

**Perancangan dan Implementasi *Backend* Sistem Informasi
Bersama (SIBER) di PT PLN Unit Pelayanan dan Pelaksanaan
(UP3) Yogyakarta**



Disusun Oleh:

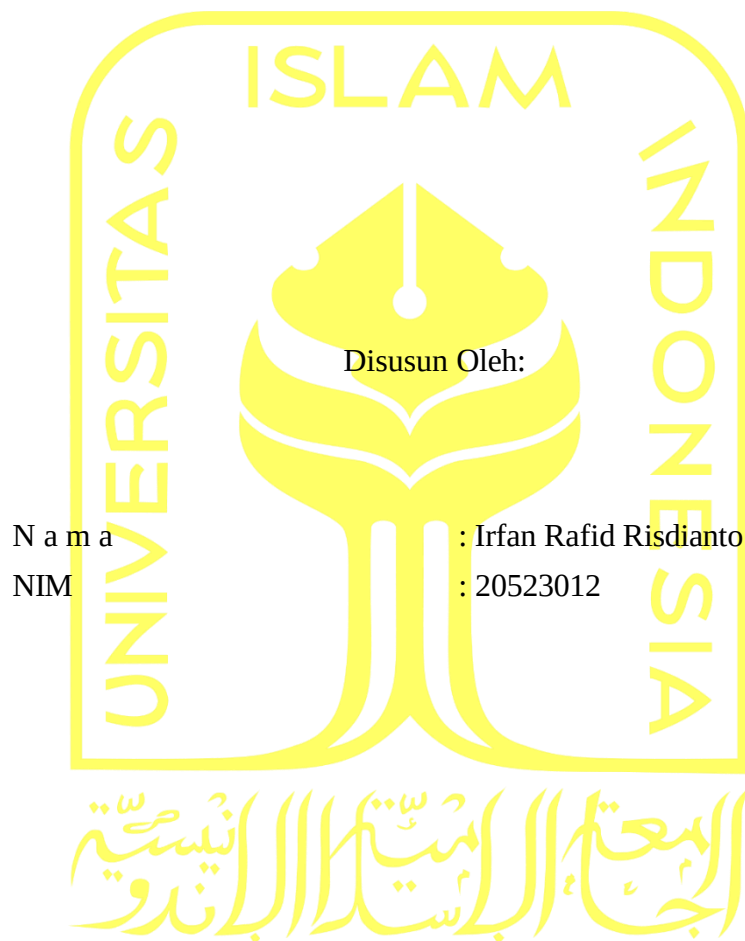
N a m a : Irfan Rafid Risdianto
NIM : 20523012

**PROGRAM STUDI INFORMATIKA-PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2025**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**Perancangan dan Implementasi *Backend* Untuk Sistem Bersama
(SIBER) di PT PLN Unit Pelayanan dan Pelaksanaan (UP3)
Yogyakarta**

TUGAS AKHIR



Yogyakarta, 22 Januari 2024 Pembimbing,

(Dr. Ahmad Luthfi, S.Kom, M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

Perancangan dan Implementasi *Backend* Untuk Sistem Informasi Bersama (SIBER) di PT PLN Unit Pelayanan dan Pelaksanaan (UP3) Yogyakarta

TUGAS AKHIR

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

Yogyakarta, 22 Januari 2025

Tim Penguji

Dr. Ahmad Luthfi, S.Kom., M.Kom.

Anggota 1

Moh. Idris, S.Kom., M.Kom.

Anggota 2

Dr. Feri Wijayanto, S.T., M.T.

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 : Irfan Rafid Risdianto

NIM : 20523012

Tugas akhir dengan judul:

**Perancangan dan Implementasi *Backend* Sistem Informasi
Bersama (SIBER) di PT PLN Unit Pelayanan dan Pelaksanaan
(UP3)
Yogyakarta**

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, 22 Januari 2025



(Irfan Rafid Risdianto)

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya, skripsi ini saya persembahkan kepada kedua orang tua tercinta, yang selalu memberikan dukungan, doa, dan kasih sayang tanpa batas. Terima kasih atas segala pengorbanan dan motivasi yang tiada henti. Juga kepada para dosen dan pembimbing akademik, yang telah membimbing dan memberikan ilmu serta arahan selama proses penyusunan skripsi ini. Tanpa bimbingan dan pengetahuan yang diberikan, skripsi ini tidak akan terselesaikan. Tidak lupa kepada sahabat dan teman-teman seperjuangan, yang selalu memberikan semangat, berbagi tawa, dan menjadi tempat berbagi suka maupun duka selama masa kuliah. Terima kasih atas dukungan dan kebersamaannya. Skripsi ini juga saya persembahkan kepada almamater tercinta, yang telah memberikan wadah dan kesempatan untuk belajar, berkembang, dan mengembangkan diri. Akhir kata, persembahan ini juga untuk diri sendiri, yang telah berusaha keras, bersabar, dan tidak pernah menyerah dalam menghadapi segala rintangan dan tantangan selama proses penyusunan skripsi ini. Semoga skripsi ini dapat memberikan manfaat dan kontribusi bagi ilmu pengetahuan serta menjadi inspirasi bagi generasi selanjutnya.

HALAMAN MOTO

"Kesuksesan bukanlah kunci kebahagiaan. Kebahagiaan adalah kunci kesuksesan. Jika Anda mencintai apa yang Anda lakukan, Anda akan sukses." – Albert Schweitzer

"Dan orang-orang yang berjuang untuk (mencari keridaan) Kami, benar-benar akan Kami tunjukkan kepada mereka jalan-jalan Kami."

– (QS. *Al-Ankabut*: 69)

"Apabila kamu telah bertekad, maka bertawakallah kepada Allah."

– (QS. *Ali Imran*: 159)

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Dengan segala kerendahan hati, penulis memanjatkan puji dan syukur ke hadirat Allah SWT atas rahmat dan hidayah-Nya yang senantiasa menyertai. Shalawat dan salam tak lupa penulis haturkan kepada Nabi Muhammad SAW, yang telah menjadi teladan umat manusia. Berkat pertolongan-Nya, laporan magang ini dapat diselesaikan dengan baik sebagai bagian dari persyaratan jalur magang di Fakultas Teknologi Industri, Program Studi Informatika, Universitas Islam Indonesia. Laporan ini disusun sebagai dokumentasi dari pelaksanaan program magang di PT PLN UP3 Yogyakarta sekaligus sebagai bentuk apresiasi kepada berbagai pihak yang telah membantu dan mendukung. Oleh karena itu, penulis dengan tulus menyampaikan terima kasih kepada:

1. Allah Subhanahu Wata'ala yang telah memberikan kesehatan, kekuatan, dan kelancaran dalam menyelesaikan kegiatan magang ini.
2. Kedua orang tua yang senantiasa memberikan doa, motivasi, serta dukungan baik secara moral maupun material.
3. Bapak Prof. Dr. Ir. Hari Purnomo., M.T., IPU, ASEAN.Eng, selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia, yang telah memberikan kebijakan dan dukungan bagi pelaksanaan program magang ini.
4. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Jurusan Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia, atas arahnya selama penulis menjalani proses magang.
5. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia, atas motivasi dan bimbingannya yang sangat berarti selama magang berlangsung.
6. Bapak Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D, selaku Dosen Pembimbing Akademik, yang telah memberikan nasihat serta arahan dalam pengembangan diri selama magang.
7. Bapak Dr. Ahmad Luthfi, S.Kom., M.Kom., Dosen Pembimbing Skripsi, yang telah memberikan masukan berharga yang relevan untuk menyempurnakan laporan ini.
8. PT PLN UP3 Yogyakarta, beserta seluruh jajaran pimpinan dan staf yang telah memberikan kesempatan, fasilitas, dan bimbingan selama program magang berlangsung.
9. Teman-teman dan rekan magang, atas kerjasama, semangat, dan dukungannya selama menjalani kegiatan ini.
10. Dan calon pasangan saya kedepannya, yang menemani proses skripsi ini hingga selesai.

Penulis berharap laporan ini dapat memberikan manfaat bagi pembaca dan menjadi referensi bagi mahasiswa lainnya yang akan melaksanakan magang. Semoga Allah SWT senantiasa memberikan keberkahan dan kemudahan dalam setiap langkah kita.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 1 November 2024

A handwritten signature in black ink, appearing to read 'Irfan Rafid Risdianto', with a stylized, cursive script.

(Irfan Rafid Risdianto)

SARI

Backend adalah bagian krusial dari sebuah aplikasi yang berfungsi di sisi server. Fungsinya mencakup pengelolaan interaksi dengan *database*, logika bisnis, dan memastikan kelancaran proses internal aplikasi. *Backend* juga mendukung komunikasi antara server dan antarmuka pengguna, serta memproses dan menyimpan informasi yang diperlukan. Salah satu aplikasi umum dari *backend* adalah *dashboard*, yang membantu perusahaan menyajikan data kinerja dalam format visual. *Dashboard* ini sering digunakan untuk memantau kinerja perusahaan secara *real-time*, memungkinkan pengambilan keputusan yang lebih tepat berdasarkan data aktual.

Di PT PLN UP3 Yogyakarta, teknologi *Automatic Meter Reading* (AMR) diterapkan sebagai bagian dari sistem ini. AMR secara otomatis membaca dan mengambil data meteran pelanggan melalui teknologi nirkabel, yang memungkinkan PLN untuk memantau beban listrik pelanggan serta mendeteksi potensi kerugian daya listrik, baik yang bersifat teknis maupun non-teknis. Implementasi AMR terbukti efektif dalam membantu PT PLN mengelola konsumsi listrik dan mengurangi kerugian daya.

Laporan ini bertujuan untuk merancang dan mengimplementasikan *backend* Sistem Informasi Bersama (Siber) di PT PLN UP3 Yogyakarta, termasuk integrasi *dashboard* AMR. *Dashboard* ini memungkinkan pemantauan data pelanggan secara *real-time* yang terhubung langsung dengan *database* perusahaan, sehingga meningkatkan efisiensi dan efektivitas operasional. Selain AMR, Divisi Perencanaan dan Divisi Konstruksi juga memanfaatkan sistem ini untuk memantau indikator kinerja dan transaksi energi listrik, mulai dari pengajuan hingga aktivasi di Unit Layanan Pelanggan (ULP) dan UP3. Dengan implementasi *backend* yang terintegrasi, PT PLN UP3 Yogyakarta dapat meningkatkan efisiensi operasional dan memberikan layanan yang lebih optimal bagi masyarakat.

Kata kunci: *Backend*, *Dashboard*, AMR, PLN, Kinerja, Operasional.

GLOSARIUM

Istilah	Definisi
SEO (Search Engine Optimization)	Proses meningkatkan peringkat situs di mesin pencari untuk meningkatkan trafik organik.
<i>Scrum</i>	Metodologi pengelolaan proyek berbasis Agile dengan siklus kerja iteratif (<i>sprint</i>).
CRUD (Create, Read, Update, Delete)	Empat operasi dasar untuk mengelola data: menambah, membaca, memperbarui, dan menghapus data.
RDBMS (Relational Database Management System)	Sistem manajemen basis data yang menyimpan data dalam tabel yang saling berhubungan.
Leaflet	Pustaka JavaScript untuk membuat peta interaktif di web.
Datatables	Pustaka JavaScript untuk membuat tabel interaktif dengan fitur pencarian, pengurutan, dan paging.
<i>Chart</i>	Visualisasi data dalam bentuk grafik, seperti batang, garis, atau pie.
Agile	Metode pengembangan perangkat lunak yang menekankan iterasi cepat, kolaborasi, dan fleksibilitas.
Retrospective	Pertemuan dalam Agile untuk mengevaluasi dan memperbaiki proses kerja setelah setiap iterasi.

DAFTAR ISI

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	xi
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Ruang Lingkup	4
1.3 Tujuan	5
1.4 Manfaat	5
1.5 Sistematika Penulisan	6
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA	7
2.1 Sistem Informasi	7
2.2 <i>Backend</i>	7
2.3 <i>Dashboard</i>	10
2.4 <i>Scrum</i>	10
2.5 PHP	12
2.6 MySQL	12
2.7 Git	13
2.8 AJAX	13
2.9 UAT	13
2.10 Tinjauan Pustaka	14
BAB III PELAKSANAAN MAGANG	16
3.1 Alur Pengembangan Sistem	16
3.2 Struktur tim <i>scrum</i>	18
3.3 Manajemen Proyek	19
3.4 Pengembangan Sistem Informasi Bersama (SiBer)	20
BAB IV REFLEKSI PELAKSANAAN MAGANG	49
4.1 Relevansi Akademik	49
4.2 Pembelajaran magang	52
BAB V PENUTUP	55
5.1 Kesimpulan	55
5.2 Saran	55
DAFTAR PUSTAKA	58
LAMPIRAN	59

DAFTAR GAMBAR

Gambar 1.1 Tampak Depan PT. PLN UP3 Yogyakarta.....	2
Gambar 1.2 Struktur organisasi PT. PLN UP3 Yogyakarta.....	3
Gambar 2.1 <i>Backend</i> Architecture Scheme.....	8
Gambar 2.2 Alur kerangka kerja <i>scrum</i>	11
Gambar 3.1 Alur pengembangan sistem	19
Gambar 3.2 Dokumentasi <i>briefing</i> dengan <i>Project Owner(PO)</i>	22
Gambar 3.3 Alur proses Transaksi Energi	24
Gambar 3.4 alur proses Perencanaan.....	25
Gambar 3.5 alur proses Konstruksi	25
Gambar 3.6 use case diagram Transaksi energi	27
Gambar 3.7 Use case diagram Perencanaan	27
Gambar 3.8 Use case diagram Konstruksi	28
Gambar 3.9 Potongan kode logika php pengambilan data transaksi energi	30
Gambar 3.10 Basis data pln.....	31
Gambar 3.11 Basis data simantri.....	32
Gambar 3.12 Projek Sistem Informasi Bersama	33
Gambar 3.13 Potongan kode logika php get data	35
Gambar 3.14 Tampilan penggunaan logika php pada konsol browser	35
Gambar 3.15 Potongan kode <code>updateChart()</code>	36
Gambar 3.16 Tampilan dari grafik pie <i>chart</i> dengan layer	37
Gambar 3.17 Potongan kode maps menggunakan leaflet.js.....	38
Gambar 3.18 visualisasi maps.....	38
Gambar 3.19 Potongan kode untuk datatables	39
Gambar 3.20 Tampilan datatables.....	40
Gambar 3.21 hasil akhir <i>dashboard</i> AMR.....	41
Gambar 3.22 potongan kode salah satu <i>scoreboard</i>	42
Gambar 3.23 contoh <i>scoreboard</i> pada menu pelanggan dan daya.....	43
Gambar 3.24 contoh potongan kode grafik daya terpasang	44

DAFTAR TABEL

Table 3.1 Struktur tim <i>scrum</i>	21
Table 3.2 Tabel fitur dan tabel yang digunakan pada basis data PLN	31
Table 3.3 Tabel fitur dan tabel yang digunakan pada basis data Simantri	32

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini, teknologi informasi merupakan sebuah kebutuhan dan informasi yang diinginkan bisa diberikan oleh teknologi informasi dengan sangat mudah dan praktis. Penerapan teknologi informasi memiliki banyak manfaat dalam menjalankan aktivitas. Hampir seluruh kegiatan yang terjadi tidak lepas dari teknologi informasi saat ini. Perkembangan teknologi informasi ini menjadikan kehidupan bermasyarakat lebih cepat dalam mendapatkan informasi.

Teknologi informasi dipergunakan dalam berbagai bidang termasuk dalam bidang energi. Salah satu perusahaan dalam bidang energi adalah PT PLN UP3 Yogyakarta yang aktivitasnya banyak memanfaatkan teknologi informasi.

PT Perusahaan Listrik Negara (PLN) Unit Pelaksanaan Pelayanan Pelanggan (UP3) merupakan perusahaan yang beroperasi di Yogyakarta. Unit ini memegang peranan penting untuk memberikan pasokan listrik di daerah jogjakarta dan juga memberikan pelayanan kepada masyarakat jogjakarta. Visi dari PT PLN adalah “Menjadi Perusahaan Listrik Terkemuka se-Asia Tenggara dan #1 Pilihan Pelanggan untuk Solusi Energi” dengan misi menjalankan bisnis kelistrikan dan bidang lain yang terkait, berorientasi dengan kepuasan pelanggan, anggota perusahaan, dan pemegang saham, menjadikan tenaga listrik sebagai media untuk meningkatkan kualitas kehidupan masyarakat, mengupayakan tenaga listrik menjadi pendorong kegiatan ekonomi, menjalankan kegiatan usaha yang berwawasan lingkungan. Semua itu dipadukan dengan moto listrik untuk kehidupan yang lebih baik. (PLN , 2023)

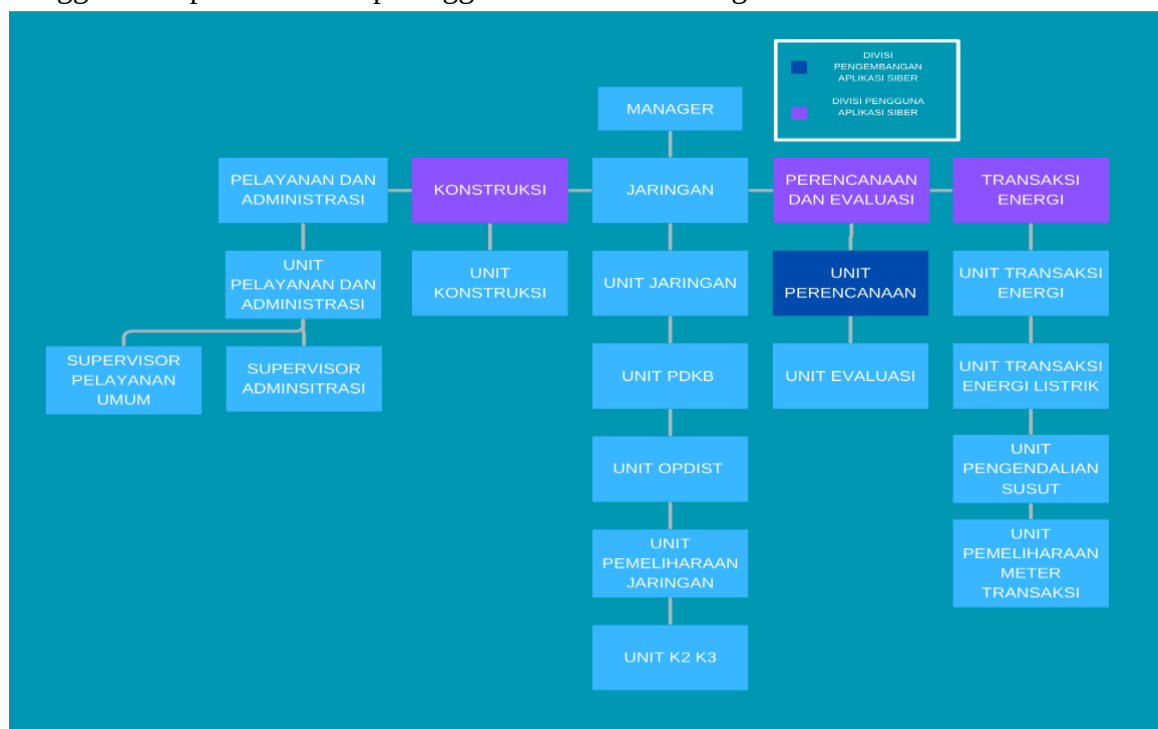


Gambar 1.1 Tampak Depan PT. PLN UP3 Yogyakarta

Teknologi informasi sangat membantu kinerja operasional dan pelayanan pelanggan PT PLN UP3 Yogyakarta. Namun demikian, penerapan teknologi informasi di PT PLN UP3 Yogyakarta masih dapat dilakukan pengembangan lebih lanjut.

PT Perusahaan Listrik Negara (PLN) Unit Pelaksanaan Pelayanan Pelanggan (UP3) Yogyakarta juga memiliki struktur organisasi yang mendukung pencapaian visi dan misinya. Salah satu divisi penting dalam organisasi ini adalah Divisi Perencanaan, yang berperan strategis dalam menyusun rencana operasional perusahaan. Divisi ini juga bertanggung jawab untuk mengembangkan berbagai aplikasi sistem informasi. Salah satu aplikasi yang dibutuhkan adalah Sistem Informasi Bersama (SIBER), sebuah platform yang dirancang dengan tujuan untuk meningkatkan efisiensi dan kolaborasi antara berbagai divisi di PLN UP3 Yogyakarta. Aplikasi ini diharapkan dapat mendukung koordinasi dalam proses perencanaan dan operasional, memastikan bahwa pasokan listrik berjalan lancar dan pelayanan kepada masyarakat tetap optimal.

Selain Divisi Perencanaan, terdapat divisi-divisi lain yang mendukung operasional PLN, seperti Divisi Konstruksi yang berfokus pada pembangunan infrastruktur listrik, Divisi Jaringan yang bertanggung jawab atas distribusi dan pemeliharaan jaringan, Divisi Pelayanan dan Administrasi yang memastikan layanan pelanggan berjalan dengan baik, serta Divisi Transaksi Energi yang menangani transaksi energi dan pengelolaan meteran listrik. Sinergi antar divisi ini, seperti yang ditampilkan dalam Gambar 1.2, memungkinkan PLN UP3 Yogyakarta untuk terus mewujudkan visinya sebagai perusahaan listrik terkemuka di Asia Tenggara dan pilihan utama pelanggan untuk solusi energi.



Gambar 1.2 Struktur organisasi PT. PLN UP3 Yogyakarta

Aplikasi Sistem Informasi Bersama (SIBER) ini secara khusus digunakan oleh tiga divisi utama di PT. PLN UP3 Yogyakarta, yaitu Divisi Perencanaan, Divisi Transaksi Energi, dan Divisi Konstruksi. Divisi Perencanaan menggunakan aplikasi ini untuk merencanakan dan mengoordinasikan proyek-proyek strategis secara lebih efisien. Divisi Transaksi Energi memanfaatkan Siber untuk memantau dan mengelola transaksi energi, sementara Divisi Konstruksi menggunakan aplikasi ini untuk memfasilitasi pembangunan infrastruktur kelistrikan yang terintegrasi dengan rencana dan kebutuhan jaringan yang ada. Dengan Siber, ketiga divisi ini dapat berkolaborasi lebih baik dan memastikan kelancaran operasional yang mendukung pelayanan optimal kepada masyarakat, seperti pada Gambar 1.2.

Salah satu inovasi layanan yang ada di PT PLN UP3 Yogyakarta saat ini adalah sistem AMR (*Automatic Meter Reading*). Teknologi ini diterapkan pada APP (Alat Pengukur dan Pembatas) pada kWh meter listrik. Aplikasi *Automatic Meter Reading* (AMR) merupakan alat bantu untuk *monitoring*, analisis dan evaluasi menentukan penyebab terjadinya pelanggaran yang dilakukan oleh pelanggan yang mempunyai daya-daya besar. AMR ini juga dapat mengetahui profil beban pelanggan berupa tegangan, arus, dan pemakaian kWh serta konfigurasi pengawatan, yang akhirnya dapat diketahui penyebab penurunan jam nyala yang sangat signifikan.

Kondisi saat ini, data yang tercatat melalui device AMR hanya terbaca sebagai data excel yang masih memerlukan proses manual untuk menjadi data laporan yang lebih spesifik sesuai kebutuhan. Proses manual ini kurang efisien secara sumber daya manusia dan waktu. Untuk itu diperlukan upaya penambahan aplikasi yang dapat mempersingkat proses pengolahan data AMR menjadi output data yang akurat.

Untuk itu penulis melakukan observasi dan membuat aplikasi Sistem Informasi Bersama (SIBER) yang nantinya dapat mengolah data yang menghasilkan output data akurat lebih cepat dan meningkatkan kinerja operasional dan pelayanan pelanggan serta mendukung integrasi antara 3 divisi yang berbeda. Dalam prosesnya diperlukan pembuatan dashboard dengan dukungan backend yang sesuai.

Dashboard merupakan sebuah model aplikasi sistem informasi yang disediakan untuk menyajikan informasi kualitas kinerja, dari sebuah perusahaan atau lembaga organisasi. Konsep kinerja *dashboard* sudah ada selama bertahun-tahun. *Dashboard* telah banyak diadopsi oleh perusahaan atau kalangan bisnis. Salah satu contohnya pada tahun 2004, survei yang dilakukan oleh institusi Data Warehousing ada sekitar 473 perusahaan menggunakan *dashboard* sebagai *Business Intelligence* informasi organisasinya. Survei terbaru yang dilakukan oleh Gartner Inc. menemukan bahwa *dashboard* menggantikan pelaporan dan analisis ad-hoc dalam sebuah organisasi perusahaan di Negara Barat (Few, 2013)

Backend adalah tempat di mana proses pada suatu sistem informasi atau aplikasi

berjalan, data dapat ditambahkan, diubah, maupun dihapus. Lebih lanjut, backend adalah suatu bagian dari aplikasi yang beroperasi pada sisi server dan bertanggung jawab untuk berinteraksi langsung dengan database, khususnya terkait manipulasi data seperti menyimpan, mengambil, memperbarui, dan menghapus data. Backend biasanya mengurus segala jenis proses yang tidak berhubungan langsung dengan pengguna, seperti server dan basis data. Backend dibutuhkan dalam pengembangan sistem dan manajemen data pada sistem

Secara khusus, pengembangan *backend* dengan menggunakan *dashboard* AMR diharapkan dapat memvisualisasi data yang informatif seperti *chart* dan tabel yang berguna untuk membantu pengguna memahami situasi secara cepat dan efisien. Selain itu, juga diharapkan dapat menganalisis data tegangan dengan tiga kombinasi untuk mengidentifikasi kondisi normal, memerlukan perhatian, atau memerlukan perbaikan

1.2 Ruang Lingkup

Ruang lingkup laporan ini mencakup proyek yang dikerjakan oleh penulis. Kegiatan magang ini dilakukan di kantor PT. PLN UP3 Yogyakarta, yang berlokasi di Jl. Gedongkuning No.3, Tegal Tandan, Banguntapan, Kec. Banguntapan, Kabupaten Bantul, Daerah Istimewa Yogyakarta 55198. Kegiatan magang ini dilaksanakan dari tanggal 15 September 2023 hingga 15 Februari 2024.

Pelaksanaan kegiatan magang di PT. PLN UP3 Yogyakarta dilakukan selama enam bulan, dengan fokus utama pada pengembangan sistem *backend* aplikasi Siber. Adapun aktivitas yang dilakukan selama magang adalah sebagai berikut:

- 1) Pembuatan basis data aplikasi Siber: membangun basis data untuk mendukung operasi *backend* aplikasi Siber, memastikan integrasi data antar-divisi berjalan dengan lancar.
- 2) Pembuatan logika dalam bahasa php pengambilan data: merancang dan mengimplementasikan untuk pengambilan data secara efisien dari basis data.
- 3) Pembuatan logika dalam bahasa php *import* data: mengembangkan untuk proses *import* data, termasuk sinkronisasi data dari file Excel ke dalam *database* perusahaan.
- 4) Dashboard Transaksi Energi:

Pembuatan *dashboard* dengan fitur utama:

- a) Pencarian dan edit data pelanggan.
- b) *Import* data pelanggan dari Excel ke *database*.
- c) Pembuatan peta dengan titik koordinat pelanggan menggunakan pin.
- d) Pencarian dan sortir data pelanggan berdasarkan daerah, kondisi, dan jenis pelanggan.

- 5) Dashboard Perencanaan:

Menyediakan *chart* dan *cardboard* yang menampilkan:

- a) Data pelanggan dan daya listrik.
- b) SAIDI, SAIFI, gangguan 100kms, gangguan reclosure, dan recovery time.
- c) Data susut energi, kWh jual, dan kWh produksi.

d) HPL tanpa perluasan, HPL perluasan, dan HPL TM.

6) Dashboard Konstruksi:

Pengembangan alur permohonan transaksi energi listrik yang mencakup:

- a) Permohonan dari masing-masing ULP dan dari semua ULP untuk halaman UP3.
- b) Transaksi yang sudah dibayar dan yang sedang diproses di masing-masing ULP dan di UP3.
- c) Transaksi yang sudah menyala di masing-masing ULP dan di UP3.

Kegiatan-kegiatan ini merupakan bagian penting dari pengembangan sistem Siber, yang dirancang untuk mendukung efisiensi operasional di PT. PLN UP3 Yogyakarta.

1.3 Tujuan

Tujuan pengerjaan proyek magang ini adalah untuk mengembangkan sistem yang dapat membantu PT. PLN UP3 Yogyakarta dalam menangani data dari tiga divisi yang berbeda. Selain itu, proyek ini juga bertujuan untuk mengembangkan sistem yang mendukung PT. PLN UP3 Yogyakarta dalam mengoptimalkan pelayanan kepada pelanggan. Dengan pengembangan sistem ini, diharapkan PT. PLN UP3 Yogyakarta dapat meningkatkan efisiensi dalam pengelolaan data dan memberikan pelayanan yang lebih baik kepada pelanggan. Proyek magang ini bertujuan untuk memberikan solusi yang relevan dan aplikatif, sehingga dapat mendukung keberlanjutan dan peningkatan mutu layanan secara keseluruhan.

1.4 Manfaat

Manfaat dari pengerjaan proyek magang ini adalah untuk meningkatkan efisiensi pengelolaan data dari tiga divisi yang berbeda di PT. PLN UP3 Yogyakarta, sehingga memungkinkan pengelolaan data yang lebih terstruktur dan cepat. Selain itu, proyek ini juga bertujuan untuk meningkatkan kualitas pelayanan kepada pelanggan, di mana sistem yang dikembangkan akan memungkinkan PT. PLN UP3 Yogyakarta memberikan respon yang lebih cepat dan akurat terhadap permintaan dan keluhan pelanggan. Dengan adanya sistem ini, diharapkan PT. PLN UP3 Yogyakarta dapat melakukan analisis data pelanggan yang lebih mendalam, sehingga dapat memahami kebutuhan dan preferensi pelanggan dengan lebih baik. Akhirnya, dengan pelayanan yang lebih baik dan responsif, proyek ini diharapkan dapat meningkatkan loyalitas pelanggan terhadap PT. PLN UP3 Yogyakarta, yang pada gilirannya akan berdampak positif pada reputasi dan pertumbuhan perusahaan secara keseluruhan.

1.5 Sistematika Penulisan

Laporan proyek ini disusun dalam beberapa bab dengan tujuan memudahkan pencarian informasi yang diperlukan, serta menampilkan penyelesaian pekerjaan secara sistematis. Pembagian bab tersebut meliputi:

BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang, ruang lingkup magang, tujuan dan manfaat, serta sistematika penulisan untuk pengembangan sistem informasi bersama (SIBER).

BAB II LANDASAN TEORI DAN KAJIAN PUSTAKA

Bab ini berisi teori-teori dan penjelasan yang diperoleh dari kajian literatur yang relevan dengan penyusunan laporan pengembangan sistem informasi bersama (SIBER).

BAB III PELAKSANAAN MAGANG

Bab ini mendokumentasikan tugas dan aktivitas yang dilaksanakan selama kegiatan magang berlangsung.

BAB IV REFLEKSI PELAKSANAAN MAGANG

Bab ini memaparkan relevansi akademis serta pembelajaran yang diperoleh dari pengalaman pelaksanaan magang.

BAB V KESIMPULAN DAN SARAN

Bab ini memberikan kesimpulan dari pembahasan sebelumnya serta saran untuk perbaikan dan pengembangan lebih lanjut.

DAFTAR PUSTAKA

LAMPIRAN

BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 Sistem Informasi

Sistem informasi terdiri dari komponen-komponen seperti perangkat keras, perangkat lunak, data, prosedur, dan manusia yang bekerja bersama untuk mengelola dan mendistribusikan informasi, membantu pengambilan keputusan dalam organisasi. Sistem ini berfungsi untuk mengolah data menjadi informasi yang berguna, memungkinkan operasi yang efisien. (Erwan *et al*, 2023).

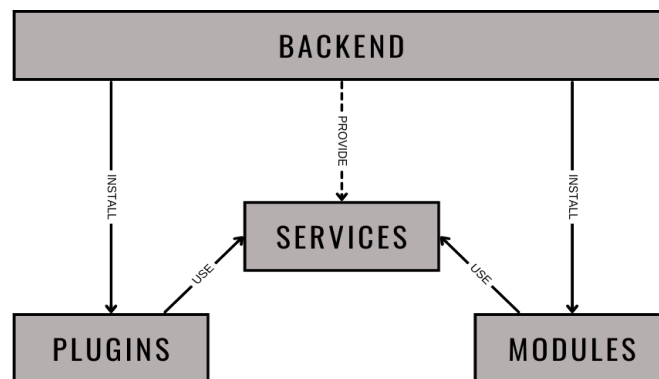
Sistem informasi membantu mengotomatiskan proses bisnis, memastikan operasi berjalan lancar, dan menyederhanakan pengambilan keputusan melalui data terstruktur. Integrasi sistem informasi dengan teknologi baru seperti big data dan kecerdasan buatan semakin meningkatkan efisiensi dan keandalan sistem ini (Laudon, 2018)

2.2 Backend

Backend adalah bagian dari aplikasi yang berjalan di server dan bertanggung jawab untuk logika bisnis, interaksi *database*, dan pemrosesan data. *Backend* menerima permintaan dari frontend, memprosesnya, dan mengirim respons yang sesuai. Bahasa pemrograman seperti PHP, Python, dan Node.js sering digunakan untuk membangun logika *backend*. *Backend* sangat penting karena merupakan fondasi aplikasi yang mendukung semua operasi internal. Keamanan, efisiensi, dan skalabilitas aplikasi bergantung pada pengelolaan *backend* yang baik (Severance, 2012)

2.2.1 Backend Development Architecture

Backend Development Architecture adalah bagian dari arsitektur sistem yang berfokus pada bagaimana komponen *backend* dibangun, diorganisasikan, dan diintegrasikan dalam sebuah aplikasi. Dalam konteks pengembangan perangkat lunak, *backend* merupakan bagian dari aplikasi yang berjalan di sisi server, bertanggung jawab atas pemrosesan data, manajemen logika bisnis, serta interaksi dengan *database* atau API eksternal.



Gambar 2.1 *Backend Architecture Scheme*

(Sumber : backstage.io)

Pada **Gambar 2.1**, dapat dilihat representasi sederhana dari Arsitektur Pengembangan *Backend* yang menggambarkan bagaimana berbagai komponen *backend* saling berinteraksi. Komponen utama yang ditampilkan meliputi *Backend*, *Service*, *Modules*, dan *Plugins*. Berikut ini adalah penjelasan lebih rinci mengenai **Gambar 2.1**:

1. *Service*:

Penjelasan: *Service* adalah komponen yang berfungsi sebagai penyedia layanan dan pengelolaan data antara *backend* dan komponen lain seperti *modules* dan *plugins*. Fungsinya mencakup pengambilan data dari sumber seperti *database*, pemrosesan logika bisnis, dan menyiapkan data untuk dikonsumsi oleh modul atau *plugin* dalam aplikasi.

Contoh:

- a) REST API: *Service* ini menyajikan data produk dari *database* untuk *aplikasi e-commerce*. Saat *frontend* meminta data produk, REST API bertugas mengambil informasi seperti nama produk, harga, dan stok dari *database*, kemudian mengirimkan data tersebut dalam format JSON ke modul yang memerlukan.
- b) Auth Service: Layanan yang menangani otentikasi dan verifikasi pengguna. Misalnya, saat pengguna mencoba *login*, Auth Service akan memeriksa kredensial, dan jika berhasil, memberikan akses token yang memungkinkan pengguna untuk mengakses fitur aplikasi.

2. *Modules*:

Penjelasan: *Modules* adalah komponen yang mengimplementasikan fungsionalitas spesifik dalam aplikasi. Modul ini bertindak sebagai *user-facing component*, yang biasanya berinteraksi dengan pengguna dan menampilkan data dari *backend* melalui Service.

Contoh:

- c) *Dashboard Module*: Modul *dashboard* yang menampilkan ringkasan informasi penting seperti penjualan harian, performa produk, dan laporan keuangan. Modul ini menggunakan data dari Service API untuk menampilkan grafik, tabel, atau statistik yang mudah dipahami pengguna.
- d) *User Management Module*: Modul ini memungkinkan admin untuk mengelola pengguna dalam aplikasi. Admin bisa menambahkan, menghapus, atau mengedit pengguna. Modul ini menggunakan *Auth Service* untuk otorisasi dan akses data pengguna dari *database*.

3. *Plugins*:

Penjelasan: *Plugins* adalah komponen tambahan yang memperluas fungsionalitas aplikasi utama. *Plugin* dapat diinstal dan dihapus tanpa mempengaruhi struktur utama aplikasi. Mereka sering berinteraksi dengan *Service* atau *Modules* untuk menambah fitur tertentu.

Contoh:

- e) *DataTables Plugin*: *Plugin* ini menambahkan fitur pencarian, filter, dan pengurutan di tabel data yang digunakan oleh modul manajemen data. *DataTables Plugin* memanfaatkan data dari Service API untuk menampilkan tabel interaktif yang dapat diakses pengguna dengan mudah.
- f) *SEO Plugin*: *Plugin* SEO digunakan dalam aplikasi manajemen konten (CMS) untuk menganalisis dan mengoptimalkan halaman agar lebih ramah mesin pencari. *Plugin* ini memanfaatkan data dari *backend* untuk memberikan saran optimasi dan analisis performa SEO dari halaman tertentu.

Secara keseluruhan, arsitektur ini menunjukkan bagaimana berbagai komponen seperti *Service*, *Modules*, dan *Plugins* berinteraksi dalam ekosistem *backend*. Setiap komponen memiliki peran spesifik yang mendukung alur kerja aplikasi secara menyeluruh, baik untuk menyediakan data, menjalankan logika bisnis, maupun memberikan fitur tambahan yang memperkaya pengalaman pengguna.

2.3 Dashboard

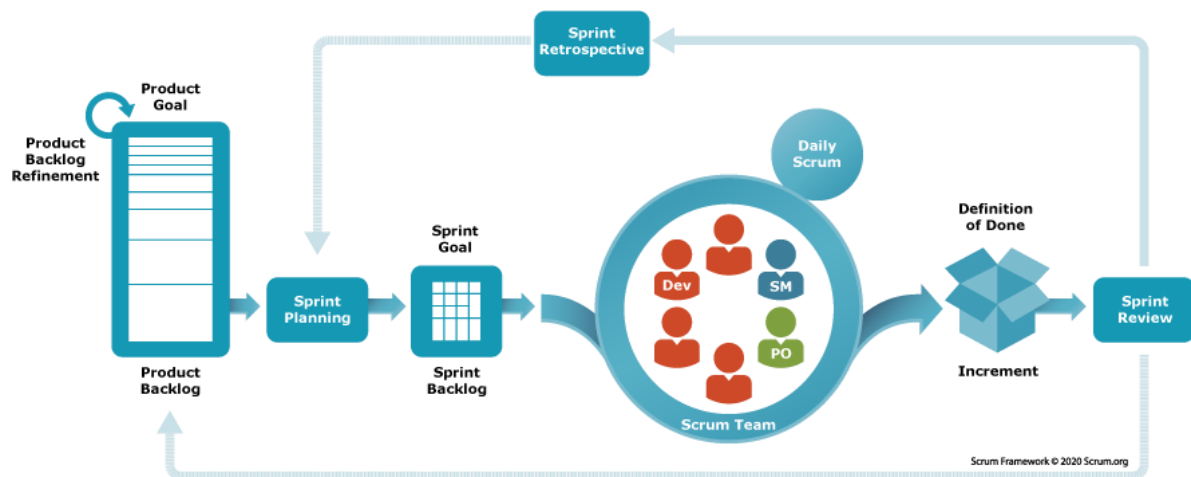
Dashboard adalah antarmuka yang digunakan untuk menyajikan informasi penting secara visual, sering kali dalam bentuk grafik atau tabel (Few, 2013). Ini membantu pengguna memantau kinerja perusahaan secara *real-time* dan membuat keputusan strategis berdasarkan data. *Dashboard* dirancang untuk menyajikan KPI atau metrik penting yang relevan untuk manajemen. Penggunaan *dashboard* telah berkembang di berbagai sektor bisnis karena memungkinkan akses cepat ke informasi yang diperlukan untuk pengambilan keputusan (Few, 2013).

2.4 Scrum

Scrum adalah kerangka kerja manajemen proyek yang digunakan secara luas dalam pengembangan perangkat lunak dengan metodologi Agile. *Scrum* memfasilitasi tim untuk bekerja dalam iterasi singkat yang disebut "*sprint*," biasanya berdurasi antara 2 hingga 4 minggu, di mana setiap *sprint* memiliki tujuan untuk menghasilkan inkremental produk yang berfungsi. Setiap *sprint* didorong oleh *Product Backlog*, yang merupakan daftar prioritas pekerjaan yang diatur oleh *Product Owner*. Produk yang dihasilkan dalam setiap *sprint* harus dapat digunakan atau "*potentially shippable*" (Sutherland, 2017). Penerapan *Scrum* dalam tim pengembangan perangkat lunak melibatkan peran yang jelas seperti *Scrum Master*, *Product Owner*, dan tim pengembangan. *Scrum Master* bertanggung jawab untuk memfasilitasi proses dan memastikan tim mengikuti prinsip-prinsip *Scrum*, sedangkan *Product Owner* menentukan prioritas *backlog* sesuai dengan kebutuhan bisnis. Dalam setiap *sprint*, tim mengadakan beberapa pertemuan seperti *Sprint Planning* untuk merencanakan pekerjaan, *Daily Scrum* untuk pembaruan harian, dan *Sprint Review* untuk mendemonstrasikan hasil kerja. Setelah itu, tim melakukan *Sprint Retrospective* untuk mengevaluasi apa yang bisa diperbaiki di *sprint* berikutnya (Cohn, 2010). Pendekatan iteratif ini memungkinkan tim untuk tetap fleksibel dan beradaptasi dengan perubahan selama proyek berlangsung.

2.4.1 Alur Scrum

Kerangka kerja *Scrum* memiliki alur yang terstruktur dan berfokus pada efisiensi serta adaptasi selama pengembangan proyek seperti yang ditunjukkan pada gambar 2.1.



Gambar 2.2 Alur kerangka kerja *scrum*

(sumber : Scrum.org. (n.d.). learning series. Scrum.org. <https://www.scrum.org>)

Berikut ini adalah penjelasan dari alur kerangka kerja *scrum*:

a) Product Back Log

Merupakan daftar prioritas pekerjaan yang mencakup semua fitur, perbaikan, dan tugas yang diperlukan untuk pengembangan produk. *Product Backlog* dikelola oleh *Product Owner* dan selalu berkembang berdasarkan masukan dari tim dan pemangku kepentingan. Setiap *item* dalam *backlog* diprioritaskan sesuai kebutuhan bisnis dan nilai yang dihasilkan bagi produk.

b) Sprint Planning

Merupakan pertemuan yang diadakan pada awal setiap *sprint*. Selama *Sprint Planning*, tim pengembang, *Product Owner*, dan *Scrum Master* berkolaborasi untuk memilih item dari *Product Backlog* yang akan dikerjakan dalam *sprint* tersebut. Tim kemudian menentukan tujuan *sprint* (*Sprint Goal*) dan merinci tugas-tugas yang diperlukan untuk mencapai target tersebut.

c) Sprint

Merupakan periode waktu yang ditetapkan, biasanya antara 2 hingga 4 minggu, di mana tim pengembang bekerja untuk menyelesaikan item yang dipilih selama *Sprint Planning*. Selama *Sprint*, tim berfokus pada penyelesaian pekerjaan sesuai prioritas tanpa interupsi dari luar. Setiap hari, tim melakukan *Daily Scrum* atau *Standup Meeting* untuk melaporkan kemajuan dan mengatasi hambatan.

d) Sprint Review

Merupakan pertemuan yang diadakan di akhir setiap *sprint* di mana tim pengembang mempresentasikan hasil kerja mereka kepada pemangku kepentingan. Tujuannya adalah untuk mendapatkan umpan balik dari pengguna atau pemilik produk dan menentukan apakah pekerjaan yang dilakukan sudah memenuhi kebutuhan. Hasil dari *Sprint Review* akan mempengaruhi pembaruan *Product Backlog* untuk *sprint* berikutnya.

2.5 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side* yang dirancang khusus untuk pengembangan web. PHP digunakan untuk menangani logika *backend* dan berinteraksi dengan *database*, menjadikannya sangat cocok untuk membuat situs web dinamis. PHP dapat mengintegrasikan HTML, CSS, dan JavaScript, sehingga memungkinkan pengembang menciptakan halaman web yang interaktif. Selain itu, PHP kompatibel dengan berbagai sistem manajemen basis data seperti MySQL, PostgreSQL, dan Oracle.

Karena sifatnya yang *open source* dan mudah dipelajari, PHP telah menjadi pilihan populer bagi banyak pengembang web di seluruh dunia. Banyak platform web besar seperti WordPress dan Drupal dibangun dengan PHP. Kemampuan PHP untuk menangani tugas-tugas seperti otentikasi pengguna, pengelolaan sesi, pengiriman email, serta manipulasi file menjadikannya alat yang kuat untuk membangun aplikasi web yang kompleks. Popularitas PHP berasal dari kemampuannya untuk menangani tugas-tugas *server-side* dengan cepat dan efisien, serta komunitas *open-source* yang besar yang mendukung pengembangan PHP (Lerdorf, 1995).

2.6 MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang menggunakan SQL (*Structured Query Language*) sebagai bahasa utama untuk mengelola data. MySQL memungkinkan pengembang untuk membuat, mengelola, dan memelihara basis data dengan *query* yang efisien. Data dalam MySQL diatur dalam tabel yang saling berhubungan, sehingga memudahkan untuk menyimpan dan mengambil informasi secara struktural. MySQL sering digunakan dalam pengembangan aplikasi web, terutama yang berbasis PHP, untuk menyimpan data pengguna, konten, dan informasi transaksi.

MySQL dikenal karena kehandalannya, skalabilitasnya, dan kecepatan prosesnya, sehingga banyak digunakan oleh perusahaan besar dan startup. Karena merupakan perangkat lunak *open-source*, MySQL tersedia secara gratis dan telah digunakan dalam banyak aplikasi skala besar seperti Facebook, YouTube, dan Twitter. Fitur MySQL yang kuat dan fleksibel menjadikannya salah satu sistem basis data terpopuler di dunia. MySQL populer karena

kemampuannya menangani aplikasi berskala besar dengan performa tinggi, serta merupakan bagian integral dari teknologi LAMP (Linux, Apache, MySQL, PHP) stack (DuBois, 2013)

2.7 Git

Git adalah sistem kontrol versi terdistribusi yang memungkinkan pengembang melacak perubahan kode secara *real-time* dan bekerja secara kolaboratif. Dengan fitur seperti branching dan merging, pengembang dapat bekerja pada fitur atau perbaikan secara paralel tanpa mengganggu kode utama. Git memudahkan pengembang bekerja dalam tim dengan sistem repositori seperti GitHub atau GitLab, memastikan setiap perubahan dapat dikelola dengan baik (Straub, 2014)

2.8 AJAX

AJAX (*Asynchronous JavaScript and XML*) adalah teknik pengembangan web yang memungkinkan pemuatan data tanpa harus memuat ulang seluruh halaman. AJAX memungkinkan aplikasi untuk mengambil dan menampilkan data secara asinkron, meningkatkan pengalaman pengguna dan kinerja aplikasi.

AJAX banyak digunakan untuk membuat aplikasi web dinamis, seperti pencarian otomatis dan pembaruan tabel tanpa *refresh*. Penggunaan AJAX membuat aplikasi lebih cepat dan responsif (Garrett, 2005)

2.9 UAT

UAT (*User Acceptance Testing*) adalah tahap akhir dalam siklus pengembangan perangkat lunak di mana pengguna akhir atau klien melakukan pengujian untuk memastikan bahwa sistem memenuhi kebutuhan dan persyaratan bisnis. UAT bertujuan untuk memverifikasi bahwa aplikasi yang telah dikembangkan sesuai dengan spesifikasi dan siap untuk digunakan dalam lingkungan produksi. Biasanya, UAT dilakukan di lingkungan yang menyerupai kondisi operasional sebenarnya, dan hasilnya menentukan apakah perangkat lunak dapat diterima atau perlu revisi lebih lanjut.

UAT adalah langkah penting karena memberikan kesempatan kepada pengguna akhir untuk mengidentifikasi masalah atau kekurangan yang mungkin tidak terdeteksi selama pengujian teknis atau fungsional sebelumnya. Dengan melibatkan pengguna dalam proses pengujian ini, pengembang dapat memastikan bahwa produk akhir berfungsi sesuai dengan harapan pengguna dan kebutuhan bisnis, sehingga meminimalkan risiko kesalahan saat implementasi (Myers, 2011)

2.10 Tinjauan Pustaka

Dalam tinjauan pustaka, penulis menyajikan berbagai hasil penelitian yang memiliki relevansi dengan penelitian ini. Penyajian ini bertujuan untuk menunjukkan keterkaitan antara penelitian yang sedang dikerjakan dengan penelitian-penelitian sebelumnya. Hasil-hasil penelitian yang relevan tersebut diuraikan secara terperinci untuk memberikan gambaran yang lebih jelas mengenai kontribusi penelitian sebelumnya terhadap dasar teori dan metodologi penelitian ini.

Penelitian terkait Perancangan Sistem Informasi Pengelolaan Sarana dan Prasarana dengan Pengujian *User Acceptance Testing* (UAT). Metode UAT digunakan dalam tahap akhir pengembangan sistem untuk memastikan bahwa semua fitur telah memenuhi kebutuhan pengguna dan dapat digunakan tanpa hambatan signifikan (Wibowo & Darwati, 2022).

Penelitian tentang Penerapan AJAX dalam Aplikasi Mobile Berbasis Web untuk Meningkatkan Efisiensi Bandwidth Penggunaan AJAX dalam aplikasi mobile berbasis web bertujuan untuk mengurangi konsumsi bandwidth dan mempercepat pemuatan data tanpa perlu melakukan refresh halaman secara keseluruhan (Wijaya, 2018)

Penelitian tentang Penggunaan Metode *Scrum* dalam Penerapan Metode Agile Scrum Pada Perancangan Sistem Informasi Pelaporan keuangan BUMDesa menunjukkan pendekatan iteratif dan inkremental dalam Agile Scrum memungkinkan pengembangan sistem dilakukan dalam siklus pendek yang disebut sprint. Setiap sprint menghasilkan suatu fitur yang siap diuji oleh pengguna (Susilo & Azimah, 2023)

Penelitian terkait Sistem Informasi Perpustakaan Menggunakan PHP Native dan Database MySQL Berbasis Web pada SMP Negeri 2 Dawan menunjukkan bahwa Penggunaan PHP dalam pengembangan sistem perpustakaan memungkinkan pengelolaan data yang lebih fleksibel (Supriatmaja et al., 2023)

Penelitian terkait penggunaan aplikasi *database* MySQL pada Perancangan Sistem Informasi Persediaan Barang Berbasis Web. Penggunaan MySQL dalam sistem informasi persediaan barang berbasis web memungkinkan integrasi data yang lebih aman dan efisien, terutama dalam mengelola transaksi dan laporan stok barang (Rohman & Bhakti, 2023) .

Penelitian terkait *User Acceptance Testing* (UAT) pada Sistem Informasi Manajemen Pendidikan Berbasis Web melibatkan simulasi pengguna akhir untuk mengevaluasi kegunaan sistem. Proses UAT meliputi pengujian fungsionalitas, antarmuka pengguna, dan kompatibilitas. Hasil penelitian menunjukkan sistem memenuhi kebutuhan pengguna dengan tingkat kepuasan 85%, meskipun terdapat rekomendasi untuk peningkatan fitur tertentu (Syachbana, 2022)

Dari hasil tinjauan beberapa penelitian yang telah dikemukakan, terdapat beberapa persamaan. Beberapa poin persamaan di antaranya:

1. Metodologi pengembangan: Sama-sama menggunakan metode *Scrum* dalam pengembangan sistem.
2. Alur metodologi: Sama-sama mengikuti seluruh tahapan metodologi pengembangan, termasuk *sprint planning*, *product backlog*, dan *sprint review*.
3. Produk yang dikembangkan: Sama-sama mengembangkan sistem informasi yang berbasis web, dengan fokus pada pengelolaan data dan interaksi antara *backend* dan *frontend*.
4. Teknologi *backend*: Sama-sama menggunakan PHP sebagai bahasa pemrograman dan MySQL sebagai sistem manajemen basis data serta menggunakan AJAX sebagai metode untuk mengambil dan menampilkan data

Beberapa poin yang berbeda dari penelitian ini dengan penelitian yang ditinjau adalah sebagai berikut:

1. Fokus pengembangan: Penelitian ini berfokus pada pengembangan sistem informasi kelistrikan untuk manajemen data di PT. PLN UP3 Yogyakarta.
2. Teknologi tambahan: Penelitian ini mengintegrasikan teknologi *Automatic Meter Reading* (AMR) yang tidak ditemukan pada penelitian lainnya.

BAB III

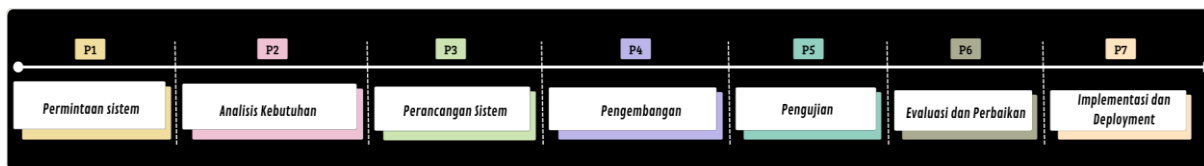
PELAKSANAAN MAGANG

3.1 Alur Pengembangan Sistem

Setiap perusahaan memiliki proses pengembangan sistem yang berbeda di lingkungan kerja, termasuk PT PLN UP3 Yogyakarta. Dalam hal ini, pengembangan sistem dan aplikasi sepenuhnya dilakukan oleh Divisi Perencanaan, tanpa perlu penggabungan divisi lain. Divisi Perencanaan bertanggung jawab sebagai pusat pengembangan berbagai aplikasi dan sistem teknologi informasi yang mendukung operasi perusahaan.

Tidak seperti perusahaan lain yang mungkin harus mengintegrasikan tim IT dari berbagai departemen, PT PLN UP3 Yogyakarta memiliki alur yang terorganisir. Permintaan untuk pengembangan aplikasi berasal dari divisi terkait, yang kemudian diarahkan langsung ke Divisi Perencanaan untuk ditindaklanjuti. Proses pengembangan aplikasi, mulai dari permintaan hingga pelaksanaan, dilakukan sepenuhnya oleh tim internal di divisi ini.

Alur pengembangan dari Divisi Perencanaan terdiri dari 7 tahapan. Seperti yang ditunjukkan pada Gambar 3.1, alur pengembangan sistem pada Divisi Perencanaan dimulai dari permintaan sistem, analisis kebutuhan, perancangan sistem, pengembangan, pengujian, evaluasi, dan implementasi.



Gambar 3.1 Alur pengembangan sistem

Adapun penjelasan secara rinci tentang alur pengembangan sistem di Divisi Perencanaan sebagai berikut:

1. Permintaan sistem

Proses pengembangan sistem dimulai dengan adanya kebutuhan dari divisi operasional di PT PLN UP3 Yogyakarta. Divisi yang membutuhkan sistem baru atau perbaikan fitur mengajukan permintaan kepada Divisi Perencanaan. Permintaan ini bisa berupa pengembangan aplikasi baru, pembaruan fitur pada sistem yang sudah ada, atau peningkatan performa sistem. Pengajuan permintaan ini harus didokumentasikan secara jelas, mencakup rincian spesifik mengenai kebutuhan dan tujuan pengembangan sistem.

2. Analisis kebutuhan

Setelah menerima permintaan, tim di Divisi Perencanaan melakukan analisis kebutuhan. Tahap ini mencakup wawancara dengan pihak-pihak terkait untuk mengumpulkan informasi yang dibutuhkan terkait sistem yang akan dikembangkan. Tim juga meninjau dokumen internal dan standar operasional yang berlaku untuk memastikan bahwa semua persyaratan sistem tercatat secara lengkap. Hasil dari analisis ini adalah dokumen spesifikasi yang mencakup kebutuhan fungsional (fungsi-fungsi yang harus ada dalam sistem) dan non-fungsional (misalnya, performa sistem, keamanan, dan kompatibilitas).

3. Perancangan sistem

Tahap perancangan sistem dilakukan setelah kebutuhan sistem diidentifikasi. Tim pengembang mulai merancang struktur *backend* sistem, mencakup desain arsitektur sistem, diagram alur data, dan model basis data. Dengan adanya desain yang matang, tim dapat memastikan bahwa sistem yang dikembangkan memenuhi kebutuhan pengguna dan dapat dikembangkan secara efisien. Desain ini juga menjadi acuan bagi penulis selama tahap implementasi.

4. Pengembangan

Tahap pengembangan dimulai setelah desain selesai. Di sini, Divisi Perencanaan menggunakan metodologi *Scrum* untuk mengelola proyek pengembangan secara bertahap dan berkelanjutan. Pengembangan sistem dibagi menjadi beberapa *sprint*, biasanya dengan durasi dua minggu. Pada awal setiap *sprint*, dilakukan *Sprint Planning* di mana tim menentukan prioritas pekerjaan dan menetapkan tujuan untuk *sprint* tersebut. Setiap hari, tim melakukan *Daily Stand-up Meeting* untuk memantau kemajuan dan mengatasi hambatan yang mungkin muncul. Pengembangan *backend* dan modul-modul sistem dilakukan secara bertahap, mengikuti prioritas yang telah ditentukan dalam *Sprint Planning*.

5. Pengujian

Setelah setiap *sprint* selesai, sistem atau modul yang dikembangkan harus melalui fase pengujian. Pengujian dilakukan untuk memastikan bahwa sistem berfungsi sesuai dengan spesifikasi yang telah ditetapkan. Terdapat beberapa jenis pengujian yang dilakukan, yaitu unit testing untuk memeriksa setiap komponen secara individu, integration testing untuk memastikan bahwa modul-modul sistem dapat berinteraksi dengan baik, serta *User Acceptance Testing* (UAT) yang melibatkan pengguna akhir untuk memvalidasi apakah sistem memenuhi kebutuhan mereka.

6. Evaluasi dan perbaikan

Pada akhir setiap *sprint*, diadakan *Sprint Review* untuk mengevaluasi hasil kerja dan mempresentasikannya kepada pemangku kepentingan, termasuk pembimbing magang. Dalam *Sprint Review*, tim menerima umpan balik dan saran untuk perbaikan. Selain itu, diadakan *Sprint Retrospective*, di mana tim mengevaluasi proses kerja selama *sprint* dan mencari *query* untuk meningkatkan produktivitas dan efektivitas di *sprint* berikutnya. Proses evaluasi ini penting untuk memastikan bahwa tim terus meningkatkan kualitas kerja dan pengembangan sistem berjalan lebih baik dari waktu ke waktu.

7. Implementasi

Setelah pengujian selesai dan sistem dinyatakan siap, tahap implementasi dan deployment dilakukan. Sistem *backend* yang telah dikembangkan di-deploy di lingkungan produksi, di mana sistem mulai digunakan secara nyata oleh divisi operasional PT PLN UP3 Yogyakarta. Proses ini mencakup instalasi, konfigurasi, serta penyesuaian sistem sesuai dengan infrastruktur teknologi informasi yang ada. Selain itu, pengguna sistem juga mendapatkan pelatihan agar mereka dapat menggunakan sistem dengan efektif dan memahami fungsi-fungsi yang ada.

3.2 Struktur tim *scrum*

Manajemen proyek di PT PLN UP3 Yogyakarta menggunakan metode *Scrum* untuk pengembangan sistem bersama. Struktur tim *Scrum* terdiri dari beberapa peran penting, yang masing-masing memiliki tanggung jawab berbeda dalam setiap tahapan pengembangan. struktur tim *Scrum* di PT PLN UP3 Yogyakarta dapat dilihat dalam tabel 3.1

Table 3.1 Struktur tim *scrum*

Nama	Peran	Tanggung Jawab
Ryan Permana	Product Owner	Mengelola product backlog, memprioritaskan fitur, dan memastikan produk sesuai dengan kebutuhan bisnis.
Ikhwan Cahyono	<i>Scrum</i> Master	Memastikan pelaksanaan proses <i>Scrum</i> berjalan lancar, memfasilitasi tim, dan mengatasi hambatan yang muncul selama pengembangan.
Hestri Widowati Annisa Dinda Irfan Rafid Risdianto Sandi Wijaya	Development Team	Bertanggung jawab untuk mengembangkan produk sesuai dengan <i>sprint</i> backlog dan menyelesaikan tugas-tugas teknis yang telah ditentukan.

Peran-peran ini berkolaborasi untuk memastikan pengembangan sistem berjalan efisien dan sesuai dengan tujuan yang telah ditetapkan.

3.3 Manajemen Proyek

Di PT PLN UP3 Yogyakarta, pengembangan sistem dilakukan dengan menggunakan kerangka kerja *Scrum*. Proses dimulai dengan menyusun *product backlog*, yang memuat daftar fitur-fitur yang telah diprioritaskan oleh *Product Owner* (PO). Kemudian, tim *Scrum* melakukan *Sprint Planning* untuk menetapkan target dan batasan yang akan dicapai selama *sprint* berlangsung. Hasil dari perencanaan ini adalah *sprint backlog*, yang berisi tugas-tugas yang akan diselesaikan dalam *sprint* tersebut. Selama *sprint*, tim melakukan *daily scrum* untuk memberikan pembaruan mengenai Progress-Plan-Problem (P3), yakni laporan harian tentang perkembangan, rencana, dan masalah yang dihadapi. Proses ini diakhiri dengan *briefing* setiap hari untuk menjaga komunikasi dan memastikan *sprint* berjalan lancar, seperti ditampilkan pada Gambar 3.2.



Gambar 3.2 Dokumentasi *briefing* dengan *Project Owner*(PO)

Setelah *sprint* selesai, kegiatan dilanjutkan dengan *sprint review* untuk mengevaluasi apakah tujuan *sprint* telah tercapai dan apakah hasil yang diperoleh sudah memenuhi *definition of done* (DoD). Evaluasi ini dilakukan oleh *Product Owner* bersama dengan seluruh Tim *Scrum* dan pemangku kepentingan. Jika hasil *sprint* memenuhi DoD, maka hasil tersebut dianggap sebagai *increment* atau peningkatan dari produk.

Setelah itu, diadakan pertemuan *sprint retrospective*, yang membahas pelaksanaan *sprint* yang baru saja selesai. Tujuan dari pertemuan ini adalah untuk mengevaluasi proses yang telah berlangsung, mengidentifikasi kekurangan, dan menemukan *query* untuk meningkatkan efisiensi pada *sprint* berikutnya. *Scrum Master* bersama Development Team berpartisipasi dalam *retrospective* ini.

Kegiatan *Scrum* ini dimulai dari *sprint planning*, *daily scrum*, hingga *sprint review* dan *sprint retrospective*. Semua proses ini dilakukan secara berurutan hingga seluruh pengembangan selesai. Setiap iterasi *sprint* yang mencapai DoD menjadi *increment* yang menambah nilai produk secara bertahap.

3.4 Pengembangan Sistem Informasi Bersama (SiBer)

3.4.1 Pendefinisian Proyek

Di PT PLN UP3 Yogyakarta, alur proses proyek dimulai dengan setiap divisi mengimpor data melalui *dashboard* yang disesuaikan dengan kebutuhan masing-masing divisi. *Dashboard* ini memungkinkan setiap divisi untuk memasukkan data yang relevan ke dalam *database* terpusat, di mana informasi tersebut kemudian diolah dan ditampilkan dalam berbagai modul sesuai kebutuhan. Modul-modul tersebut mencakup peta interaktif untuk memantau lokasi titik koordinat, grafik dan *chart* untuk analisis visual data, *scorecard* untuk pelacakan performa, dan *DataTables* untuk pengelolaan data yang lebih terstruktur.

Proses impor data melalui *dashboard* ini memberi akses langsung kepada masing-masing divisi terhadap informasi yang mereka butuhkan dalam format yang sesuai untuk tujuan analisis, pelaporan, atau pengambilan keputusan. Setiap divisi dapat memantau, memproses, dan mengelola data mereka secara *monitoring*, namun tetap terintegrasi dalam sistem terpusat, yang membantu mempercepat alur kerja dan meningkatkan efisiensi operasional di PT PLN UP3 Yogyakarta.

Dalam Proyek Transaksi Energi, data yang telah diimpor divisualisasikan dalam beberapa modul. Modul-modul tersebut mencakup peta interaktif dengan titik koordinat, grafik dalam bentuk *chart*, *scorecard*, dan *DataTables* yang memiliki fungsi untuk mengedit, menghapus, serta menampilkan titik koordinat di peta. Fitur ekspor ke format PDF, CSV, dan cetak juga disediakan untuk mendukung dokumentasi dan pelaporan yang terstruktur.

Pada Proyek Perencanaan, alur pengelolaan data dimulai dari pengimporan hingga visualisasi di *dashboard* dalam bentuk *chart*. Admin memiliki fleksibilitas untuk memperbarui nilai *chart* melalui menu *dropdown* dan memilih tahun yang ingin ditampilkan, memberikan keleluasaan dalam analisis data tahunan. Fitur ekspor ke PDF, CSV, dan cetak juga disediakan untuk kemudahan pemantauan rencana dan hasil capaian. Proyek ini mendukung manajemen dalam memproyeksikan kebutuhan dan mengevaluasi hasil secara lebih efisien, memastikan alokasi sumber daya secara optimal.

Proyek Konstruksi terdiri dari dua *dashboard* yang berbeda untuk ULP dan UP3. Di *dashboard* ULP, unit layanan pelanggan dapat mengajukan permohonan layanan, seperti pemasangan baru atau penambahan daya, dan memantau status permohonan, apakah sudah dibayar, sedang dalam proses, atau telah aktif. Di sisi lain, *dashboard* UP3 memantau dan memperbarui status permohonan dari setiap ULP secara terpusat. Kedua *dashboard* ini

dilengkapi dengan fitur ekspor ke PDF, CSV, dan cetak, baik untuk satu data maupun keseluruhan data, memastikan fleksibilitas dalam penyimpanan dan pelaporan data. Manfaat dari proyek ini adalah memastikan transparansi serta efisiensi dalam proses pengajuan layanan, yang secara signifikan meningkatkan kepuasan pelanggan dan mempercepat waktu layanan PT PLN UP3 Yogyakarta.

3.4.2 *Inisialisasi Proyek*

Inisialisasi proyek diawali dengan melakukan analisis kebutuhan secara menyeluruh untuk PT PLN UP3 Yogyakarta. Pada tahap ini, setiap kebutuhan yang relevan dengan operasional PT PLN UP3, seperti kebutuhan fungsional, target pengguna, serta integrasi dengan sistem yang sudah ada, diidentifikasi dan didokumentasikan. Analisis kebutuhan ini menjadi dasar dalam menyusun ruang lingkup proyek, menetapkan tujuan pengembangan, serta menyelaraskan ekspektasi antar pemangku kepentingan untuk memastikan hasil akhir yang optimal. Proses pengembangan proyek ini melibatkan tiga peran utama:

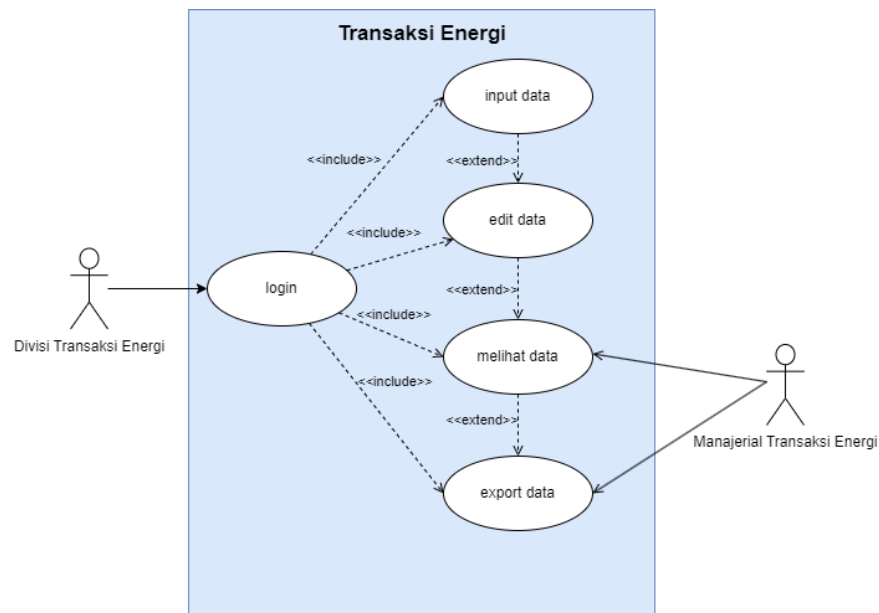
- *Development Team* bertanggung jawab mengembangkan sistem sesuai tugas yang tercantum dalam *sprint backlog*.
- *Scrum Master* bertugas memastikan semua proses *Scrum* berjalan lancar selama pengembangan proyek.
- *Product Owner* menentukan prioritas fitur dan memastikan hasil pengembangan memenuhi standar *definition of done* (DoD).

Pada tahap ini, hasil analisis yang diperoleh dari *Product Owner* bersama *Development Team* digunakan oleh *Development Team* sebagai salah satu acuan dalam perancangan dan pengembangan proyek. Hasil analisis tersebut mencakup prioritas fungsional yang akan diimplementasikan, dan menjadi dasar bagi *Development Team* dalam menetapkan tugas-tugas spesifik pada tiap *sprint*. Sebelum masuk ke tahap pengembangan, terdapat rangkaian aktivitas persiapan. Aktivitas ini mencakup inisiasi proyek, pengumpulan kebutuhan pengguna, pembuatan prototipe, serta perancangan basis data utama.

3.4.3 *Perencanaan Proyek*

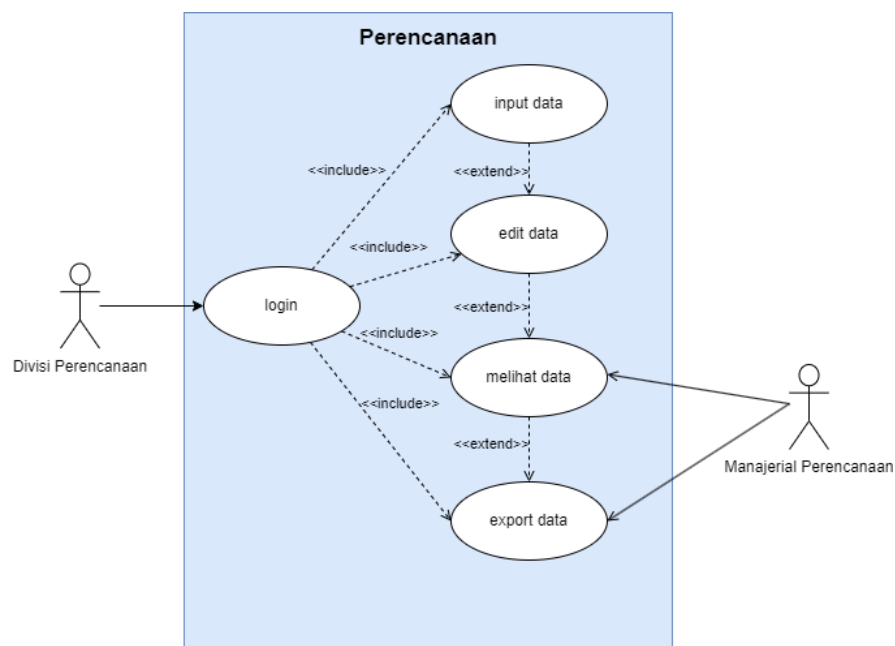
Sistem Informasi Bersama (SIBER) di PT PLN UP3 Yogyakarta dirancang sebagai alat manajemen data yang melibatkan dua aktor utama di setiap divisi, yaitu divisi Transaksi energi dan manajerial transaksi energi pada divisi Transaksi Energi, divisi Perencanaan dan manajerial Perencanaan pada divisi Perencanaan, dan UP3 dan ULP pada Divisi Konstruksi.

Pada Divisi Transaksi Energi, diagram *use case* yang digambarkan pada Gambar 3.6 menunjukkan interaksi antara Admin Divisi Transaksi Energi dan Manajerial Transaksi Energi dalam mengelola, menampilkan, dan mengekspor data transaksi energi menggunakan modul-modul seperti peta, *chart*, dan DataTables.



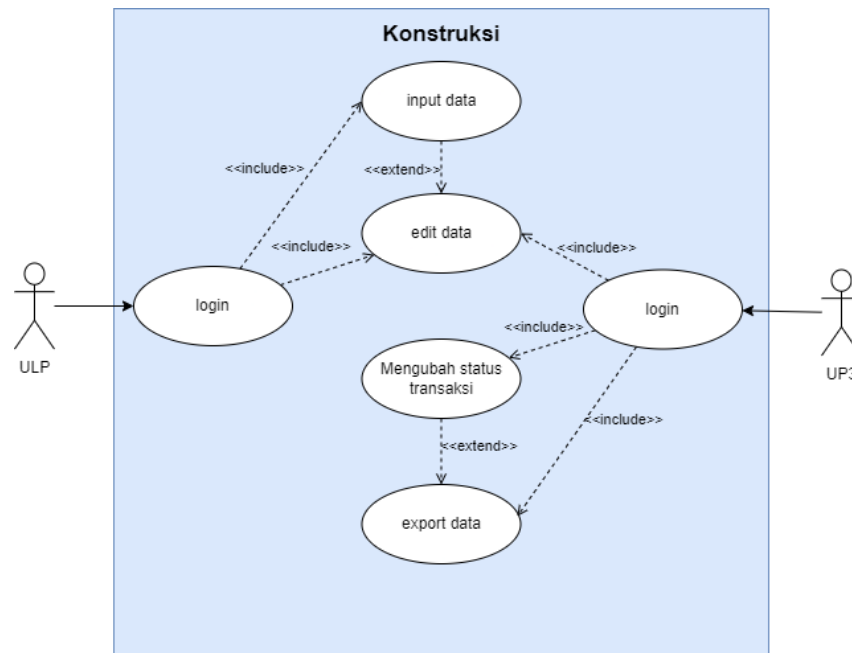
Gambar 3.6 use case diagram Transaksi energi

Sementara itu, Divisi Perencanaan memiliki *use case diagram* yang ditunjukkan pada Gambar 3.7, di mana Admin Divisi Perencanaan dan Manajerial Perencanaan berperan dalam mengimpor data perencanaan, memperbarui *chart*



Gambar 3.7 Use case diagram Perencanaan

Pada Divisi Konstruksi, terdapat dua aktor utama: ULP dan UP3. Diagram *use case* pada Gambar 3.8 menggambarkan akses ULP untuk mengajukan layanan baru dan memantau status permohonan, sementara UP3 berperan dalam memverifikasi dan memantau seluruh permohonan yang diajukan oleh ULP.



Gambar 3.8 Use case diagram Konstruksi

3.4.4 Pelaksanaan Proyek Pra-pembuatan proyek

4. Analisis Kebutuhan Sistem Informasi Bersama (SIBER)

Analisis kebutuhan untuk Sistem Informasi Bersama (SIBER) di PT PLN UP3 Yogyakarta dilakukan untuk memahami kebutuhan operasional pada setiap divisi yang akan menggunakan sistem ini, yaitu Transaksi Energi, Perencanaan, dan Konstruksi. Sistem ini dirancang untuk mendukung efisiensi dalam pengelolaan data serta membantu pengambilan keputusan secara *real-time*. Berikut adalah hasil analisis kebutuhan berdasarkan fungsi utama di tiap divisi:

1. Divisi Transaksi Energi

- Manajemen Data Transaksi: Sistem diharapkan mampu mengimpor data transaksi energi ke *database* dan menyajikannya dalam berbagai format visual, seperti peta dengan koordinat, grafik, *scorecard*, dan DataTables.
- Fitur Edit dan Hapus Data: Admin harus dapat memperbarui dan menghapus data transaksi langsung dari *dashboard*.
- Ekspor Data: Sistem perlu menyediakan opsi ekspor data ke format PDF, CSV, dan cetak untuk keperluan dokumentasi dan laporan.
- Pemetaan Lokasi: Admin harus bisa mengakses peta yang menampilkan data transaksi berdasarkan lokasi.

2. Divisi Perencanaan

- Manajemen Data Perencanaan: Sistem harus mendukung impor data

perencanaan dan menampilkannya dalam bentuk grafik di *dashboard*.

- Fitur Pembaruan Grafik: Sistem memerlukan menu dropdown yang memungkinkan admin memperbarui nilai grafik sesuai data terbaru.
- Pemilihan Tahun: Sistem harus menyediakan fitur pemilihan tahun untuk menampilkan grafik sesuai periode data yang diinginkan.
- Ekspor Data: Tersedia fitur ekspor ke format PDF, CSV, dan cetak untuk memudahkan analisis dan pelaporan.

3. Divisi Konstruksi

- *Dashboard* Terpisah untuk ULP dan UP3: Sistem harus menyediakan dua *dashboard*, satu untuk ULP dan satu lagi untuk UP3, sesuai kebutuhan masing-masing unit.
- Formulir Permohonan Layanan: Sistem harus menyediakan formulir untuk pengajuan layanan baru, penambahan daya, dan permintaan lainnya yang dapat diakses oleh ULP.
- Pemantauan Status Permohonan: ULP dapat memantau status permohonan (dibayar, dalam proses, atau aktif) secara *real-time*.
- Verifikasi Permohonan oleh UP3: Admin UP3 dapat mengakses *dashboard* untuk memantau status permohonan dari ULP dan memastikan semua layanan ditindaklanjuti sesuai statusnya.
- Ekspor Data: Fitur ekspor ke PDF, CSV, dan cetak diperlukan pada masing-masing *dashboard* untuk dokumentasi dan arsip.

5. Membuat logika php pengambilan data

Pada tahap ini, dirancang dan dikembangkan logika php pengambilan data untuk Sistem Informasi Bersama (SIBER) di PT PLN UP3 Yogyakarta, yang difokuskan secara khusus pada kebutuhan Divisi Transaksi Energi. Logika php ini menjadi jembatan utama antara *front-end* dan *back-end* sistem, memungkinkan data transaksi energi yang mencapai volume hingga 26,000 data untuk diakses, diperbarui, dan disinkronkan secara efisien dan terstruktur. Dengan logika ini, Divisi Transaksi Energi dapat mengelola data operasionalnya secara cepat dan akurat, mendukung efisiensi dalam pemantauan dan pelaporan.

Langkah-langkah Pembuatan Logika php Pengambilan Data

5.1 Perancangan Struktur Logika

Logika Manajemen Transaksi Energi: logika ini mendukung operasi CRUD (Create, Read, Update, Delete) untuk data transaksi energi, termasuk koordinat lokasi pada peta, yang memudahkan pemetaan dan pengelolaan data dalam jumlah besar. Seperti yang ditampilkan pada Gambar 3.9.

```
$sql = "SELECT t1.ENDPOINT, t1.SATUAN_UP, t1.TARIF, t1.NAMAUP,
t1.location_code AS LOCATION_CODE, t1.READ_DATE,
COALESCE(t2.NAMA_PELANGGAN, 'kosong') as NAMA_PELANGGAN, t1.POWER,
t1.VOLTAGE_L1, t1.VOLTAGE_L2, t1.VOLTAGE_L3,
COALESCE(lokalasi.KOORDINAT_X, 'kosong') as KOORDINAT_X, COALESCE(lokalasi.KOOR
DINAT_Y, 'kosong') as KOORDINAT_Y
FROM datpelkap AS t1
INNER JOIN datpelnam AS t2 ON t1.LOCATION_CODE = t2.LOCATION_CODE INNER
JOIN (SELECT LOCATION_CODE, MAX(read_date) AS max_read_date
FROM datpelkap GROUP BY LOCATION_CODE) AS t3 ON t1.LOCATION_CODE =
t3.LOCATION_CODE AND t1.read_date = t3.max_read_date
LEFT JOIN lokasi ON t1.LOCATION_CODE=lokasi.idpel
ORDER BY t1.read_date DESC";
$result = $conn->query($sql);
```

Gambar 3.9 Potongan kode logika php pengambilan data transaksi energi

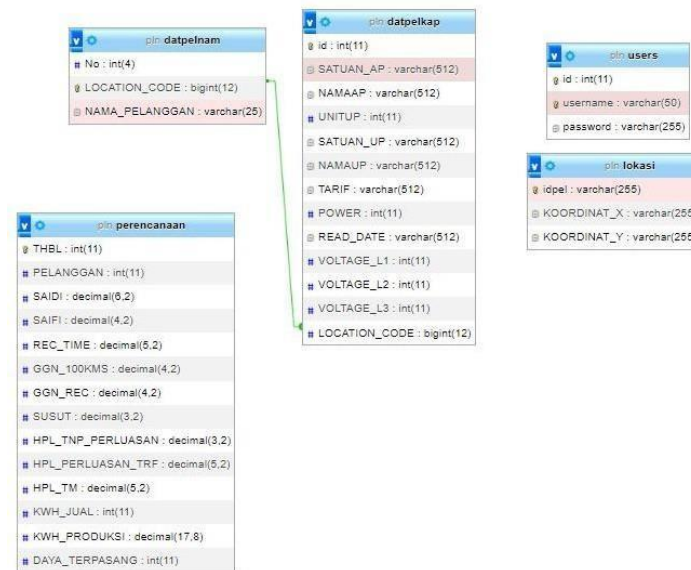
5.2 Pengujian Unit dan Integrasi: Pengujian dilakukan pada logika untuk mengambil data dari basis data guna memastikan logika bekerja sesuai dengan fungsionalitas yang diharapkan, dengan fokus pada kecepatan respon dan integritas data dalam pengelolaan data yang besar.

Dengan logika ini, Sistem Informasi Bersama (SIBER) mendukung aliran data yang cepat dan akurat di Divisi Transaksi Energi PT PLN UP3 Yogyakarta, sehingga meningkatkan efisiensi dalam pengelolaan dan pelaporan data transaksi energi yang besar.

6. Pembuatan basis data menggunakan MySQL

Setelah melalui diskusi tim pengembang di PT PLN UP3 Yogyakarta, MySQL dipilih sebagai RDBMS untuk mendukung sistem yang dibangun. MySQL, sebagai *Database Management System (DBMS) open-source*, menggunakan Structured Query Language (SQL) dan cocok untuk mengelola data terstruktur yang terkait antar-proses bisnis, sesuai kebutuhan di PT PLN UP3. Selain familiar bagi tim, MySQL memungkinkan pemeliharaan yang mudah serta fleksibilitas untuk penambahan dan penyesuaian sistem di masa depan.

Pengembangan sistem basis data MySQL ini berlangsung melalui beberapa iterasi untuk memastikan setiap fitur baru terpenuhi hingga sistem siap. Struktur *database* terdiri dari dua bagian utama: *database* pln dengan 5 tabel, yang mendukung data terkait Transaksi Energi dan Perencanaan sebagaimana ditunjukkan pada Gambar 3.10 dan Tabel 3.2 dan *database* simantri dengan 8 tabel, yang mendukung kebutuhan Konstruksi diperlihatkan pada Gambar 3.11 dan Tabel 3.3. Setiap tabel dalam kedua *database* ini dirancang untuk menyokong fitur sistem masing-masing secara optimal.



Gambar 3.10 Basis data pln

Table 3.2 Tabel fitur dan tabel yang digunakan pada basis data PLN

No	Fitur	Tabel yang dipakai
1	Login	users : tabel yang berisi data user
2	<i>Dashboard</i> Perencanaan	Perencanaan : tabel yang berisi data bulanan dari divisi perencanaan
3	<i>Dashboard</i> Transaksi Energi	Datpelnam : tabel yang berisi nama pelanggan Datpelkap : tabel yang berisi data lengkap pelanggan
4	Maps pada <i>dashboard</i> transaksi energi	Lokasi : tabel yang berisi titik koordinat dari pelanggan



Gambar 3.11 Basis data simantri

Table 3.3 Tabel fitur dan tabel yang digunakan pada basis data Simantri

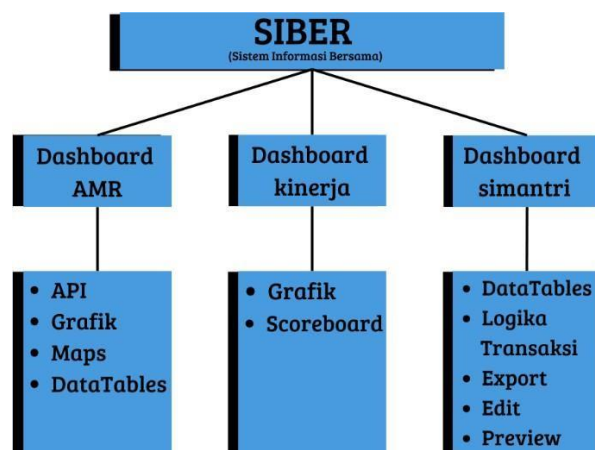
No	Fitur	Tabel yang dipakai
1	Login ULP dan UP3	user : tabel yang berisi data user
2	<i>Dashboard ULP</i>	<p>Permohonan: tabel yang berisi permohonan pelanggan</p> <p>Kode unit: tabel yang berisi kode untuk masing-masing ulp</p> <p>Kode tarif: tabel yang berisi kode dari tarif-tarif yang ada di pln</p> <p>Kode status: tabel yang berisi kode status pembayaran</p> <p>Kode status progress: tabel yang berisi kode status progress</p> <p>Kode tegangan: tabel yang berisi kode tegangan dari pelanggan</p> <p>Kode jenis transaksi: tabel yang berisi kode dari jenis transaksi pelanggan</p>

3	<i>Dashboard UP3</i>	Permohonan: tabel yang berisi permohonan pelanggan Kode unit: tabel yang berisi kode untuk masing-masing ulp Kode tarif: tabel yang berisi kode dari tarif-tarif yang ada di pln Kode status: tabel yang berisi kode status pembayaran Kode status progress: tabel yang berisi kode status
---	----------------------	---

No	Fitur	Tabel yang dipakai
		progress Kode tegangan: tabel yang berisi kode tegangan dari pelanggan Kode jenis transaksi: tabel yang berisi kode dari jenis transaksi pelanggan

Pembuatan proyek

Dalam tahap pembuatan proyek Sistem Informasi Bersama (SIBER) untuk PT PLN UP3 Yogyakarta, berbagai elemen utama dikembangkan untuk mendukung sistem dalam menyajikan data secara informatif dan interaktif kepada pengguna. SIBER terdiri dari tiga *dashboard* utama *Dashboard AMR*, *Dashboard Kinerja*, dan *Dashboard Simantri* seperti yang terlihat pada gambar 3.12 yang masing-masing dirancang untuk memenuhi kebutuhan spesifik dari tiga divisi, yaitu Transaksi Energi, Perencanaan, dan Konstruksi. Setiap elemen sistem dirancang dengan modul dan teknologi yang sesuai untuk memastikan kelancaran pengolahan data, integrasi antar-modul, serta kemudahan pengguna dalam berinteraksi dengan sistem.



Gambar 3.12 Proyek Sistem Informasi Bersama

Pembuatan proyek dilakukan dengan pendekatan *Scrum* yang membagi proses pengembangan ke dalam *sprint-sprint* terstruktur. Dengan durasi proyek enam bulan, tiap divisi mendapatkan dua bulan untuk pengembangan fitur-fiturnya. Setiap *sprint* melibatkan proses analisis kebutuhan, desain, implementasi, pengujian, dan evaluasi, disertai *sprint review* dan *sprint retrospective* yang dilakukan setiap dua minggu.

Dashboard AMR

Dashboard AMR adalah sebuah antarmuka visual yang dirancang untuk memantau data yang di-input oleh Divisi Transaksi Energi melalui sistem *Automatic Meter Reading* (AMR). *Dashboard* ini berfungsi untuk mengelompokkan data pelanggan ke dalam tiga status utama, yaitu *normal*, *butuh perhatian*, dan *butuh perbaikan*. Pada pengembangannya, beberapa fitur utama yang dikembangkan meliputi logika php pengambilan data, grafik, peta (maps), dan DataTables, yang secara kolektif berfungsi untuk menyajikan data secara informatif dan interaktif kepada pengguna.

Sprint 1 : Logika php pengambilan data

Pada *sprint* pertama, pengembangan difokuskan pada pembuatan logika php sebagai sumber data utama untuk kebutuhan visualisasi di frontend. Tujuan utama logika php ini adalah menyediakan data pelanggan dalam format JSON, sehingga dapat diakses dengan mudah oleh komponen lain di sistem. Data yang dikelola oleh logika ini meliputi informasi pelanggan, seperti ID Pelanggan, Nama, Unit, Daya, Tanggal Pencatatan (*Read Date*), Tegangan (*Voltage*) yang terbagi dalam tiga parameter (*Voltage L1*, *Voltage L2*, *Voltage L3*). Proses pengembangan logika ini dimulai dengan merancang struktur data pada *database* menggunakan MySQL. Data pelanggan diambil melalui penggabungan beberapa tabel dengan menggunakan *query SQL join*, yang memastikan data terbaru dari setiap pelanggan dapat diakses. Untuk melengkapi informasi, seperti koordinat pelanggan, *query SQL* juga mengintegrasikan data dari tabel lain yang relevan. Setelah *query* selesai dirancang, logika pengolahan data diimplementasikan dalam file *getdata.php*. File ini bertanggung jawab untuk mengambil data dari *database*, mengolahnya, dan menyajikannya dalam format JSON. Berikut adalah potongan kode yang digunakan dalam proses pengambilan data:

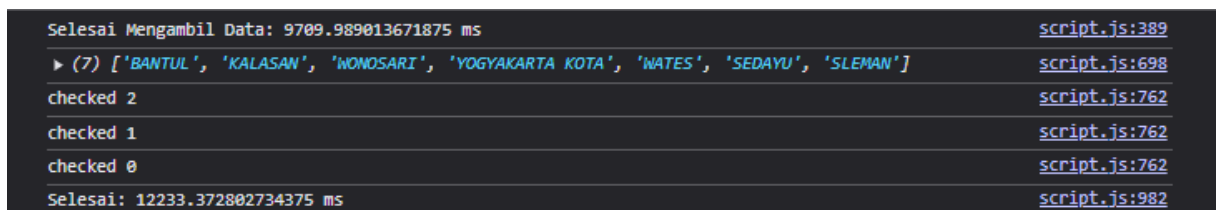
```

<?php
require '../config/koneksi.php';
$data = [];
$sql = "SELECT t1.ENDPOINT, t1.SATUAN_UP, t1.TARIF, t1.NAMAUP, t1.location_code
AS LOCATION_CODE, t1.READ_DATE,
COALESCE(t2.NAMA_PELANGGAN, 'kosong') as NAMA_PELANGGAN, t1.POWER, t1.VOLTAGE_L1,
t1.VOLTAGE_L2, t1.VOLTAGE_L3, COALESCE(lokas.kOORDINAT_X, 'kosong') as
KOORDINAT_X, COALESCE(lokas.kOORDINAT_Y, 'kosong') as KOORDINAT_Y
FROM datpelkap AS t1
INNER JOIN datpelnam AS t2 ON t1.LOCATION_CODE = t2.LOCATION_CODE
INNER JOIN (
    SELECT LOCATION_CODE, MAX(read_date) AS max_read_date
    FROM datpelkap
    GROUP BY LOCATION_CODE
) AS t3 ON t1.LOCATION_CODE = t3.LOCATION_CODE AND t1.read_date =
t3.max_read_date
LEFT JOIN lokasi ON t1.LOCATION_CODE=lokasi.idpel
ORDER BY t1.read_date DESC";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        array_push($data, $row); }
}
print_r(json_encode($data));

```

Gambar 3.13 Potongan kode logika get data

Pada tahap pengujian, logika php yang dikembangkan berhasil menyediakan data pelanggan dari sistem AMR dengan cepat dan akurat. Data yang diambil meliputi informasi seperti nama pelanggan, daya, tegangan, lokasi yang disajikan dalam format JSON. Hasil pengujian ditampilkan melalui konsol *browser* untuk memastikan bahwa data dapat diakses secara dinamis dan terstruktur. Tampilan penggunaan logika php pada konsol browser dapat dilihat pada gambar 3.14.



```

Selesai Mengambil Data: 9709.989013671875 ms script.js:389
> (7) ['BANTUL', 'KALASAN', 'NONGSARI', 'YOGYAKARTA KOTA', 'WATES', 'SEDAYU', 'SLEMAN'] script.js:698
checked 2 script.js:762
checked 1 script.js:762
checked 0 script.js:762
Selesai: 12233.372802734375 ms script.js:982

```

Gambar 3.14 Tampilan penggunaan logika php pada konsol browser

Hasil dari *sprint* pertama ini menjadi fondasi utama bagi pengembangan fitur visualisasi pada *sprint* berikutnya, seperti grafik, peta, dan tabel interaktif. Pengujian ini memastikan kejelasan dan efisiensi dalam pengolahan data, sehingga mendukung integrasi yang optimal antara *backend* dan *frontend*

Sprint 2 : Grafik

Pada *sprint* kedua, fokus pengembangan adalah visualisasi data dalam bentuk pie *chart* yang dapat memberikan gambaran distribusi status pelanggan (normal, butuh perhatian, dan butuh perbaikan). Grafik ini dirancang untuk memberikan informasi yang jelas dan interaktif, memungkinkan pengguna untuk memahami data pelanggan secara lebih mendalam.

Fungsi `updateChart()` digunakan sebagai inti logika untuk memproses data dari logika php pengambilan data dan mengatur struktur data yang akan digunakan oleh grafik. kode lengkap fungsi `updateChart()` dapat dilihat dalam gambar 3.15.

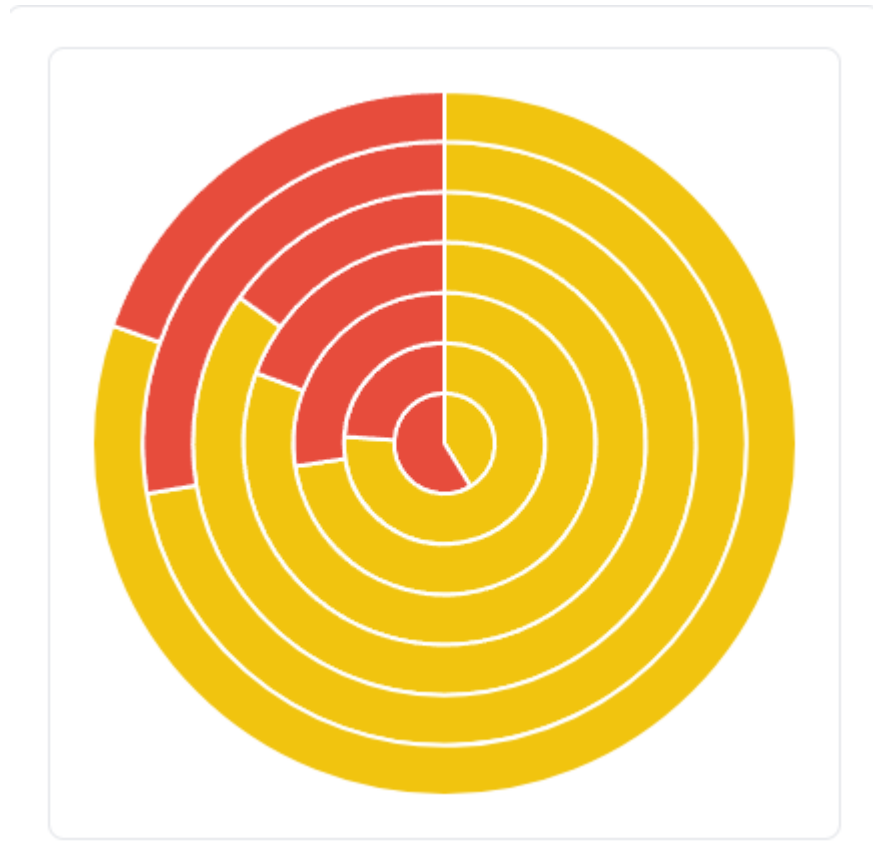
```
function updateChart(isUpdate = false) {
  let dataNormal = {}
  let dataButuhPerhatian = {}
  let dataButuhPerbaikan = {}
  let temporaryData = data.filter(d => { const lokasi = d.nama.replace(" ", "")
  let tegangan = d.tegangan let status = d.status return
  !hiddenLokasi.includes(lokasi) && !hiddenStatus.includes(status.toString()) &&
  !hiddenTegangan.includes(tegangan) })
```

Gambar 3.15 Potongan kode `updateChart()`

Fungsi ini memiliki beberapa langkah utama:

1. **Filter Data:** Data yang diambil dari logika php pengambilan data diproses berdasarkan filter tertentu, seperti lokasi, tegangan, dan status yang tidak termasuk dalam daftar *hidden*.
2. **Pemrosesan Lokasi:** Fungsi memetakan data pelanggan berdasarkan lokasi, memastikan setiap lokasi hanya muncul sekali dalam grafik.
3. **Klasifikasi Status:** Setiap lokasi memiliki tiga kategori data: *normal*, *butuh perhatian*, dan *butuh perbaikan*, yang dihitung berdasarkan status pelanggan.
4. **Struktur Data untuk Chart.js:** Data yang telah diproses diubah menjadi struktur yang sesuai untuk digunakan oleh *library Chart.js*, termasuk label dan warna setiap kategori.
5. **Pembaruan Dinamis:** Jika `isUpdate` diset sebagai `true`, grafik yang sudah dirender akan diperbarui dengan data baru tanpa perlu memuat ulang halaman.

Fungsi ini berhasil memastikan bahwa grafik pie *chart* dapat memvisualisasikan data pelanggan dengan akurat dan sesuai kategori. Hasil dari fungsi ini digunakan oleh kode grafik untuk merender grafik interaktif, seperti yang ditunjukkan pada gambar 3.16.



Gambar 3.16 Tampilan dari grafik pie *chart* dengan layer

Pada tahap pengujian, fungsi ini berhasil memproses data dari logika php dengan efisien, bahkan jumlah data yang dikelola cukup besar. Fungsi ini juga memungkinkan pengaturan ulang data secara dinamis berdasarkan filter, menjadikannya sangat fleksibel untuk kebutuhan pengguna. Grafik pie *chart* yang dihasilkan memberikan informasi mendalam yang dapat mendukung pengambilan keputusan berbasis data bagi divisi transaksi energi.

Sprint 3: Peta (Maps)

Pada *sprint* ketiga, fokus pengembangan adalah visualisasi data dalam bentuk peta interaktif yang menampilkan lokasi pelanggan berdasarkan data yang diambil dari logika php. Visualisasi peta ini membantu pengguna untuk memahami sebaran pelanggan, termasuk status setiap lokasi (normal, butuh perhatian, dan butuh perbaikan). Fitur ini memberikan gambaran geografis yang relevan untuk mendukung analisis dan pengambilan keputusan.

Kode Utama Peta (Maps)

Kode implementasi utama untuk peta dapat dilihat dalam gambar 3.17, menggunakan *library* seperti Leaflet.js untuk memetakan data pelanggan:

```

$("#mapContainer").show();
map = L.map("map").setView([-7.797068, 110.370529],13);
L.tileLayer("https://{s}.google.com/vt/lyrs=m&x={x}&y={y}&z={z}", {
    subdomains: ["mt0", "mt1", "mt2", "mt3"], attribution:
    '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
    contributors',
}).addTo(map);
let listMap = data.map((d) => d.nama);
listMap = listMap.filter(function (elem, pos) {
    return listMap.indexOf(elem) == pos; });

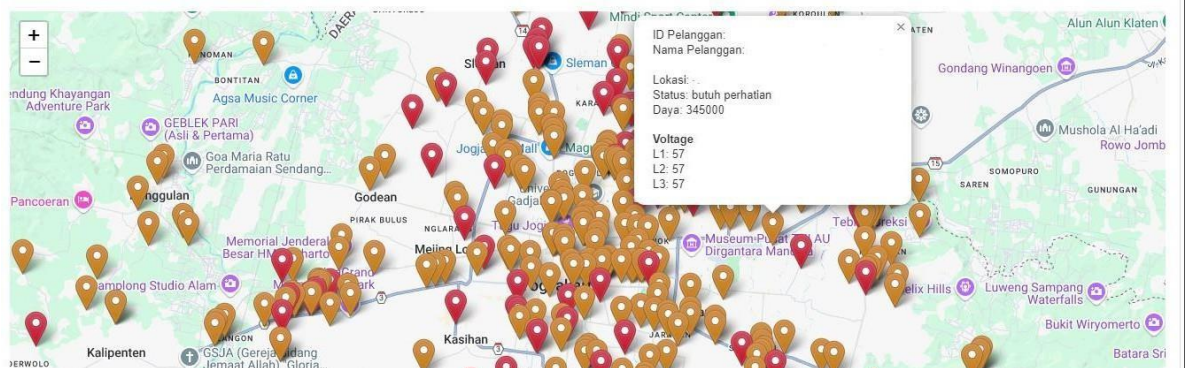
```

Gambar 3.17 Potongan kode maps menggunakan leaflet.js

Penjelasan Kode:

1. Inisialisasi Peta: Peta dimulai dengan koordinat pusat di wilayah Yogyakarta, dengan *zoom level* yang dapat diatur sesuai kebutuhan.
2. Tile Layer: Menggunakan OpenStreetMap untuk menampilkan peta dasar.
3. Filter Data: Data pelanggan dari logika php difilter berdasarkan lokasi dan status untuk memastikan hanya data yang relevan ditampilkan.
4. Warna Marker: Marker pada peta diberi warna sesuai dengan status pelanggan: hijau untuk *normal*, kuning untuk *butuh perhatian*, dan merah untuk *butuh perbaikan*.
5. Pop-up Informasi: Setiap marker dilengkapi dengan pop-up yang menampilkan informasi detail pelanggan, seperti nama, status, dan tegangan.

Hasil dari fungsi ini ditampilkan dalam peta interaktif, seperti pada gambar berikut:



Gambar 3.18 visualisasi maps

Pada tahap pengujian, peta berhasil menampilkan lokasi pelanggan dengan akurat berdasarkan koordinat yang diambil dari logika php. Warna marker membantu pengguna untuk dengan cepat mengenali status pelanggan di setiap lokasi. Selain itu, fungsi pop-up memberikan informasi detail secara langsung, membuat peta menjadi lebih informatif dan interaktif.

Peta ini memberikan nilai tambah dalam visualisasi data pelanggan, membantu divisi transaksi energi untuk memahami distribusi status pelanggan berdasarkan lokasi geografis.

Sprint 4: DataTables

Pada *sprint* keempat, fokus pengembangan adalah implementasi tabel data interaktif menggunakan *library* DataTables. Fitur ini dirancang untuk memungkinkan pengguna mengelola dan memfilter data pelanggan dengan mudah melalui antarmuka yang responsif. Tabel ini mendukung fungsi pencarian, pemilihan kolom yang ditampilkan, serta ekspor data ke berbagai format seperti CSV, PDF, dan salinan clipboard.

Kode Utama DataTables

Kode implementasi utama untuk DataTables yang digunakan pada sistem dapat dilihat dalam gambar 3.19:

```
table = $("#example1").DataTable({
  columnDefs: [
    {target: 3, visible: true},
    {target: 4, visible: true},
    {target: 5, visible: true},
    {target: 6, visible: true },
    {target: 7, visible: true}],
  initComplete: function () {
    this.api().columns().every(function () {
      let column = this;
      let title = column.footer().textContent;...
```

Gambar 3.19 Potongan kode untuk datatables

Penjelasan Kode lebih rinci:

1. Pencarian Kolom:
 - Setiap kolom tabel memiliki input pencarian khusus untuk memfilter data secara langsung.
2. Tombol Ekspor:
 - Copy: Menyalin data ke clipboard.
 - CSV: Mengunduh data dalam format CSV.
 - PDF: Mengunduh data dalam format PDF dengan pewarnaan sesuai status.
3. Filter dan Visibilitas Kolom:
 - Tombol *ColVis* memungkinkan pengguna memilih kolom yang ingin ditampilkan.
4. Tombol Aksi Tambahan:
 - *Import*: Membuka modal untuk mengimpor data.
 - *Delete*: Menghapus data melalui konfirmasi.
5. Responsivitas:
 - Tabel ini dirancang agar tetap dapat digunakan dengan baik pada berbagai ukuran layar.

○

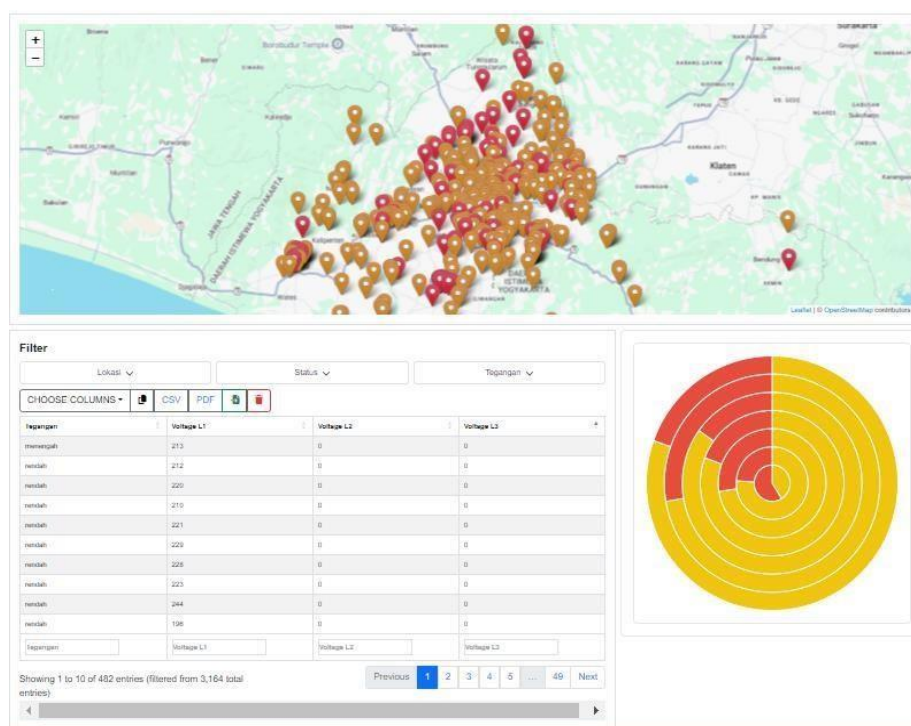
Pada tahap pengujian, DataTables berhasil memuat dan memproses data dengan performa yang optimal, termasuk fungsi pencarian, ekspor, dan pengaturan visibilitas kolom. Fitur ini memberikan fleksibilitas kepada pengguna untuk mengelola data dengan lebih efisien. Hal ini mendukung tujuan utama *sprint*, yaitu menyediakan antarmuka data yang mudah digunakan dan interaktif.

Hasil implementasi DataTables pada sistem dapat dilihat pada gambar 3.20, yang menampilkan fitur pencarian kolom, pemilihan kolom, serta tombol ekspor data :

Lokasi ▾		Status ▾		Tegangan ▾	
CHOOSE COLUMNS ▾		CSV	PDF		
Read Date	Voltage L1	Voltage L2	Voltage L3	Status	
8/8/23 9:59	213	0	0		Butuh Perbaikan
8/8/23 9:09	212	0	0		Butuh Perbaikan
8/8/23 9:01	220	0	0		Butuh Perbaikan
8/8/23 8:59	210	0	0		Butuh Perbaikan
8/8/23 8:58	221	0	0		Butuh Perbaikan
8/8/23 8:58	229	0	0		Butuh Perbaikan
8/8/23 8:58	228	0	0		Butuh Perbaikan
8/8/23 8:57	223	0	0		Butuh Perbaikan
8/8/23 8:56	244	0	0		Butuh Perbaikan
8/8/23 8:55	196	0	0		Butuh Perbaikan
Read Date	Voltage L1	Voltage L2	Voltage L3	Status	

Gambar 3.20 Tampilan datatables

hasil akhir dari seluruh *dashboard* yang menggunakan *backend* diatas dapat dilihat dalam gambar 3.21.



Gambar 3.21 hasil akhir *dashboard* AMR

Dashboard Kinerja

Dashboard Kinerja adalah antarmuka visual yang digunakan untuk memantau dan mengevaluasi indikator-indikator kinerja utama (KPI) dari PT.PLN UP3 Yogyakarta. *Dashboard* ini mengelompokkan data menjadi beberapa kategori utama, yang mencakup jumlah pelanggan, daya terpasang, kWh produksi, kWh jual, susut SAIDI, SAIFI, *recovery time*, gangguan 100 kms, gangguan recloser, hpl tanpa perluasan jaringan, hpl dengan perluasan jaringan, dan hpl tegangan menengah

Dalam *Dashboard* Kinerja, pengguna dapat memilih tahun untuk menampilkan data kinerja pada *Scoreboard* dan Grafik. *Dashboard* ini bertujuan untuk memberikan pemahaman yang jelas dan visual tentang kinerja PT. PLN UP3 Yogyakarta berdasarkan data yang dimasukkan oleh divisi perencanaan.

Sprint 1: Scoreboard

Pada *Sprint 1*, pengembangan difokuskan pada fitur *Scoreboard*, yang menjadi salah satu menu utama dalam *Dashboard* Kinerja. *Scoreboard* dirancang untuk menyajikan berbagai indikator kinerja utama, seperti:

- Jumlah Pelanggan
- Daya Terpasang
- kWh Produksi
- kWh Jual
- Susut
- SAIDI (*System Average Interruption Duration Index*)
- SAIFI (*System Average Interruption Frequency Index*)
- Recovery Time
- Gangguan 100 km
- Gangguan Recloser
- HPL Tanpa Perluasan Jaringan
- HPL dengan Perluasan Jaringan
- HPL Tegangan Menengah

Setiap indikator dipisahkan berdasarkan kategori menu sesuai kebutuhan, dan data ditampilkan secara dinamis berdasarkan tahun yang dipilih oleh pengguna. Fitur ini memungkinkan pengguna untuk melihat pencapaian kinerja secara ringkas dan akurat melalui data yang relevan dan terkini. Contoh potongan kode salah satu *scoreboard* yang terdapat pada *dashboard* kinerja dapat dilihat pada gambar 3.22.

```
function scoreBoard() {
  const tahunTerpilih = document.getElementById('selectYear1').value;
  const dataTerpilih = mainData.filter(data => data.thbl.startsWith(tahunTerpilih))
    .sort((a, b) => a.thbl.localeCompare(b.thbl)).reverse()[0] || null;
  console.log(dataTerpilih);
  if (dataTerpilih) {
    document.getElementById('scoreBoardJumlahPelanggan').innerHTML =
      dataTerpilih.PELANGGAN || 0;
    document.getElementById('scoreBoardDaya').innerHTML = dataTerpilih.DAYA_TERPASANG
      || 0;}}

```

Gambar 3.22 potongan kode salah satu *scoreboard*

Pada tahap pengujian, *Scoreboard* berhasil menampilkan data indikator kinerja seperti jumlah pelanggan, daya terpasang, kWh produksi, kWh jual, susut, SAIDI, SAIFI, recovery time, gangguan 100 km, gangguan recloser, HPL tanpa perluasan jaringan, HPL dengan perluasan jaringan, dan HPL tegangan menengah. Data disajikan secara dinamis sesuai tahun yang dipilih pengguna dan otomatis pada data bulan yang paling terakhir. Fitur ini memberikan gambaran ringkas dan akurat, mendukung fleksibilitas pengguna dalam memantau indikator kinerja. Hasil pengujian menunjukkan bahwa *Scoreboard* berfungsi dengan optimal, termasuk kemampuan untuk memperbarui data secara *real-time* tanpa memerlukan pemuatan ulang halaman. Contoh hasil *scoreboard* dapat dilihat pada gambar 3.23



Gambar 3.23 contoh scoreboard pada menu pelanggan dan daya

Sprint 2: Grafik

Pada *Sprint 2*, pengembangan difokuskan pada pembuatan berbagai grafik untuk memvisualisasikan data kinerja secara informatif dan interaktif. Grafik-grafik ini dirancang untuk memberikan wawasan mendalam mengenai indikator kinerja utama berdasarkan data yang diperoleh selama periode satu tahun dan setiap bulan.

Berikut adalah jenis grafik yang dikembangkan dalam *Sprint 2*:

1. Pelanggan Daya:
 - *Bar Chart*: Menampilkan jumlah pelanggan berdasarkan kategori tertentu untuk memberikan gambaran distribusi pelanggan.
 - *Line Chart*: Menunjukkan tren daya terpasang dari waktu ke waktu untuk memantau perkembangan kapasitas daya.
2. KWH Produksi, KWH Jual, dan Susut:
 - *Line Chart*: Menyajikan data kWh produksi, kWh jual, dan tingkat susut secara bersamaan untuk memvisualisasikan efisiensi dan perbandingan antara produksi dan distribusi energi.

3. SAIDI, SAIFI, Recovery Time, Gangguan 100 KM, dan Gangguan Recloser:

- *Line Chart*: Memonitor indikator performa jaringan listrik dengan menampilkan tren data gangguan, waktu pemulihan, dan indeks SAIDI-SAIFI dalam periode tertentu.

4. HPL (Hari Pelanggan Lama):

- *Line Chart*:
 - HPL Tanpa Perluasan Jaringan: Menggambarkan durasi rata-rata penyelesaian pelanggan tanpa perluasan jaringan.
 - HPL Dengan Perluasan Jaringan: Menunjukkan durasi rata-rata penyelesaian pelanggan dengan proyek perluasan jaringan.
 - HPL Tegangan Menengah: Menyajikan data durasi penyelesaian untuk pelanggan tegangan menengah.

Setiap grafik menggunakan library *Chart.js*, yang mendukung pembaruan data secara dinamis berdasarkan tahun atau filter tertentu yang dipilih oleh pengguna. Grafik ini memudahkan pengguna dalam memahami performa operasional melalui visualisasi yang jelas dan intuitif. Salah satu potongan kode contoh grafik dari daya terpasang dapat dilihat pada gambar 3.24.

```
function updateLineChart(isUpdate = false) {
const filteredData = mainData.filter(data => data.thbl.substring(0,4) == selectedYear1)
var powerValues2 = filteredData.map((data) => (data.DAYA_TERPASANG))
var labels = filteredData.map(function(item) {
return bulan_indonesia[parseInt(item.thbl.toString().replace(selectedYear1,"")) - 1]});

var datasets = [{
  label: 'Daya',
  data: filteredData.map(data => data.DAYA_TERPASANG),
  backgroundColor: 'rgb(217,61,70)',
  borderColor: 'rgb(217,61,70)',
  borderWidth: 1,},...]
```

Gambar 3.24 contoh potongan kode grafik daya terpasang

Pada tahap pengujian, grafik-grafik yang dikembangkan berhasil memvisualisasikan data secara akurat dan memberikan fleksibilitas dalam menampilkan berbagai indikator kinerja berdasarkan pilihan pengguna. Grafik ini juga memastikan selalu menampilkan data terbaru secara *real-time*. Contoh hasil grafik yang dihasilkan dapat dilihat pada gambar 3.25.



Gambar 3.25 Contoh grafik yang berada didashboard kinerja

Dashboard SIMANTRI

Dashboard SIMANTRI adalah sebuah platform yang digunakan untuk mengelola transaksi pelanggan terkait pasang baru, tambah daya, dan keperluan lainnya dalam layanan PLN UP3 Yogyakarta. *Dashboard* ini memfasilitasi koordinasi antara Unit Layanan Pelanggan (ULP) dan UP3 Yogyakarta dalam memastikan setiap transaksi pelanggan berjalan dengan lancar, mulai dari pembayaran hingga status akhir transaksi (penyalan).

Proses transaksi yang dikelola dalam *Dashboard SIMANTRI* melibatkan langkah-langkah berikut:

1. Unit Layanan Pelanggan (ULP) menerima transaksi pelanggan.
2. Transaksi diteruskan ke UP3 Yogyakarta untuk pengecekan pembayaran pelanggan.
3. Jika pembayaran telah dilakukan, UP3 Yogyakarta membuat Surat Perintah Kerja (SPK) untuk ULP terkait.
4. ULP melaksanakan proses transaksi pelanggan (misalnya pemasangan baru atau penambahan daya).
5. Setelah transaksi selesai, status pelanggan diubah menjadi sudah nyala.

Fitur pada Dashboard SIMANTRI

Dashboard ini memiliki dua antarmuka utama:

1. *Dashboard UP3*
 - Melihat semua transaksi pelanggan.
 - Mengubah status pelanggan: Sudah Bayar, Proses Pengerjaan, hingga Sudah Nyala.
 - input SPK untuk ULP terkait.
2. *Dashboard ULP*
 - Menambahkan data transaksi baru.
 - Memantau proses transaksi pelanggan.

Sprint 1: DataTables

Pada *sprint* pertama, fokus pengembangan diarahkan pada implementasi DataTables untuk menampilkan data transaksi pelanggan secara interaktif. Fitur ini mencakup fungsi pencarian, pagination, pengurutan data, dan pengelolaan tombol aksi.

Logika DataTables

Potongan kode pada gambar 3.26 diimplementasikan untuk menginisialisasi DataTables dengan fitur yang dibutuhkan.

```
$(function() {
  let table;
  table = $('#example1').DataTable({ "responsive": false, "lengthChange": false,
    "autoWidth": false, "select": { "style": 'multi', "info": false }, ...
```

Gambar 3.26 potongan kode datatables

Fitur Utama

1. Export Options:

- Data dapat diekspor ke format CSV, Excel, atau PDF.
- Terdapat pilihan untuk mencetak data langsung dari tabel.

2. Preview Selected:

- Menampilkan tombol *preview* dengan jumlah baris yang dipilih.
- Fitur ini hanya aktif jika terdapat baris yang dipilih.

3. Column Visibility:

- Pengguna dapat mengatur visibilitas kolom tertentu.

Pada tahap pengujian ditemukan bahwa fitur *DataTables* telah diimplementasikan dengan baik dengan rincian sebagai berikut:

- Interaktivitas: Pencarian, pengurutan, dan pagination bekerja secara optimal.
- Ekspor Data: Data berhasil diekspor ke berbagai format sesuai kebutuhan.
- *Preview*: Tombol *preview* dinamis berdasarkan baris yang dipilih.

Fitur ini memberikan pengalaman pengguna yang responsif dan mudah digunakan dalam mengelola data transaksi pelanggan. Hasil dari datatables ini dapat dilihat pada gambar 3.27.

The screenshot shows a web interface titled "Daftar Pemohon". At the top, there are buttons for "Copy to Clipboard", "Export to CSV", "Export to Excel", "Export to PDF", and "Print". A search bar is located on the right. Below these is a "Column Visibility" dropdown and a "0 Preview Selected" indicator. The main part of the interface is a table with the following columns: Id Pemohonan, Unit, Nama Pelanggan, Alamat, ID Pelanggan, Jenis Transaksi, Peruntukan, Tarif Lama, Tarif Baru, Daya Lama, Daya Baru, Jumlah Pelanggan, Tanggal Permohonan, Jenis Pelanggan, and Action. The table body is empty, displaying the message "No data available in table". At the bottom, it says "Showing 0 to 0 of 0 entries" and has "Previous" and "Next" navigation buttons.

Gambar 3.27 hasil datatables pada dashboard UP3

Sprint 2: Logika Transaksi

Pada *sprint* ini, fokus pengembangan adalah memastikan alur transaksi pelanggan berjalan sesuai tahapannya, dengan setiap aksi terpisah melalui beberapa Logika php. Logika transaksi ini memanfaatkan tiga logika utama untuk memperbarui status permohonan pelanggan sesuai dengan tahapan yang sudah ditentukan.

Logika yang Digunakan

1. updatePermohonan.php:

- Fungsi: Mengupdate status permohonan menjadi "Sudah Bayar" dan menambahkan informasi terkait seperti SPK, tanggal bayar, nomor agenda, dan nama vendor.
- Tahap Pemanggilan: Dilakukan ketika pelanggan sudah melakukan pembayaran.

2. updateBayar.php:

- Fungsi: Mengupdate status progress menjadi "Proses" atau tetap di "Sudah Bayar".
- Tahap Pemanggilan: Dilakukan pada tahap konfirmasi progress pekerjaan.

3. updateProses.php:

- Fungsi: Mengupdate status permohonan menjadi "Sudah Nyala", atau tetap di tahap Proses, dengan opsi untuk memperbarui tanggal nyala.
- Tahap Pemanggilan: Dilakukan ketika proses selesai dan layanan sudah menyala.

Logika Perubahan Status

Daftar Permohonan

- Aksi: Tombol "Sudah Bayar".
- Logika: updatePermohonan.php.
- Deskripsi: Mengubah status permohonan menjadi "Sudah Bayar", menambahkan informasi SPK, tanggal bayar, nomor agenda, dan vendor.
- 2. Sudah Bayar:
 - Aksi: Tombol "Proses".
 - Logika: updateBayar.php.
 - Deskripsi: Memberikan opsi "Proses" atau tetap "Sudah Bayar".
- 3. Proses:
 - Aksi: Tombol "Sudah Nyala".
 - Logika: updateProses.php.
- Deskripsi: Memberikan opsi untuk mengubah status menjadi "Sudah Nyala" atau tetap di status "Proses".

Detail Implementasi Logika php

1. updatePermohonan.php

Logika ini digunakan untuk memperbarui status permohonan menjadi "Sudah Bayar" (STATUS_BAYAR = 2) sekaligus menyimpan data tambahan seperti nomor SPK, tanggal SPK, tanggal bayar, nomor agenda, dan nama vendor. Proses ini dapat dilihat pada gambar 3.28 yang menampilkan potongan kode logika php, diikuti oleh gambar 3.29 yang menunjukkan form SPK untuk memasukkan data terkait. Setelah data berhasil disimpan, permohonan akan berpindah ke halaman "Sudah Bayar", seperti yang ditunjukkan pada gambar 3.30.

```
$idPermohonan = $_POST['id_permohonan'];
$no_spk = $_POST['no_spk'] ?? '';
$tanggal_spk = $_POST['tanggal_spk'] ?? '';
$sql = "UPDATE permohonan SET STATUS_BAYAR = '2', TANGGAL_BAYAR = ?, NO_SPK
= ?, TANGGAL_SPK = ?, NO_AGENDA = ?, NAMA_VENDOR = ? WHERE ID_PERMOHONAN =
?"; ...
```

Gambar 3.28 potongan kode logika php updatePermohonan.php

Edit Nomor SPK

Nomor SPK:

Tanggal SPK: 17/12/2024

Tanggal Bayar: 27/12/2024

Nomor Agenda:

Nama Vendor: PLN ULP BANTUL

Submit

Gambar 3.29 Tampilan Form Surat Perintah Kerja (SPK)

Daftar Pelanggan Bayar

Copy to Clipboard Export to CSV Export to Excel Export to PDF Print Search:

Column Visibility 0 Preview Selected

No	TimeStamp	Nama Unit	Nama Pelanggan	No Agenda	Nama Vendor	No SPK	Tanggal SPK	Tanggal Bayar	Status bayar	Action
1	2024-12-02 12:26:44	BANTUL	ISYANA	234346	PLN ULP Bantul	PLN-2024212-342	2024-12-02	2024-12-01	SUDAH	✓ 📄
2	2024-12-06 18:26:21	BANTUL	BERNADYA	2147483647	PLN ULP BANTUL	PLN-2024117-984	2024-12-17	2024-12-27	SUDAH	✓ 📄

Gambar 3.30 Data yang sudah berpindah ke halaman sudah bayar

1. updateBayar.php

Logika ini digunakan untuk memperbarui status progress (STATUS_PROGRESS) berdasarkan pilihan pengguna. Jika status progress yang dipilih adalah "Proses", maka status akan berubah menjadi "Sedang Diproses", dan data transaksi akan berpindah ke halaman "Sedang Proses". Proses ini dijelaskan dalam tiga gambar: gambar 3.31 menampilkan potongan kode logika php, gambar 3.32 menunjukkan pilihan status "Proses" pada form, dan gambar 3.33 menggambarkan data yang sudah berhasil berpindah ke halaman "Sedang Proses".

```
...
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id_permohonan = $_POST['id_permohonan'];
    $status_progress = $_POST['status_progress'];
    $status_nyala = $status_progress == "2" ? "2" : "1";
    $sql = "UPDATE permohonan SET STATUS_PROGRESS = ?, STATUS_NYALA = ? WHERE
    ID_PERMOHONAN = ?"; ...
}
```

Gambar 3.31 potongan kode logika php updateBayar.php

The image shows a web form titled "Ubah Status Progress" with a close button (X). Below the title, there are three radio button options stacked vertically: "ON PROGRESS" (the first one is selected and highlighted with a blue border), "ON PROGRESS" (the second one is highlighted with a blue background), and "BELUM".

Gambar 3.32 Form pilihan status pelanggan

Daftar Pelanggan Proses

Copy to ClipboardExport to CSVExport to ExcelExport to PDFPrint

Search:

Column Visibility

0 Preview Selected

No	Time Stamp	Nama Unit	Nama Pelanggan	No Agenda	Nama Vendor	No SPK	Tanggal SPK	Tanggal Bayar	Status bayar	Status Progress	Action
1	2024-12-02 12:26:44	BANTUL	ISYANA	234346	PLN ULP Bantul	PLN-2024212-342	2024-12-02	2024-12-01	SUDAH	ON PROGRESS	<div><div></div><div></div></div>

Showing 1 to 1 of 1 entries

Previous

1

Next

Gambar 3.33 Data yang sudah berpindah ke halaman sudah proses

2. updateProses.php

Deskripsi: logika php ini digunakan untuk memperbarui status progress menjadi "Sudah Nyala" atau tetap berada dalam status "Proses" berdasarkan pilihan pengguna. Jika status progress diperbarui menjadi "Sudah Nyala", maka data transaksi akan berpindah ke halaman "Sudah Nyala". Penjelasan proses ini dapat dilihat pada tiga gambar: gambar 3.34

menampilkan potongan kode logika php, gambar 3.35 menunjukkan tampilan form dengan pilihan status "Proses," dan gambar 3.36 menggambarkan data yang telah berhasil berpindah ke halaman "Sudah Nyala".

```
...
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id_permohonan = $_POST["id_permohonan"];
    if (isset($_POST["tanggal_nyala"])) {
        $tanggal_nyala = $_POST["tanggal_nyala"];
        $sql = "UPDATE permohonan SET tanggal_nyala = ?, status_nyala = '2',
            status_progress = '2' WHERE ID_PERMOHONAN = ?";
        $stmt = $conn->prepare($sql); $stmt->bind_param('si', $tanggal_nyala,
            $id_permohonan); }...
```

Gambar 3.34 potongan kode logika php updateProses.php

Gambar 3.35 Form pilihan status progress pelanggan

Daftar Pemohon

Copy to Clipboard Export to CSV Export to Excel Export to PDF Print Search: isy

Column Visibility 0 Preview Selected

Id Permohonan	Unit	Nama Pelanggan	Alamat	Jenis Transaksi	Peruntukan	Tarif Lama	Tarif Baru	Daya Lama	Daya Baru	Jumlah Pelanggan	Tanggal Permohonan	Jenis Pelanggan	Action
16	BANTUL	ISYANA	JALAN BANTUL NO 23	PASANG BARU	RUMAH TINGGAL	-	RUMAH TANGGA	0	1300	1	2024-12-02	TR	

Showing 1 to 1 of 1 entries (filtered from 16 total entries)

Previous 1 Next

Gambar 3.36 Data pelanggan yang sudah berpindah ke halaman nyala

Hasil Sprint

Hasil dari *sprint 2* menunjukkan bahwa pengelolaan status transaksi berhasil dilakukan sesuai dengan logika yang dirancang, di mana data dapat berpindah antar halaman dengan baik sesuai dengan status yang diperbarui. Penulis juga telah menerapkan validasi input yang efektif untuk mencegah risiko SQL Injection dengan menggunakan prepared statements, sehingga memberikan lapisan keamanan tambahan bagi data pengguna dan sistem secara keseluruhan.

Selain itu, pengujian terhadap logika php menunjukkan bahwa setiap logika php berfungsi dengan baik, dan semua perubahan status diterapkan sesuai dengan logika yang telah ditentukan, yang menegaskan bahwa sistem beroperasi secara optimal dan responsif. Pemisahan logika php berdasarkan tahap juga telah membuat pengelolaan kode menjadi lebih modular, mempermudah pengembangan dan pemeliharaan kode, serta meningkatkan keterbacaan dan kolaborasi di antara tim pengembang. Secara keseluruhan, hasil dan pengujian ini menunjukkan bahwa sistem telah dirancang dengan baik, memenuhi standar keamanan, dan memberikan kinerja yang optimal, serta memudahkan pengembangan di masa depan.

Sprint 3: Edit Data

Pada tahap ini, fokus diarahkan pada implementasi fitur Edit Data untuk memungkinkan pengguna memperbarui informasi yang sudah ada. Potongan kode untuk melakukan edit data dapat dilihat pada gambar 3.37, sementara gambar 3.38 menunjukkan tampilan form untuk mengedit data yang telah diterapkan.

```
function openEditModal(button, nama_pelanggan, alamat, idPelanggan,
jenisTransaksi, peruntukan, tarifLama, tarifBaru, dayaLama, dayaBaru,
jumlahPelanggan, tanggalPermohonan, jenisPelanggan){
var                                id_permohonan=                                button.getAttribute('data-id');
document.getElementById('idPer').value=id_permohonan;
document.getElementById('editModal').style.display = 'block'; }
```

Gambar 3.37 potongan kode untuk edit data

Gambar 3.38 Gambar form edit data pelanggan

Hasil *sprint*

Hasil dari *Sprint 3* menunjukkan bahwa fitur Edit Data telah berhasil dikembangkan, memungkinkan pengguna untuk memperbarui data transaksi secara dinamis. Pengguna dapat mengedit data melalui modal yang muncul tanpa perlu berpindah halaman, dan modal tersebut secara otomatis terisi dengan data terkait, memudahkan proses perubahan.

Dalam pengujian, penulis melakukan beberapa uji coba fungsional yang menunjukkan bahwa modal muncul dengan benar saat pengguna memilih opsi edit, data yang ditampilkan sesuai dengan transaksi yang dipilih, dan perubahan yang dilakukan tersimpan dengan baik. Selain itu, pengujian usability menunjukkan bahwa pengguna dapat dengan mudah mengakses fitur edit dan rata-rata waktu yang diperlukan untuk menyelesaikan tugas ini berkurang dibandingkan dengan metode sebelumnya.

Sprint 4: Preview Data Permohonan Pelanggan

Pada *sprint* ini, fokus pengembangan adalah memastikan fitur preview data pelanggan berjalan dengan baik untuk membantu pengguna memverifikasi detail permohonan. Logika preview ini menampilkan data pelanggan dalam modal dengan struktur rapi.

Fungsi untuk menampilkan modal preview:

Fungsi ini bertugas menampilkan modal dengan data yang diambil dari *backend*, kemudian mengisi tabel dengan data pelanggan. Fungsi ini dapat dilihat pada potongan kode digambar 3.39.

```
function showPreviewMultiple(data) {
  const previewTableBody =
    document.getElementById("previewBodyTable");
  previewTableBody.innerHTML = "";...
```

Gambar 3.39 potongan kode untuk modal preview

Hasil pengujian menunjukkan bahwa *backend* berhasil menyediakan data pelanggan dalam format JSON melalui logika php. Selain itu, fitur Preview telah diuji dan memungkinkan pengguna untuk melihat data pelanggan dengan mudah melalui modal yang rapi dan terstruktur, seperti yang ditunjukkan pada Gambar 3.40. Gambar tersebut menampilkan contoh tampilan hasil preview data pelanggan yang memberikan pengalaman pengguna yang intuitif dan efisien.

Preview 1 selected data

✕

Id Permohonan	Unit	Nama Pelanggan	Alamat	ID Pelanggan	Jenis Transaksi	Peruntukan	Tarif Lama
1	BANTUL	ISYANA	JALAN BANTUL NO 23	5210344433	PASANG BARU	RUMAH TINGGAL	-

[Export All](#)

Gambar 3.40 hasil tampilan preview

BAB IV REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

Pengalaman magang di PT PLN UP3 Yogyakarta mengungkapkan sejumlah kesenjangan antara teori akademis dan penerapannya di lapangan, khususnya dalam proyek pengembangan sistem *backend* dengan metode *Scrum*. Analisis relevansi ini melibatkan mata kuliah yang relevan seperti *Manajemen Proyek Teknologi Industri* (MPTI) dan *Pengembangan Aplikasi Berbasis Web* (PABW) untuk memahami penerapan teori dalam konteks praktik..

1.1.1 Teknis

Dalam proses magang, sejumlah kesenjangan ditemukan antara teori yang dipelajari pada mata kuliah MPTI dan PABW dengan penerapannya di lapangan.

a. Metode *Scrum* pada Pengembangan Sistem Backend

Pengelolaan sistem *backend* di PLN menggunakan metodologi *Scrum*, yang dalam teori memberikan fleksibilitas melalui iterasi berkelanjutan dan siklus *sprint*. Dalam mata kuliah MPTI, konsep project environment dan peran project manager diajarkan untuk mengelola perubahan serta memastikan kelancaran komunikasi antar tim. Namun, pada praktiknya, kebutuhan pengguna sering kali berubah selama proses pengembangan, sehingga rencana *sprint* perlu disesuaikan, yang tidak sepenuhnya selaras dengan teori yang diajarkan.

Selain itu, relevansi dengan PABW terlihat pada proses pengembangan aplikasi berbasis web yang dilakukan dalam proyek *backend*. Pada mata kuliah ini, dasar-dasar teknologi frontend dan *backend* dibahas secara teoritis. Namun, saat magang, tantangan nyata seperti keterbatasan data dan perubahan kebutuhan pengguna memerlukan solusi yang tidak tercakup secara eksplisit dalam materi kuliah.

b. Pelaksanaan *Daily Stand-Up Meetings*

Dalam teori *Scrum* yang diajarkan pada MPTI, daily stand-ups dirancang untuk memberikan pembaruan singkat tanpa menyita waktu. Namun, dalam praktik di PLN, rapat ini sering kali berlangsung lebih lama karena membahas detail teknis secara mendalam, yang mencerminkan pentingnya komunikasi lintas tim dalam *project environment*. Selain itu, waktu rapat kadang terbatas karena anggota tim memiliki tanggung jawab lain, yang menantang konsep efisiensi yang diajarkan dalam teori.

c. *Sprint Review dan Sprint Retrospective*

Sprint review dan *retrospective* tidak secara langsung dibahas dalam mata kuliah MPTI. Namun, dalam MPTI, diajarkan tentang peran manajer proyek dalam mengevaluasi hasil proyek, menganalisis risiko, dan menyusun langkah strategis untuk perbaikan di masa depan. Relevansi ini terlihat di PLN, di mana hasil *sprint review* dan *retrospective* digunakan untuk mengevaluasi performa tim serta merencanakan perbaikan. Meski demikian, pelaksanaan masukan dari *sprint retrospective* sering kali memerlukan waktu lebih lama karena tantangan implementasi pada sistem yang kompleks.

Pengalaman ini menunjukkan bahwa meskipun teori pada MPTI dan PABW sangat relevan, pelaksanaannya membutuhkan adaptasi terhadap situasi nyata, terutama dalam konteks proyek dengan kebutuhan dinamis dan tantangan teknis yang tinggi.

1.1.2 *Non Teknis*

Selama magang di PT PLN UP3 Yogyakarta, nilai-nilai AKHLAK menjadi bagian penting dari budaya kerja perusahaan, memberikan pengaruh positif pada pengembangan diri dan lingkungan kerja yang harmonis. AKHLAK, yang merupakan akronim dari Amanah, Kompeten, Harmonis, Loyal, Adaptif, dan Kolaboratif, menjadi landasan dalam etos kerja dan interaksi antar karyawan.

1. Amanah

Selama magang di PT PLN UP3 Yogyakarta, saya diberikan tanggung jawab untuk bekerja dengan data yang sangat penting bagi operasional perusahaan. Data yang saya kelola harus akurat, aman, dan tepat waktu. Dalam proses ini, saya harus memastikan bahwa setiap data yang diproses, disimpan, dan digunakan dalam pengembangan sistem *backend* sesuai dengan standar dan kebijakan yang berlaku di PLN. Saya belajar untuk memegang amanah dengan penuh tanggung jawab, karena setiap keputusan yang saya buat bisa memengaruhi banyak pihak. Mengelola data dengan integritas tinggi bukan hanya tentang menyelesaikan tugas, tetapi juga menjaga kepercayaan yang diberikan oleh perusahaan dan stakeholders.

2. Kompeten

Selama magang, saya terlibat dalam proses integrasi logika php serta pembuatan website skala besar yang memungkinkan perusahaan untuk mengelola data dan melakukan interaksi dengan pengguna secara lebih efektif. Dalam hal ini, saya mengembangkan kemampuan teknis saya terutama dalam hal integrasi sistem yang berjalan tanpa

menggunakan framework tertentu, melainkan dengan menerapkan pengkodean dan struktur sistem secara manual. Proses ini mengajarkan saya untuk memahami lebih dalam mengenai konsep dasar pengembangan sistem dan pentingnya efisiensi dalam merancang aplikasi yang dapat menangani skala besar tanpa bantuan framework. Pengalaman ini memberi saya wawasan yang lebih luas tentang *query* mengoptimalkan sistem pengembangan tanpa bergantung pada alat bantu yang lebih kompleks.

3. Harmonis

Di PT PLN UP3 Yogyakarta, saya merasakan betapa pentingnya menciptakan lingkungan kerja yang harmonis. Kolaborasi antar tim sangat mendukung tercapainya tujuan bersama dalam proyek pengembangan sistem. Saya melihat bagaimana tim yang terdiri dari berbagai divisi berkomunikasi dan bekerja sama dengan penuh pengertian. Sinergi yang terjalin antara anggota tim, baik itu dalam pengembangan sistem *backend* maupun dalam perencanaan proyek, sangat membantu dalam memecahkan masalah yang muncul. Pengalaman ini mengajarkan saya untuk selalu menjaga komunikasi yang baik dan saling mendukung agar hasil kerja tim dapat lebih maksimal.

4. Loyal

Loyalitas kepada perusahaan tercermin dalam dedikasi saya untuk menyelesaikan tugas dengan penuh komitmen, meskipun terdapat berbagai tantangan teknis. PT PLN UP3 Yogyakarta memberi kesempatan bagi saya untuk berkontribusi dalam pengembangan sistem yang akan berdampak langsung pada peningkatan layanan kepada masyarakat. Saya menyadari bahwa loyalitas bukan hanya tentang menyelesaikan tugas dengan baik, tetapi juga berusaha untuk terus meningkatkan sistem yang ada demi kepentingan publik. Saya berkomitmen untuk memberikan yang terbaik selama magang agar dapat memberikan kontribusi yang maksimal terhadap keberhasilan proyek yang sedang dikerjakan.

5. Adaptif

Sifat adaptif sangat ditekankan di PLN, terutama dalam pengembangan sistem menggunakan metode *Scrum*. Saya mengalami sendiri bagaimana rencana *sprint* sering kali harus diubah karena adanya kebutuhan baru dari pengguna. Sebagai contoh, pada salah satu *sprint*, perubahan besar harus dilakukan pada alur sistem karena adanya permintaan fitur tambahan. Saya belajar untuk tetap fleksibel dan berorientasi pada solusi, meskipun perubahan tersebut menambah beban kerja. Pengalaman ini juga mengasah kemampuan saya untuk berpikir cepat dan mencari alternatif terbaik dalam situasi yang serba dinamis.

6. Kolaboratif

Kolaborasi menjadi inti dari setiap aktivitas di PLN. Saya melihat bagaimana setiap divisi berkontribusi secara sinergis untuk menyelesaikan proyek dengan hasil terbaik. Selama magang, saya sering berkolaborasi dengan anggota tim lainnya untuk menyelesaikan tugas-tugas yang saling berkaitan, seperti integrasi data dan pengujian sistem. Salah satu pengalaman yang berkesan adalah saat mengerjakan *sprint review*, di mana seluruh tim memberikan umpan balik dan ide-ide untuk meningkatkan kualitas sistem. Proses ini mengajarkan saya bagaimana bekerja dalam tim yang menghargai masukan setiap anggotanya, sekaligus memastikan bahwa setiap keputusan diambil secara kolektif untuk kepentingan bersama.

4.2 Pembelajaran magang

Bagian ini merangkum pembelajaran selama magang di PT PLN UP3 Yogyakarta, yang mencakup manfaat, kendala, hambatan, dan tantangan dari sisi teknis dan non-teknis dalam pengembangan sistem berbasis metode *Scrum* serta penerapan budaya kerja.

4.2.1 Teknis

a. Manfaat

Selama magang di PT PLN UP3 Yogyakarta, user memperoleh banyak manfaat teknis yang memperdalam keterampilan dalam pengembangan sistem *backend*. Pengalaman ini mencakup pemahaman mendalam mengenai pengelolaan data *backend* dengan dukungan keterampilan teknis seperti visualisasi data menggunakan *Chart.js*, pemetaan interaktif dengan *Leaflet*, dan pengelolaan tabel data secara dinamis dengan *DataTables*. Selain itu, user juga memperkuat kemampuan dalam konfigurasi dan pengelolaan *database* MySQL untuk memastikan penyimpanan dan akses data yang efisien. Keterampilan-keterampilan ini memungkinkan pengembangan sistem yang responsif dan informatif, sesuai kebutuhan proyek di PLN.

b. Kendala, Hambatan, dan Tantangan

Selama magang, saya menghadapi tantangan utama terkait komunikasi internal dengan tim pengembang dan koordinasi dengan pihak PLN. Salah satu kendala terbesar adalah perbedaan antara tim magang dan tim internal PLN, yang sebagian besar terdiri dari pegawai tetap dengan *jobdesc* dan tanggung jawab yang sudah jelas. Dengan adanya struktur hierarki yang ketat, waktu yang tersedia untuk diskusi dan pengambilan keputusan seringkali terbatas. Pegawai PLN

memiliki kewajiban harian yang harus diutamakan, sehingga sangat sulit untuk menyamakan waktu yang tepat bagi semua pihak untuk berdiskusi.

Sering kali, kami membutuhkan persetujuan atau informasi tertentu yang penting untuk kemajuan proyek, tetapi terkendala oleh kesibukan para pegawai PLN. Hal ini terkadang menyebabkan keterlambatan dalam proses pengambilan keputusan atau pengolahan data yang dibutuhkan, yang pada gilirannya berpengaruh pada perkembangan proyek secara keseluruhan. Sebagai contoh, beberapa keputusan yang seharusnya dapat segera diambil untuk memperbaiki alur pengembangan sistem harus tertunda karena ketidakcocokan waktu antara anggota tim internal PLN dan tim magang.

Untuk mengatasi masalah ini, tim pengembang berperan penting dalam menjembatani komunikasi antara pihak PLN dan saya sebagai magang. Mereka membantu mengatur jadwal yang lebih fleksibel dan menyederhanakan beberapa proses komunikasi agar dapat berjalan lebih efisien. Meskipun demikian, pengalaman ini menggarisbawahi pentingnya memiliki fleksibilitas dalam berkomunikasi dan berkoordinasi, khususnya dalam lingkungan kerja yang memiliki struktur formal dan hirarki yang kaku. Saya belajar untuk lebih kreatif dalam mencari solusi untuk mengatasi kendala waktu dan komunikasi ini, baik dengan memanfaatkan alat komunikasi online, pengaturan jadwal lebih fleksibel, maupun dengan merancang sistem yang lebih efisien dalam alur pengambilan keputusan.

4.2.2 Non Teknis

a. Manfaat

Di sisi non-teknis, magang ini memberikan pengalaman yang meningkatkan soft skills seperti networking dan critical thinking, yang terasah melalui proses user requirement. Peran ganda yang diemban user juga mendukung pengembangan keterampilan dalam manajemen waktu untuk menyelesaikan tugas tepat waktu.

b. Kendala, Hambatan, dan Tantangan.

Pengelolaan waktu sudah dioptimalkan untuk menangani tingginya beban kerja, meskipun pekerjaan tersebut membutuhkan konsentrasi tinggi untuk mempertahankan produktivitas. Tantangan komunikasi juga muncul dalam upaya mencegah kesalahpahaman, khususnya dalam penggunaan istilah teknis yang kurang familiar bagi beberapa karyawan. Untuk mengatasi hal ini, user berlatih

menyederhanakan bahasa teknis agar lebih mudah dipahami oleh semua pihak yang terlibat dalam proyek.

Pembelajaran teknis dan non-teknis yang diperoleh selama magang ini menyoroti pentingnya fleksibilitas dalam menerapkan teori ke dalam praktik nyata, serta menyesuaikan metode *Scrum* dengan struktur organisasi formal di PT PLN UP3 Yogyakarta. Pengalaman ini memberikan wawasan akademik dan praktis yang bermanfaat sebagai acuan bagi pengembangan metodologi hybrid yang menggabungkan elemen-elemen dari berbagai metodologi atau kerangka kerja yang berbeda untuk menciptakan solusi yang lebih fleksibel dan sesuai dengan kebutuhan proyek tertentu pada proyek-proyek serupa di masa depan.

BAB V

PENUTUP

5.1 Kesimpulan

Kesimpulan dari laporan magang ini adalah bahwa pengembangan dan implementasi *backend* untuk Sistem Informasi Bersama (SIBER) di PT PLN UP3 Yogyakarta telah memberikan kontribusi signifikan dalam meningkatkan efisiensi operasional dan kualitas pelayanan pelanggan. Melalui integrasi teknologi *Automatic Meter Reading* (AMR) dalam sistem, PLN dapat memantau dan membaca data meteran pelanggan secara otomatis, yang mengurangi kerugian daya listrik dan meningkatkan pengelolaan konsumsi energi.

Tujuan magang ini yang berfokus pada pemahaman terhadap pengelolaan dan pengembangan sistem *backend*, serta penerapan metodologi *Scrum*, tercapai dengan baik. Implementasi *dashboard* AMR yang terintegrasi memungkinkan divisi-divisi terkait untuk melakukan pemantauan kinerja secara *real-time* dan analisis data yang lebih cepat serta akurat. Ini tidak hanya mendukung Divisi Transaksi Energi dalam mengelola transaksi listrik, tetapi juga memberikan manfaat bagi Divisi Perencanaan dan Konstruksi dalam memantau indikator kinerja dan status layanan.

Secara keseluruhan, magang ini berhasil mencapai tujuan untuk memahami tantangan yang dihadapi dalam pengembangan sistem berbasis teknologi informasi, serta bagaimana metodologi *Scrum* dapat mendukung adaptasi terhadap perubahan kebutuhan proyek. Pengalaman ini memberikan wawasan penting mengenai penerapan teknologi yang tepat untuk meningkatkan kualitas pelayanan pelanggan dan efisiensi operasional di PT PLN UP3 Yogyakarta. Diharapkan, hasil dari magang ini dapat memberikan kontribusi positif terhadap pengembangan lebih lanjut sistem informasi perusahaan dan meningkatkan mutu layanan kepada masyarakat.

5.2 Saran

Setelah melakukan magang selama 6 bulan di PT PLN UP3 Yogyakarta penulis dapat memberikan saran sebagai berikut:

A. Program Studi Informatika

1. Implementasi *backend* Sistem Informasi Bersama (SIBER) di PT PLN UP3 Yogyakarta memberikan gambaran nyata tentang tantangan yang dihadapi dalam industri energi, serta pentingnya penyesuaian kurikulum di program studi Informatika agar lebih sesuai dengan kebutuhan industri tersebut. Berdasarkan

pengalaman selama magang, disarankan beberapa langkah perbaikan untuk memperkuat kurikulum dan memperbaiki penghubungan antara pendidikan akademik dan praktik industri. Saran-saran tersebut adalah sebagai berikut:

2. Menambahkan mata kuliah pilihan terkait teknologi energi dan sistem informasi kelistrikan

Selama magang, saya bekerja dengan teknologi yang sangat spesifik untuk sektor energi, seperti *Automatic Meter Reading (AMR)*, yang memerlukan pemahaman mendalam tentang sistem kelistrikan. Namun, tidak banyak pembahasan tentang topik-topik tersebut dalam kurikulum yang ada di Program Studi Informatika. Menambahkan mata kuliah pilihan yang membahas lebih lanjut tentang teknologi energi dan sistem informasi kelistrikan akan mempersiapkan mahasiswa untuk menghadapinya secara lebih praktis ketika terjun ke industri energi, seperti yang saya alami di PLN.

3. Mengembangkan kerja sama dengan perusahaan utilitas untuk memberikan pemahaman mendalam tentang kebutuhan spesifik industri Selama magang di PLN, saya merasakan adanya kesenjangan antara teori yang dipelajari di kampus dengan tantangan praktis yang dihadapi di industri energi. Untuk mengatasi hal ini, penting bagi Program Studi Informatika untuk memperkuat kerja sama dengan perusahaan utilitas seperti PLN, sehingga mahasiswa dapat memperoleh pemahaman yang lebih baik tentang kebutuhan spesifik industri energi. Kolaborasi semacam ini dapat memberikan pengalaman langsung yang lebih relevan dan mempersiapkan mahasiswa untuk masalah yang akan mereka hadapi di lapangan.

4. Merancang kurikulum yang lebih responsif terhadap perkembangan teknologi di sektor energi

Industri energi sangat dinamis, dengan teknologi baru yang terus bermunculan, seperti AMR yang digunakan di PLN. Kurikulum yang tidak responsif terhadap perkembangan ini bisa menyebabkan mahasiswa kurang siap menghadapi tantangan terbaru di industri. Oleh karena itu, merancang kurikulum yang lebih adaptif terhadap perkembangan teknologi di sektor energi akan membantu mahasiswa lebih siap untuk memanfaatkan teknologi terbaru dan menyesuaikan diri dengan kebutuhan industri yang terus berkembang.

5. Membuat platform digital untuk informasi lowongan magang dan persyaratan aplikasi

Selama magang, saya menyadari bahwa akses ke informasi mengenai lowongan magang dan persyaratan yang diperlukan untuk bekerja di industri energi cukup terbatas. Oleh karena itu, saya menyarankan untuk membuat platform digital yang menyediakan informasi terpusat mengenai lowongan magang di perusahaan energi, persyaratan aplikasi, dan panduan persiapan magang yang komprehensif. Dengan adanya platform ini, mahasiswa dapat lebih mudah mengakses peluang magang, memahami apa yang dibutuhkan oleh perusahaan energi, serta mempersiapkan diri mereka secara maksimal sebelum memulai magang.

Dengan mengimplementasikan saran-saran ini, diharapkan Program Studi Informatika dapat lebih siap menghasilkan lulusan yang kompeten, memiliki pemahaman mendalam tentang industri energi, serta mampu menghadapi tantangan dan peluang yang ada di sektor tersebut.

B. Perusahaan (PT PLN UP3 Yogyakarta)

Berdasarkan implementasi *backend* SIBER, perusahaan disarankan untuk:

1. Mengembangkan program *monitoring* yang lebih terstruktur bagi mahasiswa magang
2. Merancang pembagian tugas yang lebih proporsional untuk tim pengembangan
3. Menyediakan kesempatan rotasi dan eksplorasi berbagai divisi teknologi informasi
4. Membangun ekosistem yang mendukung pengembangan inovasi dari mahasiswa magang

C. Mahasiswa Magang

Untuk mengoptimalkan kontribusi dan pengalaman magang, mahasiswa disarankan untuk:

1. Aktif mencari pengetahuan tambahan tentang teknologi *backend* dan sistem informasi
2. Membangun kemampuan adaptasi yang tinggi terhadap kebutuhan industri
3. Mengembangkan soft skills seperti komunikasi dan kerja tim
4. Tidak hanya fokus pada aspek teknis, tetapi juga memahami konteks bisnis dan operasional

DAFTAR PUSTAKA

- Cohn, M. (2010). *Succeeding with Agile: Software Development Using Scrum*. Upper Saddle River: Addison-Wesley Professional.
- DuBois, P. (2013). *MySQL (5th Edition)*. Boston: Addison-Wesley Professional.
- Erwan Effendi, S. H. (2023). Komponen sistem Informasi. *Jurnal Pendidikan dan Konseling (JPDK)*.
- Few, S. (2013). *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring*. Burlingame: Analytics Press.
- Garrett, J. J. (2005). *Ajax: A New Approach to Web Applications*. New York: Association for Computing Machinery (ACM).
- Laudon, K. C. (2018). *Management Information Systems: Managing the Digital Firm*. New Jersey (NJ): Pearson.
- Lerdorf, R. (1995). *Development of Early PHP: History and Basic Concepts*. Waterloo: University of Waterloo Press.
- Myers, G. J. (2011). *The Art of Software Testing*. New York: IEEE Computer Society.
- PT.PLN . (2023, desember 9). *Web PLN tentang kami profil perusahaan*. Retrieved from Website PLN: <https://web.pln.co.id/tentang-kami/profil-perusahaan>
- scrum. (2024). *learning series*. Retrieved from scrum.org: <https://www.scrum.org/learning-series/what-is-scrum/>
- Severance, C. (2012). *Introduction to Web Development*. CreateSpace : CreateSpace Independent Publishing Platform.
- Straub, S. C. (2014). *Pro Git*. Berkeley: Apress.
- Sutherland, K. S. (2017, November 17). *The Scrum Guide*. Retrieved from Scrum.org: <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>
- Syachbana, R. &. (2022). Kajian Literatur: Pengembangan Sistem Informasi Manajemen Akademik. *Jurnal Ilmiah Teknologi Informasi*.
- Rohman, A., & Bhakti, H. D. (2023). Perancangan Sistem Informasi Persediaan Barang Berbasis Web. *Syntax Literate ; Jurnal Ilmiah Indonesia*, 7(9), 15304–15313. doi:10.36418/syntax-literate.v7i9.14255
- Supriatmaja, G. A., Pratama, I. P. M. Y., Mahendra, K., Widyaputra, K. D. D., Deva, J., & Mahendra, G.S. (2023). Sistem Informasi Perpustakaan Menggunakan Framework Bootstrap Dengan PHP Native dan Database MySQL Berbasis Web Pada SMP Negeri 2 Dawan, 1(1).
- Susilo, B., & Azimah, A. (2023). Penerapan Metode Agile Scrum Pada Perancangan Sistem Informasi Pelaporan Keuangan BUMDesa. *Jutisi : Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, 12(2), 495.
- Wibowo, A., & Darwati, I. (2022). Perancangan Sistem Informasi Pengelolaan Sarana dan Prasarana dengan Pengujian User Acceptance Testing. *Reputasi: Jurnal Rekayasa Perangkat Lunak*, 3(1), 1–6.
- Wijaya, T. (2018). Penerapan AJAX dalam Aplikasi Mobile Berbasis Web Untuk Meningkatkan Efisiensi Bandwidth. *Techno.Com*, 17(2), 197–207.

LAMPIRAN

Tabel aktivitas pengembangan Sistem Informasi Bersama (SIBER)

NO	Aktivitas
1	Penerimaan dan pembekalan mahasiswa magang
2	Observasi dan analisa kebutuhan sistem
3	Observasi dan analisa kebutuhan <i>database</i>
4	Pembuatan struktur <i>database</i> pln
5	Pembuatan logika pengambilan data untuk <i>dashboard</i> AMR
6	Pembuatan grafik dengan <i>chart.js</i>
7	Pembuatan maps dengan leaflet
8	Pembuatan table penyajian data menggunakan <i>dataTables</i>
9	Pembuatan fitur edit dengan modal untuk <i>dashboard</i> AMR
10	Pembuatan fitur <i>import</i> dan <i>export</i> data untuk <i>dashboard</i> AMR
11	Proses integrasi <i>backend</i> dan <i>frontend</i> untuk <i>dashboard</i> AMR
12	Pengujian sistem untuk bagian <i>dashboard</i> AMR
13	Pembuatan <i>dashboard</i> kinerja
14	Pembuatan fitur <i>import</i> dan <i>export</i> data untuk <i>dashboard</i> kinerja
15	Pembuatan grafik untuk masing-masing kategori data menggunakan <i>chart.js</i>
16	Proses integrasi dengan <i>backend</i> dan <i>frontend</i> untuk <i>dashboard</i> kinerja
17	Pengujian sistem untuk bagian <i>dashboard</i> kinerja
18	Pembuatan struktur <i>database</i> simantri
19	Pembuatan <i>dashboard</i> konstruksi
20	Pembuatan form tambah data untuk <i>dashboard</i> konstruksi pada bagian ULP
21	Pembuatan table untuk <i>dashboard</i> simantri bagian ULP dan UP3
22	Pembuatan fitur edit dengan modal untuk <i>dashboard</i> konstruksi
23	Pembuatan fitur <i>export</i> untuk <i>dashboard</i> konstruksi bagian UP3
24	Pembuatan logika untuk update data pada setiap transaksi pada bagian UP3
25	Proses integrasi <i>backend</i> dan <i>frontend</i>
26	Pengujian sistem untuk <i>dashboard</i> konstruksi