
INS-B1	AI *Insight*	*Low-risk*	MacBook Pro 13.3 MPXT2ID	*Used-Critical;* baterai drop, bau gosong	AMP, Inventaris
INS-B2	AI *Insight*	*Low-risk*	MacBook Pro 13.3 MPXT2ID	*Used-Warning*; lemot, panas	AMP, Inventaris
INS-B3	AI *Insight*	*Low-risk*	MacBook Pro 13.3 MPXT2ID	*Used-Healthy;* kondisi baik	AMP, Inventaris
INS-B4	AI *Insight*	*Low-risk*	MacBook Air M4 Sky Blue	*New-Healthy*; aset baru	AMP, Inventaris
INS-B5	AI *Insight*	*Low-risk*	Lenovo IdeaPad SLIM 5-14ABR8	*Used-Healthy*; Lenovo 2024	AMP, Inventaris
INS-B6	AI *Insight*	*Low-risk*	MacBook Air M2 13.6	*New-Healthy*; aset baru	AMP, Inventaris
ASN-B1	AI *Insight*	*Low-risk*	Pool 6 laptop	Butuh stabil untuk koordinasi + pekerjaan umum	AMP, Inventaris
ASN-B2	AI *Insight*	*Low-risk*	Pool 6 laptop	Butuh aset aman dipinjamkan; hindari aset rusak berat	AMP, Inventaris
ASN-B3	AI *Insight*	*Low-risk*	Pool 6 laptop	Butuh stabil untuk *development* junior	AMP, Inventaris
ASN-B4	AI *Insight*	*Low-risk*	Pool 6 laptop	Butuh performa tinggi; hindari laptop lemot/panas	AMP, Inventaris
ASN-B5	AI *Insight*	*Low-risk*	Pool 6 laptop	Butuh stabil untuk layanan pengguna/operasional *support*	AMP, Inventaris
ASN-B6	AI *Insight*	*Low-risk*	Pool 6 laptop	Butuh performa tinggi untuk analisis keamanan	AMP, Inventaris

1.2 Rubrik Penilaian *Rater*

Tabel 6.2 Rubik Penilaian

Skor	Kategori	Definisi Operasional
2	Sesuai	Output sepenuhnya sesuai dengan aspek yang dinilai; tidak ada kesalahan faktual
1	Sebagian sesuai	Output mengandung informasi yang benar namun kurang lengkap atau kurang tepat pada detail minor
0	Tidak sesuai	Output mengandung kesalahan faktual, klaim tidak berdasar, atau tidak relevan

1.3 Raw Data Score Rater

A. AMP

Tabel 6.3 Hasil *Rater* AMP

Kode	Fitur	Pertanyaan	Jawaban	Alasan
HLT-A1	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-A1	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-A1	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-A2	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-A2	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-A2	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-A3	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-A3	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-A3	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-A4	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-A4	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-A4	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-B1	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B1	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-B1	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-B2	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B2	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	

Kode	Fitur	Pertanyaan	Jawaban	Alasan
HLT-B2	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-B3	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B3	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-B3	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-B4	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B4	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-B4	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-B5	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B5	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-B5	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-B6	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B6	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-B6	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
INS-A1	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-A1	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-A1	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-A2	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-A2	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-A2	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	

Kode	Fitur	Pertanyaan	Jawaban	Alasan
INS-A3	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-A3	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-A3	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-A4	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-A4	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-A4	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-B1	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-B1	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B1	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-B2	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-B2	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B2	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-B3	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-B3	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B3	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-B4	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-B4	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B4	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-B5	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	

Kode	Fitur	Pertanyaan	Jawaban	Alasan
INS-B5	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B5	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-B6	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-B6	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B6	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
ASN-B1	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B1	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	
ASN-B1	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	
ASN-B2	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B2	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	
ASN-B2	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	
ASN-B3	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B3	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	
ASN-B3	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	
ASN-B4	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B4	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	
ASN-B4	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	
ASN-B5	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B5	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	

Kode	Fitur	Pertanyaan	Jawaban	Alasan
ASN-B5	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	
ASN-B6	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B6	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	
ASN-B6	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	

B. NDC

Tabel 6.4 Hasil *Rater* NDC

Kode	Fitur	Pertanyaan	Jawaban	Alasan
HLT-A1	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-A1	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-A1	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-A2	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-A2	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-A2	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-A3	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-A3	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-A3	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-A4	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-A4	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-A4	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	

Kode	Fitur	Pertanyaan	Jawaban	Alasan
INS-A1	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-A1	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-A1	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-A2	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-A2	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-A2	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-A3	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-A3	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-A3	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-A4	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-A4	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-A4	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	

C. Inventaris

Tabel 6.5 Hasil Rater Inventaris

Kode	Fitur	Pertanyaan	Jawaban	Alasan
HLT-B1	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B1	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-B1	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	**1**	Tanggal kerusakan & perbaikan tidak akurat — kerusakan terjadi 2023, perbaikan terakhir 2024

Kode	Fitur	Pertanyaan	Jawaban	Alasan
HLT-B2	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B2	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-B2	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	**1**	Tanggal penggantian baterai tidak akurat servis Oktober 2024
HLT-B3	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B3	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-B3	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-B4	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B4	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-B4	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-B5	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B5	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-B5	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	
HLT-B6	*Health*	Apakah status *HEALTHY/WARNING/CRITICAL* sesuai dengan data monitoring dan histori?	2	
HLT-B6	*Health*	Apakah faktor utama yang ditampilkan benar-benar faktor paling signifikan?	2	
HLT-B6	*Health*	Apakah klaim temporal (tanggal kejadian, urutan servis) dapat diverifikasi terhadap data input?	2	

Kode	Fitur	Pertanyaan	Jawaban	Alasan
INS-B1	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-B1	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B1	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-B2	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-B2	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B2	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-B3	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-B3	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B3	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-B4	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-B4	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B4	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
INS-B5	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-B5	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B5	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	

Kode	Fitur	Pertanyaan	Jawaban	Alasan
INS-B6	*Insight*	Apakah deskripsi kondisi aset sesuai dengan data monitoring/histori yang tersedia?	2	
INS-B6	*Insight*	Apakah saran/rekomendasi yang diberikan sesuai dengan temuan kondisi?	2	
INS-B6	*Insight*	Apakah output menyimpulkan di luar data yang tersedia?	2	
ASN-B1	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B1	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	
ASN-B1	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	
ASN-B2	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B2	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	
ASN-B2	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	
ASN-B3	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B3	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	
ASN-B3	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	
ASN-B4	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B4	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	
ASN-B4	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	

Kode	Fitur	Pertanyaan	Jawaban	Alasan
ASN-B5	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B5	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	
ASN-B5	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	
ASN-B6	*Assign*	Apakah aset yang direkomendasikan sesuai dengan kebutuhan pengguna dalam skenario?	2	
ASN-B6	*Assign*	Apakah alasan pemilihan merujuk pada data aset aktual (spesifikasi, kondisi, riwayat)?	2	
ASN-B6	*Assign*	Apakah ada klaim yang bertentangan dengan data input?	2	

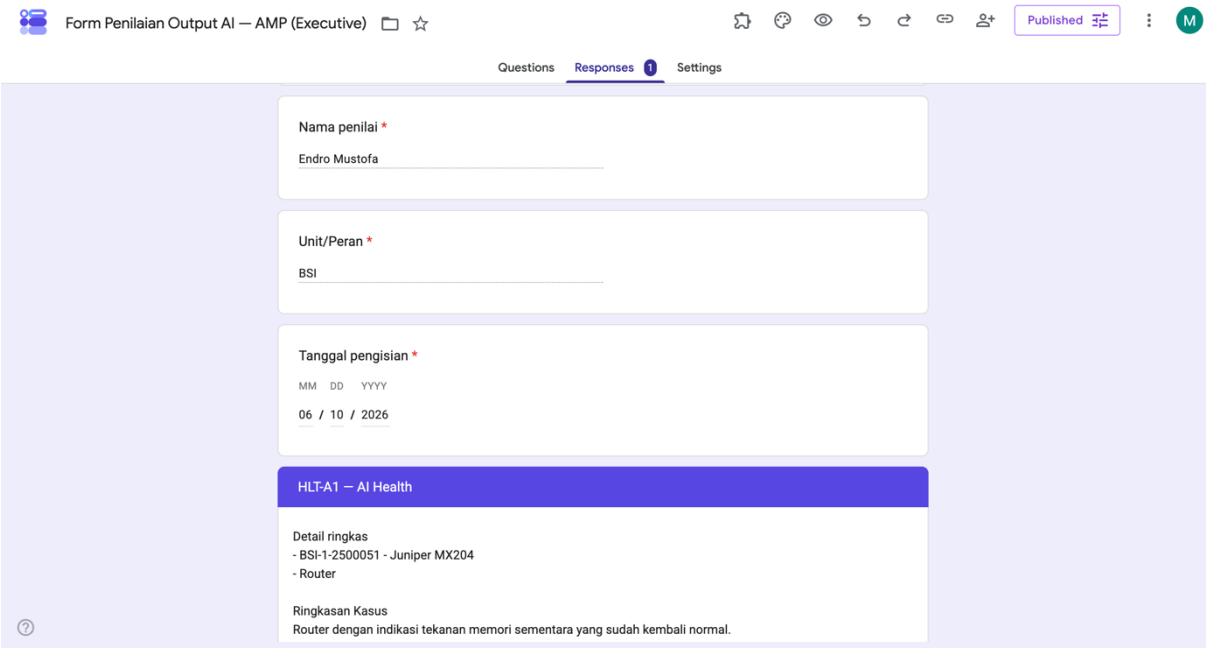
1.4 *Percent Agreement*

Tabel 6.6 Ringkasan *Percent Agreement*

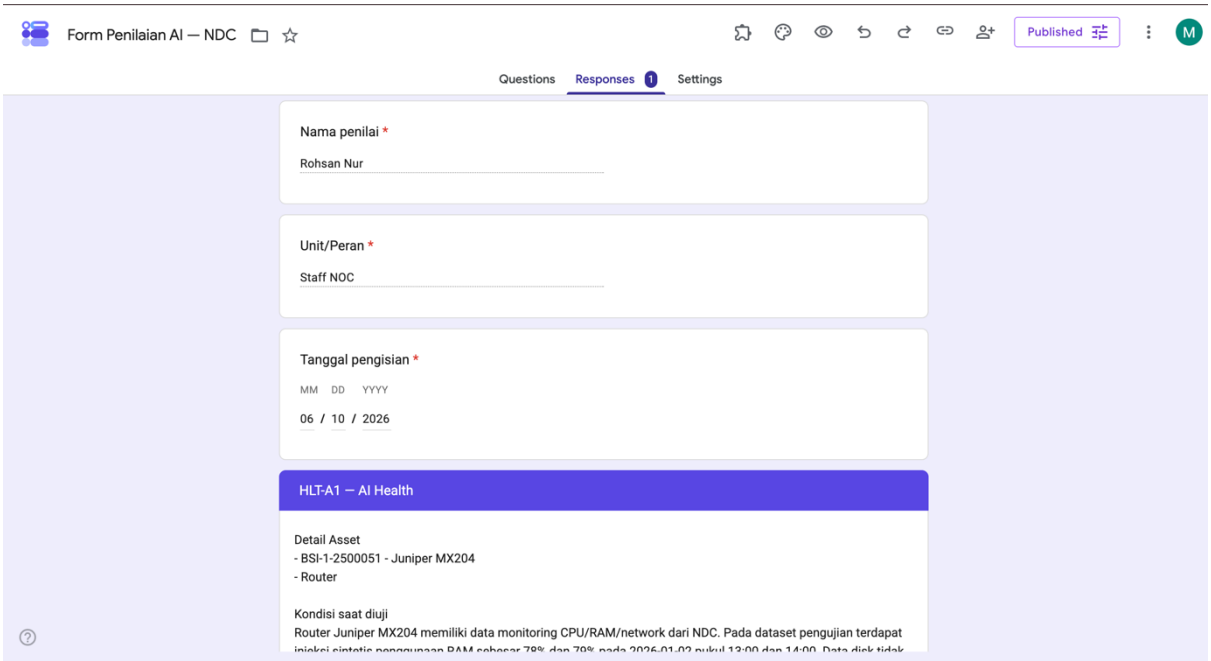
Pasangan	Domain	Fitur	Aspek	n Kasus	*Agreement*	*Disagreement*
AMP vs NDC	Monitoring	*Health*	Sesuai kondisi	4	1,000	
AMP vs NDC	Monitoring	*Health*	Alasan relevan	4	1,000	
AMP vs NDC	Monitoring	*Health*	Keakuratan klaim	4	1,000	
AMP vs NDC	Monitoring	*Insight*	Sesuai kondisi	4	1,000	
AMP vs NDC	Monitoring	*Insight*	Alasan relevan	4	1,000	
AMP vs NDC	Monitoring	*Insight*	Keakuratan klaim	4	1,000	
AMP vs Inventaris	Non-monitoring	*Health*	Sesuai kondisi	6	1,000	
AMP vs Inventaris	Non-monitoring	*Health*	Alasan relevan	6	1,000	

Pasangan	Domain	Fitur	Aspek	n Kasus	*Agreement*	*Disagreement*
AMP vs Inventaris	Non-monitoring	*Health*	Keakuratan klaim	6	**0,667**	HLT-B1, HLT-B2
AMP vs Inventaris	Non-monitoring	*Insight*	Sesuai kondisi	6	1,000	
AMP vs Inventaris	Non-monitoring	*Insight*	Alasan relevan	6	1,000	
AMP vs Inventaris	Non-monitoring	*Insight*	Keakuratan klaim	6	1,000	
AMP vs Inventaris	Non-monitoring	*Assign*	Sesuai kondisi	6	1,000	
AMP vs Inventaris	Non-monitoring	*Assign*	Alasan relevan	6	1,000	
AMP vs Inventaris	Non-monitoring	*Assign*	Keakuratan klaim	6	1,000	

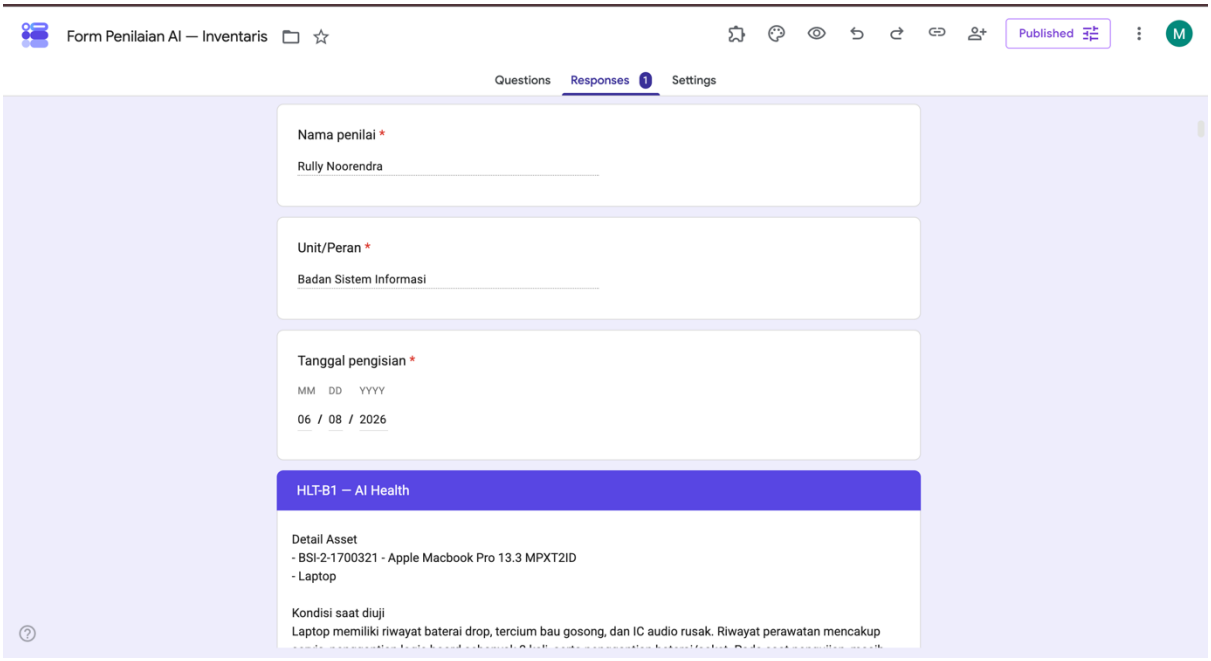
1.5 Dokumentasi *Form*



Gambar 6.1 *Form* Penilaian Output LLM AMP



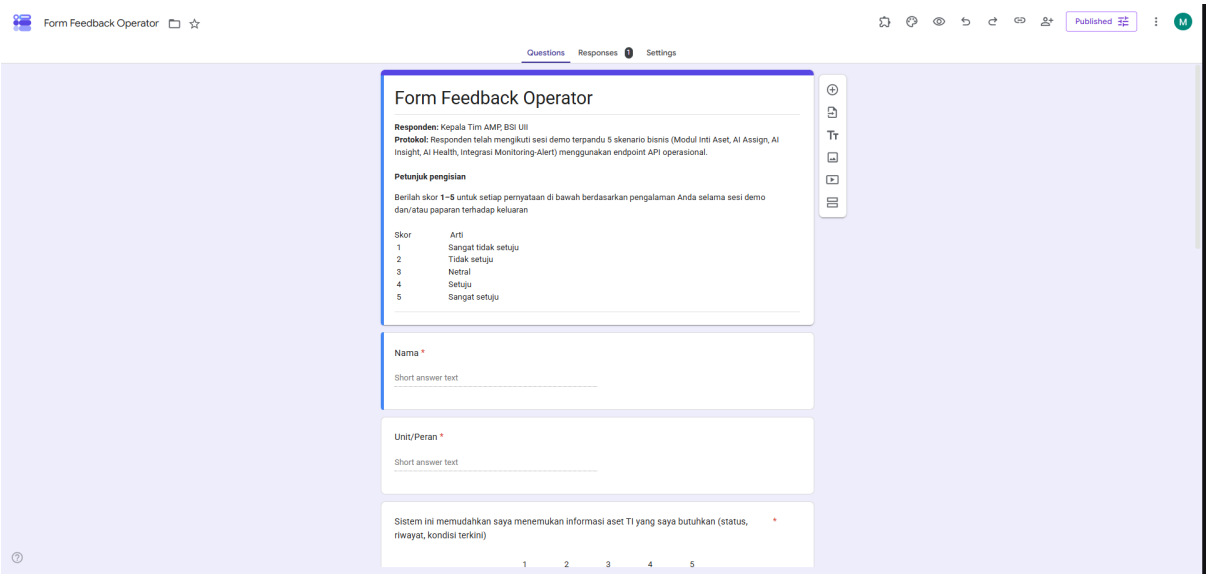
Gambar 6.2 *Form* Penilaian Output LLM AMP



Gambar 6.3 *Form* Penilaian Output LLM AMP

Lampiran 2. Instrumen dan Hasil Evaluasi Likert

2.1 *Instrument* Likert



Gambar 6.4 *Form Rater Feedback* Likert

Tabel 6.7 Pertanyaan Likert

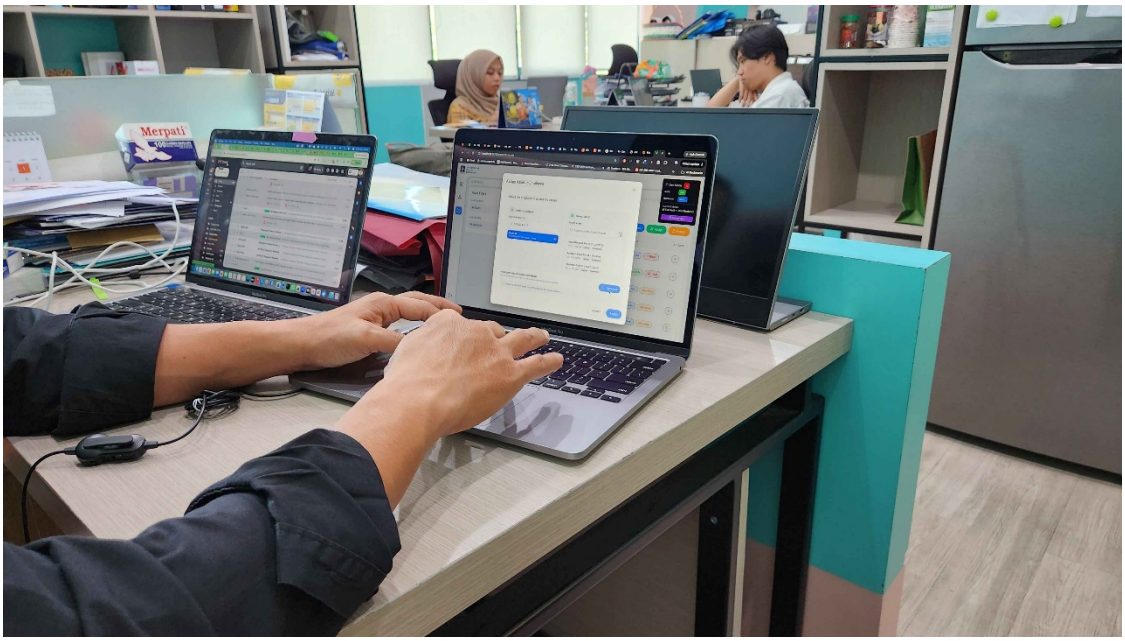
No	Aspek	Pernyataan
1	Efisiensi pencarian informasi	Sistem ini memudahkan saya menemukan informasi aset TI yang saya butuhkan (status, riwayat, kondisi terkini)
2	Dukungan AI *Assign*	Fitur rekomendasi alokasi membantu saya menentukan perangkat TI mana yang paling tepat untuk diberikan kepada pengguna
3	Keterpahaman AI *Insight*	Narasi kondisi aset yang disajikan sistem membantu saya memahami keadaan aset dengan lebih cepat dibanding membaca data mentah
4	Dukungan AI *Health*	Status kesehatan aset (*HEALTHY/WARNING/CRITICAL*) membantu saya memprioritaskan aset mana yang perlu ditangani lebih dulu
5	Verifiabilitas alasan	Alasan yang diberikan sistem cukup jelas dan dapat saya periksa kebenarannya berdasarkan data aset
6	Keputusan tetap di operator	Saya tetap merasa memegang kendali penuh atas keputusan akhir — sistem hanya membantu, bukan menggantikan
7	Relevansi operasional	Secara keseluruhan, sistem ini relevan dan sesuai dengan kebutuhan pekerjaan saya dalam mengelola aset TI

2.2 Hasil Liker

Tabel 6.8 Hasil Likert (n=1 AMP)

No	Aspek	Skor
1	Efisiensi pencarian informasi aset	5
2	Dukungan rekomendasi alokasi (AI *Assign*)	4
3	Keterpahaman narasi *insight* (AI *Insight*)	4
4	Dukungan prioritas kesehatan (AI *Health*)	4
5	Verifiabilitas alasan	5
6	Keputusan tetap di tangan operator	5
7	Relevansi dengan pekerjaan sehari-hari	4
Rata-rata		**4,4 / 5**

2.3 Dokumentasi Sesi





Lampiran 3. Surat Izin Penelitian dan Lembar Persetujuan

UNIVERSITAS ISLAM INDONESIA

FAKULTAS TEKNOLOGI INDUSTRI
Gedung KH. Mas Mansur
Kampus Terpadu Universitas Islam Indonesia
Jl. Kaliurang km 14,5 Yogyakarta 55584
T. (0274) 898444 ext. 4100, 4101
F. (0274) 895007
E. fti@uii.ac.id
W. fti.uii.ac.id

Nomor : 35/Kaprodi.IF.S1/20/IF/II/2026
Lamp : -
Hal : Permohonan Penelitian Tugas Akhir

Kepada Yth.
Kepala
Badan Sistem Informasi Universitas Islam Indonesia
di Gedung GBPH Prabuningrat (Lantai 4) Kampus Terpadu UII, Jl. Kaliurang km. 14,5, Sleman, Yogyakarta

Assalamu'alaikum Warahmatullahi Wabarokatuh

Melalui surat ini Program Studi Informatika – Program Sarjana Universitas Islam Indonesia memohon kepada Bapak/Ibu agar dapat mengizinkan mahasiswa kami untuk melaksanakan penelitian Tugas Akhir (TA) di institusi yang Bapak/Ibu pimpin. Pelaksanaan penelitian TA ini merupakan salah satu syarat lulus di program studi kami.

Mahasiswa yang akan melaksanakan penelitian adalah :

Nama : Muhammad Ilham Mahfuzh

No. Mhs : 20523139

Kami berharap mahasiswa kami dapat melaksanakan penelitian di institusi Bapak/Ibu antara bulan Februari 2026 sampai dengan Juni 2026. Namun demikian, kepastian jadwal pelaksanaan kami serahkan sepenuhnya kepada institusi Bapak/Ibu.

Demikian permohonan kami, atas perhatian serta terkabulnya permohonan ini kami ucapkan terima kasih.

Wassalamu'alaykum Warahmatullahi Wabarokatuh

Yogyakarta, 24 Februari 2026
Ketua Program Studi Informatika - Program Sarjana, Fakultas Teknologi Industri

YOGYAKARTA

Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.