PENGEMBANGAN APLIKASI WEBSITE MANAJEMEN KOST NESTIFY DENGAN PENDEKATAN *WATERFALL* DAN EVALUASI *BLACK-BOX TESTING*



Disusun Oleh:

N a m a : Andika Raka Wicaksono
NIM : 21523146

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2025

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGEMBANGAN APLIKASI WEBSITE MANAJEMEN KOST NESTIFY DENGAN PENDEKATAN *WATERFALL* DAN EVALUASI *BLACK-BOX TESTING*

TUGAS AKHIR

Disusun Oleh:

N a m a : Andika Raka Wicaksono

NIM : 21523146

Yogyakarta, 19 Juli 2025

Pembimbing,

(Ari Sujarwo, S.Kom., M.I.T.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGEMBANGAN APLIKASI WEBSITE MANAJEMEN KOST NESTIFY DENGAN PENDEKATAN *WATERFALL* DAN EVALUASI *BLACK-BOX TESTING*

TUGAS AKHIR

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

Yogyakarta, 7 Agustus 2025

Tim Penguji

Ari Sujarwo, S.Kom., M.I.T

Anggota 1

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

Anggota 2

Kurniawan Dwi Irianto, S.T., M.Sc.

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 : Andika Raka Wicaksono

NIM : 21523146

Tugas akhir dengan judul:

PENGEMBANGAN APLIKASI WEBSITE MANAJEMEN KOST NESTIFY DENGAN PENDEKATAN *WATERFALL* DAN EVALUASI *BLACK-BOX TESTING*

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, 19 Juli 2025



(Andika Raka Wicaksono)

HALAMAN PERSEMBAHAN

Dengan rasa syukur yang mendalam, skripsi ini saya persembahkan kepada kedua orang tua yang paling saya cintai dan hormati, Bapak Tris Budiono dan Ibu Harti Pamuji. Terima kasih atas segala doa, dukungan moril dan materiil, kesabaran, serta kasih sayang tak terhingga yang senantiasa mengiringi setiap langkah saya. Berkat keikhlasan dan pengorbanan Bapak dan Ibu, saya dapat menyelesaikan pendidikan Program Sarjana ini.

Ucapan terima kasih juga saya haturkan kepada seluruh sahabat dan teman-teman di Program Studi Informatika, Komunitas Musik Informatika, serta Unisi Music Community. Terima kasih telah memberikan semangat, bantuan, dan menjadi teman diskusi dalam menghadapi berbagai tantangan selama masa perkuliahan. Semoga kebersamaan kita menjadi kenangan yang indah dan kita semua dapat meraih cita-cita.

Terakhir, skripsi ini saya persembahkan untuk para pembaca yang menjadikannya sebagai referensi. Semoga hasil penelitian ini dapat memberikan manfaat dan kontribusi bagi pengembangan ilmu pengetahuan, khususnya di bidang rekayasa perangkat lunak.

HALAMAN MOTO

“...niscaya Allah akan mengangkat (derajat) orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu beberapa derajat..."

(QS. Al-Mujadilah: 11)

"Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya."

(QS. Al-Baqarah: 286)

"...Sesungguhnya Allah tidak akan mengubah keadaan suatu kaum sebelum mereka mengubah keadaan diri mereka sendiri..."

(QS. Ar-Ra'd: 11)

“Kita adalah apa yang kita lakukan berulang kali. Karena itu, keunggulan bukanlah sebuah tindakan, melainkan sebuah kebiasaan”

(Aristoteles)

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh,

Alhamdulillahirabbil 'alamin, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT, yang atas rahmat, hidayah, dan karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul "PENGEMBANGAN APLIKASI WEBSITE MANAJEMEN KOST NESTIFY DENGAN PENDEKATAN WATERFALL DAN EVALUASI BLACK-BOX TESTING". Shalawat serta salam semoga senantiasa tercurah kepada teladan kita, Nabi Muhammad SAW, beserta keluarga, sahabat, dan para pengikutnya hingga akhir zaman.

Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan dan memperoleh gelar Sarjana Komputer pada Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Penelitian ini merupakan upaya untuk menerapkan ilmu yang telah penulis peroleh selama masa perkuliahan, dengan fokus pada pengembangan sebuah prototipe aplikasi berbasis website yang fungsional dan responsif untuk manajemen indekos menggunakan metode *Research and Development (R&D)* dengan pendekatan model *Waterfall*.

Penulis menyadari bahwa penyusunan skripsi ini tidak akan terwujud tanpa adanya bimbingan, dukungan, dan doa dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan terima kasih yang tulus kepada:

1. Allah SWT, atas segala nikmat sehat, kemudahan, dan kelancaran yang tak terhingga selama proses penelitian dan penulisan skripsi ini.
2. Orang tua tercinta, Bapak Tris Budiyono dan Ibu Harti Pamuji, beserta seluruh keluarga besar yang telah memberikan dukungan moral, materiel, doa, dan kasih sayang yang tiada henti.
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc**.**, selaku Ketua Jurusan Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.
4. Bapak Dhomas Hatta Fudholi S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika Program Sarjana, Fakultas Teknologi Industri, Universitas Islam Indonesia.
5. Bapak Ari Sujarwo, S.Kom., MIT. (Hons), selaku Dosen Pembimbing, yang telah dengan sabar meluangkan waktu, tenaga, dan pikiran untuk memberikan arahan, bimbingan, dan masukan yang sangat berharga dari awal hingga akhir penyusunan skripsi ini.

6. Seluruh Bapak dan Ibu Dosen Program Studi Informatika Universitas Islam Indonesia yang telah ikhlas berbagi ilmu dan wawasan yang sangat bermanfaat selama masa perkuliahan.
7. Penghargaan istimewa untuk Nabila, selaku teman sekaligus pasangan terkasih, yang telah menjadi sumber semangat, motivasi, dan teman diskusi dalam menghadapi setiap tantangan selama masa perkuliahan dan pengerjaan skripsi.
8. Rekan-rekan seperjuangan di Program Studi Informatika UII angkatan 2021, terima kasih atas kebersamaan, dukungan, dan kenangan indah yang telah kita lalui bersama.
9. Seluruh rekan di Komunitas Musik Informatika (KOSMIK UII) dan *Unisi Music Community* (UMC UII) yang telah memberikan banyak pelajaran dan pengalaman berharga selama masa perkuliahian ini.
10. Semua pihak yang tidak dapat penulis sebutkan satu per satu, terima kasih atas segala bentuk bantuan dan dukungan yang telah diberikan.

Penulis menyadari sepenuhnya bahwa skripsi ini masih memiliki banyak kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan untuk perbaikan di masa mendatang. Semoga skripsi ini dapat memberikan manfaat bagi para pembaca, almamater, serta pengembangan teknologi di bidang sistem informasi manajemen properti.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Yogyakarta, 19 Juli 2025



(Andika Raka Wicaksono)

SARI

Aplikasi Nestify merupakan sebuah platform berbasis website yang dirancang untuk memberikan kemudahan dalam pengelolaan properti indekos. Keberhasilan operasional aplikasi ini sangat bergantung pada keandalan fungsionalitas yang sesuai dengan alur penggunaan yang diharapkan. Oleh karena itu, penelitian ini berfokus pada memastikan setiap fitur utama, mulai dari autentikasi pengguna hingga pengelolaan properti, dapat berjalan dengan benar dan bebas dari kesalahan fungsional.

Penelitian ini menggunakan metode *Research and Development* (R&D) dengan model pengembangan *Waterfall*. Pada tahap verifikasi, diterapkan pengujian *Black-box* dengan teknik *Boundary Value Analysis* (BVA) untuk mengevaluasi fungsionalitas aplikasi secara komprehensif. Pengujian ini mencakup delapan modul utama dari sisi Pemilik Kost (Registrasi Akun, Login Akun, Kelola Properti, Data Penghuni, Manajemen Tagihan) dan sisi Penyewa Kost (Pendaftaran Sewa, Pembayaran Tagihan, Pengajuan Komplain), dengan fokus pada pengujian nilai-nilai batas yang rentan terhadap kesalahan.

Hasil pengujian komprehensif pada total 65 kasus uji menunjukkan kinerja sistem yang sangat positif. Sebanyak 60 kasus uji (*Pass*) dan 5 kasus uji (*Fail*) teridentifikasi, menghasilkan *Success Rate* keseluruhan sebesar 92.3%. Berdasarkan kriteria penilaian, aplikasi Nestify secara fungsional telah berjalan dengan sangat baik dan berada dalam kategori "Sangat Baik". Meskipun demikian, kegagalan yang teridentifikasi tersebar pada aspek validasi *input* (misalnya batas panjang email, email unik, format *file* unggahan) dan penanganan *field* wajib pada beberapa fitur kunci.

Penelitian ini membuktikan bahwa pendekatan pengujian *Black-box* dengan teknik BVA sangat efektif dalam mengidentifikasi kelemahan kritis pada sistem, menegaskan pentingnya pengujian terstruktur untuk menjamin kualitas dan keandalan sebuah aplikasi website sebelum digunakan oleh pengguna akhir.

Kata kunci: manajemen indekos, pengembangan website, pengujian *black-box*, *boundary value analysis*, kualitas perangkat lunak.

GLOSARIUM

Boundary Value Analysis (BVA)	Teknik pengujian yang fokus pada nilai-nilai di batas atau tepi sebuah rentang data
Black-box Testing	Metode pengujian fungsionalitas dari sudut pandang pengguna, tanpa melihat kode program.
Entity Relationship Diagram (ERD)	Model visual untuk merancang basis data dengan menggambarkan hubungan antar entitas
Filament	Paket untuk kerangka kerja Laravel yang berfungsi mempercepat pembuatan panel administrasi
Laravel	Kerangka kerja (framework) berbasis bahasa pemrograman PHP untuk membangun aplikasi website
Indekos	Tempat tinggal sementara berupa kamar yang disewakan untuk pelajar, mahasiswa, atau pekerja
Prototype	Versi awal sebuah produk yang dibuat untuk pengujian dan evaluasi konsep sebelum produksi penuh
Agile	Metode pengembangan perangkat lunak secara bertahap dan cepat, dengan fokus pada kolaborasi tim, fleksibilitas.
Test case	Skenario pengujian yang berisi langkah-langkah dan hasil yang diharapkan untuk memverifikasi fungsionalitas sistem
Waterfall	Model pengembangan perangkat lunak yang tahapannya berurutan, di mana satu tahap harus selesai sebelum memulai tahap berikutnya
Wireframe	Sketsa dasar atau kerangka visual dari tata letak sebuah halaman website
User Interface (UI)	Elemen visual yang memungkinkan pengguna berinteraksi dengan aplikasi
User Experience (UX)	Pengalaman keseluruhan pengguna saat berinteraksi dengan aplikasi
User Centered Design (UCD)	Metodologi desain yang menempatkan pengguna sebagai fokus utama pengembangan
Dashboard	Halaman ringkasan utama yang menampilkan informasi penting dan akses cepat ke fitur

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI ix
GLOSARIUM x
DAFTAR ISI xi
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 3
1.5 Manfaat Penelitian 4
1.6 Metodologi Penelitian 4
1.7 Sistematika Penulisan 5
BAB II KAJIAN PUSTAKA 6
2.1 Pengembangan Aplikasi Berbasis Website 6
2.2 Indekos 8
2.2.1 Tinjauan Pertumbuhan dan Celah Digitalisasi di Pasar Indekos 8
2.2.2 Klasifikasi Indekos dan Implikasinya pada Desain Sistem 9
2.2.3 Pemanfaatan Aplikasi Website sebagai Solusi Manajemen Indekos 10
2.3 Model Pengembangan Perangkat Lunak *Waterfall* 11
2.3.1 Justifikasi Pemilihan Model *Waterfall* untuk Nestify 13
2.3.2 Perbandingan dengan Metodologi *Agile* 13
2.4 *Entity Relationship Diagram* 14
2.5 *Prototype* 15
2.6 *Black Box Testing* 15
2.7 Penelitian Terdahulu 16
BAB III METODOLOGI PENELITIAN 19

3.1 Analisis Kebutuhan 20
3.2 Desain Sistem 22
3.2.1 *Entity Relationship Diagram* (ERD) 23
3.2.2 *Wireframe* 26
3.3 Penulisan Kode Program 29
3.4 Pengujian Program 29
3.4.1 Strategi dan Lingkungan Pengujian 30
3.4.2 Perancangan *Test Case* 31
3.4.3 Kriteria Keberhasilan dan Penanganan Hasil 58
3.5 Pemeliharaan/ *Maintenance* 60
BAB IV HASIL DAN PEMBAHASAN 62
4.1 Implementasi Antarmuka Pengguna 62
4.1.1 Halaman Publik dan Autentikasi 62
4.1.2 Antarmuka Fungsionalitas Pemilik Kos 64
4.1.3 Antarmuka Fungsionalitas Penyewa Kos 70
4.1.4 Antarmuka Pengaturan Akun 73
4.2 Hasil Pengujian Fungsional 74
4.2.1 Hasil Pengujian Modul Registrasi Dan *Login* Akun 74
4.2.2 Hasil Pengujian Modul Tampilan Pemilik 76
4.2.3 Hasil Pengujian Modul Tampilan Penyewa 80
4.3 Pembahasan 82
4.3.1 Ringkasan Hasil Pengujian Modul 83
4.3.2 Analisis Kinerja Pengujian Keseluruhan 84
4.3.3 Analisis Kegagalan Fungsional 84
BAB V KESIMPULAN DAN SARAN 86
5.1 Kesimpulan 86
5.2 Saran 87
DAFTAR PUSTAKA 88
LAMPIRAN 94

DAFTAR TABEL

Tabel 2.1 Persamaan, Perbedaan, dan Celah Penelitian 18
Tabel 3.1 Cek Validasi Registrasi 32
Tabel 3.2 Boundary Value Analysis Registrasi 35
Tabel 3.3 Cek Validasi Login 35
Tabel 3.4 Kelas Boundary Value Analysis Login 36
Tabel 3.5 Cek Validasi Kelola Properti 37
Tabel 3.6 Kelas Boundary Value Analysis Kelola Properti 39
Tabel 3.7 Cek Validasi Data Penghuni 41
Tabel 3.8 Kelas Boundary Value Analysis Data Penghuni 43
Tabel 3. 9 Cek Validasi Manajemen Tagihan 43
Tabel 3.10 Kelas Boundary Value Analysis Manajemen Tagihan 44
Tabel 3.11 Cek Validasi Pendaftaran Sewa 46
Tabel 3.12 Kelas Boundary Value Analysis Pendaftaran Sewa 48
Tabel 3.13 Cek Validasi Pembayaran Tagihan 48
Tabel 3.14 Kelas Boundary Value Analysis Pembayaran Tagihan 49
Tabel 3.15 Cek Validasi Pengajuan Komplain 49
Tabel 3.16 Kelas Boundary Value Analysis Pengajuan Komplain 52
Tabel 3.17 Perancangan Kasus Uji Fungsionalitas Registrasi Akun Pemilik Kos 53
Tabel 3.18 Perancangan Kasus Uji Fungsionalitas Login Akun Pemilik Kos 54
Tabel 3.19 Perancangan Kasus Uji Fungsionalitas Kelola Properti 55
Tabel 3.20 Perancangan Kasus Uji Fungsionalitas Data Penghuni 56
Tabel 3.21 Perancangan Kasus Uji Fungsionalitas Manajemen Tagihan 56
Tabel 3.22 Perancangan Kasus Uji Fungsionalitas Pendaftaran Sewa 57
Tabel 3.23 Perancangan Kasus Uji Fungsionalitas Pembayaran Tagihan Penyewa 57
Tabel 3.24 Perancangan Kasus Uji Fungsionalitas Pengajuan Komplain Penyewa 58
Tabel 4.1 Hasil Pengujian Modul Registrasi Akun 75
Tabel 4.2 Hasil Pengujian Modul Login 76
Tabel 4.3 Hasil Pengujian Modul Tampilan Pemilik Kelola Properti 77
Tabel 4.4 Hasil Pengujian Modul Tampilan Pemilik Data Penghuni 78
Tabel 4.5 Hasil Pengujian Modul Tampilan Pemilik Manajemen Tagihan 79
Tabel 4.6 Hasil Pengujian Modul Tampilan Penyewa Pendaftaran Properti 80
Tabel 4.7 Hasil Pengujian Modul Tampilan Penyewa Pembayaran Tagihan Penyewa 81
Tabel 4.8 Hasil Pengujian Modul Tampilan Penyewa Pengajuan Komplain Penyewa 82

DAFTAR GAMBAR

Gambar 2.1 Data Tingkat penetrasi internet di Indonesia 2024 7
Gambar 2.2 Metode Waterfall 11
Gambar 2.3 Simbol ERD 14
Gambar 3.1 Bagan Kontribusi Penelitian 19
Gambar 3.2 User Flow Pemilik Indekos 21
Gambar 3.3 User Flow Penyewa Indekos 22
Gambar 3.4 Entity Relationship Diagram (ERD) Aplikasi Nestify 24
Gambar 3.5 Wireframe Registrasi Pengguna oleh Fauzi (2024) 27
Gambar 3.6 Wireframe Beranda Pemilik Kos oleh Fauzi (2024) 28
Gambar 3.7 Wireframe Beranda Penyewa oleh Fauzi (2024) 28
Gambar 3.8 Wireframe Beranda Penyewa oleh Fauzi (2024) 29
Gambar 4.1 Tampilan Landing page Aplikasi Nestify 63
Gambar 4.2 Tampilan Halaman Registrasi Pengguna 63
Gambar 4.3 Tampilan Halaman Login Pengguna 64
Gambar 4.4 Tampilan Dashboard Utama Pemilik Kos 65
Gambar 4.5 Tampilan Halaman Kelola Properti 65
Gambar 4.6 Tampilan Halaman Properti Setelah Penambahan Kos Pertama 66
Gambar 4.7 Tampilan Halaman Kamar 67
Gambar 4.8 Halaman Data Penghuni 67
Gambar 4.9 Tampilan Halaman Pengajuan Penghuni 68
Gambar 4.10 Tampilan Halaman Manajemen Tagihan 69
Gambar 4.11 Tampilan Halaman Kelola Komplain 69
Gambar 4.12 Tampilan Dasbor Utama Penyewa (Belum Terdaftar di Kos) 70
Gambar 4.13 Tampilan Dasbor Penyewa (Terdaftar di Kos) 71
Gambar 4.14 Tampilan Halaman Kos Saya 71
Gambar 4.15 Tampilan Halaman Tagihan Saya 72
Gambar 4.16 Tampilan Halaman Komplain Saya 73
Gambar 4.17 Tampilan Halaman Settings 73

BAB I
PENDAHULUAN

1.1 Latar Belakang

Indekos (biasa disebut kos) merupakan tempat tinggal sementara yang umumnya disewakan kepada para pendatang, seperti pelajar, mahasiswa, atau pekerja yang tinggal jauh dari rumah mereka (Effendy dkk., 2020). Pilihan indekos banyak diminati karena biayanya lebih terjangkau dibandingkan menyewa apartemen atau rumah, khususnya di kalangan milenial kota (D. Oktavallyan dkk., 2021). Selain berfungsi sebagai tempat tinggal, indekos juga sering kali menjadi tempat belajar dan beristirahat yang memengaruhi produktivitas penghuninya (Engineering dkk., 2023).

Seiring dengan kemajuan teknologi informasi dan komunikasi, peran internet semakin besar dalam mengubah berbagai aspek kehidupan, termasuk dalam dunia bisnis indekos. Pemanfaatan teknologi ini semakin penting bagi masyarakat modern yang mengandalkan kecepatan dan kemudahan akses informasi (Cholik, 2021). Perkembangan tersebut memberikan berbagai pengaruh dalam kehidupan sosial, sehingga setiap individu terdorong untuk menggunakan serta memanfaatkan kemajuan ini (Danuri, 2019). Di Indonesia, pertumbuhan penggunaan internet yang pesat telah memberikan dampak yang signifikan terhadap berbagai aspek kehidupan masyarakat, terutama di kalangan anak muda dan sektor ekonomi. Dengan jumlah pengguna internet yang mencapai 204,7 juta pada awal 2021 (Rahayu dkk., 2022), transformasi dari konvensional ke digital berkembang dengan cepat. Transformasi ini dipercepat oleh pandemi COVID-19, sehingga kondisi ini memaksa bisnis untuk segera beradaptasi dengan perilaku konsumen yang baru (Rahayu dkk., 2022).

Dalam konteks bisnis indekos, penerapan teknologi berbasis website untuk mendukung pengelolaan dan pelayanan penyewa masih belum banyak dikembangkan di Indonesia. Menurut Suheri, Sulaeman, & Widaningsih (2024), sebagian besar pemilik kos masih menggunakan metode konvensional untuk periklanan, pencatatan keuangan, dan komunikasi dengan penyewa. Sering kali terdapat potensi ketidakakuratan data dan proses administrasi yang memakan waktu (Sumiko, 2022). Dengan melihat potensi besar pengguna internet di Indonesia, pengembangan aplikasi berbasis website menjadi solusi strategis untuk meningkatkan efisiensi manajemen indekos. Aplikasi ini dapat mendukung digitalisasi layanan, mulai dari pencatatan pembayaran hingga peningkatan pengalaman pengguna (*user experience*) secara keseluruhan.

Sistem informasi rumah indekos berbasis website menawarkan keuntungan yang banyak bagi penyewa dan pemilik properti. Pengembangan aplikasi berbasis website menawarkan beberapa keuntungan dalam berbagai domain. Hal ini memberikan aksesibilitas di berbagai platform, memungkinkan pengguna untuk mengakses aplikasi melalui browser website (Febriana & Iswari, 2023; Mutmainnah Ismail dkk., 2022). Dengan adanya sistem ini, proses pencarian dan pemesanan akomodasi menjadi lebih mudah karena pengguna dapat memperoleh informasi rinci mengenai spesifikasi kamar, harga, dan fasilitas yang tersedia (Wahyudi, 2023). Bagi pemilik properti, platform berbasis website memungkinkan pemasaran dan promosi yang lebih efektif untuk persewaan mereka, melampaui metode tradisional seperti brosur cetak (Suheri dkk., 2024). Selain itu, pengelolaan operasional indekos menjadi lebih efisien, mulai dari manajemen data penyewa hingga pencatatan pembayaran, yang semuanya dapat diintegrasikan dalam satu platform berbasis website. Dengan keunggulan-keunggulan tersebut, aplikasi berbasis website dapat menjadi solusi yang efektif untuk meningkatkan manajemen dan pelayanan indekos. Bagi pemilik, sistem ini memberikan kemudahan operasional, sementara bagi penyewa, sistem ini menawarkan kenyamanan dan kepercayaan dalam proses pencarian dan pengelolaan pembayaran.

Penelitian ini merupakan kelanjutan dari tahap perancangan yang telah dilakukan sebelumnya. Penelitian awal oleh Fauzi (2024) telah berhasil merumuskan analisis kebutuhan pengguna dan menghasilkan rancangan antarmuka (*user interface*) serta pengalaman pengguna (*user experience*). Berbeda dengan fokus pada aspek desain tersebut, penelitian ini mengambil langkah selanjutnya dengan berfokus pada aspek rekayasa perangkat lunak, yaitu menerjemahkan desain tersebut menjadi sebuah prototipe aplikasi berbasis website yang fungsional. Untuk mencapai tujuan tersebut secara terstruktur, penelitian ini mengadopsi model pengembangan perangkat lunak *Waterfall*. Metode ini dipilih karena sangat sesuai dengan kondisi proyek di mana kebutuhan fungsional telah didefinisikan secara jelas pada penelitian sebelumnya, sehingga memungkinkan proses pengembangan yang sistematis dan terkontrol.

Tahapan pengembangan akan dilakukan sesuai dengan model *Waterfall*. Dimulai dari implementasi desain ke dalam kode program, dilanjutkan dengan pengujian (*testing*) fungsional untuk memastikan setiap fitur berjalan sesuai spesifikasi dan bebas dari kesalahan. Diharapkan, hasil dari penelitian ini adalah sebuah prototipe aplikasi website Nestify yang efektif dan teruji, yang tidak hanya menjadi solusi untuk meningkatkan efisiensi operasional pemilik kos, tetapi juga mempermudah pengalaman penyewa dalam mengelola indekos, serta

memberikan kontribusi positif dalam mempercepat transformasi digital di sektor manajemen properti.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang ada di atas, penulis dapat merumuskan beberapa rumusan masalah penelitian sebagai berikut:

a. Bagaimana setiap tahap dalam model *Waterfall*, mulai dari analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian (*testing),* dapat diterapkan untuk mengembangkan prototipe aplikasi website Nestify secara terstruktur dan sesuai dengan desain yang telah ada.
b. Bagaimana fungsionalitas utama aplikasi website Nestify dapat diuji secara komprehensif menggunakan metode *Black-box testing* dengan teknik *Boundary Value Analysis* (BVA) untuk memverifikasi kesesuaiannya dengan kebutuhan pengguna dan memastikan keandalan sistem.

1.3 Batasan Masalah

Dalam penelitian ini, penulis menetapkan beberapa batasan untuk mengarahkan pengembangan aplikasi dan metode evaluasi yang digunakan. Adapun hal tersebut dilakukan agar masalah yang dihadapi tidak meluas dan pembahasan tetap terarah. Batasan tersebut meliputi:

a. Hasil dari penelitian ini berupa prototipe fungsional aplikasi berbasis website responsif yang dirancang untuk manajemen indekos bagi pemilik dan penyewa.
b. Evaluasi aplikasi dilakukan dengan metode pengujian *black-box testing* yang berfokus pada pengujian fungsionalitas aplikasi, yang mencakup validasi input dan output, alur kerja pengguna, dan keandalan fitur-fitur utama.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengembangkan dan menguji prototipe aplikasi website Nestify yang dirancang khusus untuk manajemen kos. Secara khusus, penelitian ini bertujuan untuk:

a. Menerapkan setiap tahap dalam model Waterfall untuk menghasilkan aplikasi dengan antarmuka yang mudah digunakan, sehingga mendukung kemudahan interaksi serta pemenuhan kebutuhan pengguna.

b. Menguji fungsionalitas utama aplikasi menggunakan metode *Black-box Testing* dengan teknik *Boundary Value Analysis (BVA)* guna memastikan sistem dapat membantu pemilik kos mengelola properti secara efisien serta memudahkan penyewa dalam memantau status sewa.

1.5 Manfaat Penelitian

Penelitian ini berfokus pada pengembangan aplikasi berbasis website responsif untuk mendukung manajemen indekos. Penelitian diharapkan dapat memberikan manfaat sebagai berikut:

a. Penelitian ini memberikan solusi praktis berupa *prototype* dan aplikasi berbasis website yang dapat mempermudah pengelolaan indekos bagi pemilik properti, serta menyediakan kemudahan bagi penyewa dalam mengakses informasi sewa, memantau status pembayaran, dan mengajukan keluhan secara mudah dalam 1 platform website.
b. Penelitian ini berkontribusi pada pengembangan literatur di bidang teknologi informasi, khususnya seputar pembuatan aplikasi website responsif untuk manajemen properti. Dalam pengembangannya, penelitian ini menggunakan model *Waterfall* dan menerapkan metode pengujian *Black-box* pada sistem informasi manajemen properti.

1.6 Metodologi Penelitian

Penelitian ini mengimplementasikan metode *Research and Development* (R&D) berbasis model *Waterfall.* Secara umum, model *Waterfall* yang diterapkan dalam penelitian ini terdiri dari lima tahapan utama, yaitu:

a. *Requirements Analysis*

   Tahap awal ini berfokus pada pengumpulan, pemahaman, dan pendokumentasian kebutuhan pengguna serta batasan sistem. Tujuannya adalah untuk mendefinisikan secara jelas apa saja yang harus dapat dilakukan oleh perangkat lunak yang akan dikembangkan.

b. *System Design*

   Berdasarkan kebutuhan yang telah diidentifikasi, tahap ini melibatkan perancangan arsitektur perangkat lunak, struktur data, antarmuka (*interface*), dan algoritma. Hasil dari tahap ini adalah cetak biru (*blueprint*) sistem yang akan menjadi panduan bagi tahap implementasi.

c. *Implementation/Coding*

Pada tahap ini, desain sistem diterjemahkan ke dalam bahasa pemrograman yang dapat dimengerti oleh komputer. Pengembang menulis kode untuk setiap unit atau modul perangkat lunak sesuai dengan spesifikasi desain yang telah dibuat.

d. *Testing/Verification*

Setelah kode ditulis, tahap pengujian dilakukan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi kebutuhan, bebas dari kesalahan (*error* atau *bug*), dan berjalan dengan baik. Pengujian ini melibatkan verifikasi fungsionalitas dan pencarian potensi masalah dalam sistem.

e. *Deployment and Maintenance*

Tahap terakhir ini mencakup proses pemasangan (*deployment*) perangkat lunak di lingkungan operasional agar dapat digunakan oleh pengguna akhir. Setelah penerapan, aktivitas pemeliharaan dilakukan untuk memperbaiki kesalahan yang ditemukan setelah implementasi, melakukan penyesuaian, atau menambahkan peningkatan minor pada perangkat lunak seiring waktu.

1.7 Sistematika Penulisan

Berikut merupakan sistematika penulisan penelitian ini:

a. Bab 1: Pendahuluan
b. Bab 2: Kajian Pustaka
c. Bab 3: Metodologi Penelitian
d. Bab 4: Hasil dan Pembahasan
e. Bab 5: Kesimpulan dan Saran

BAB II
KAJIAN PUSTAKA

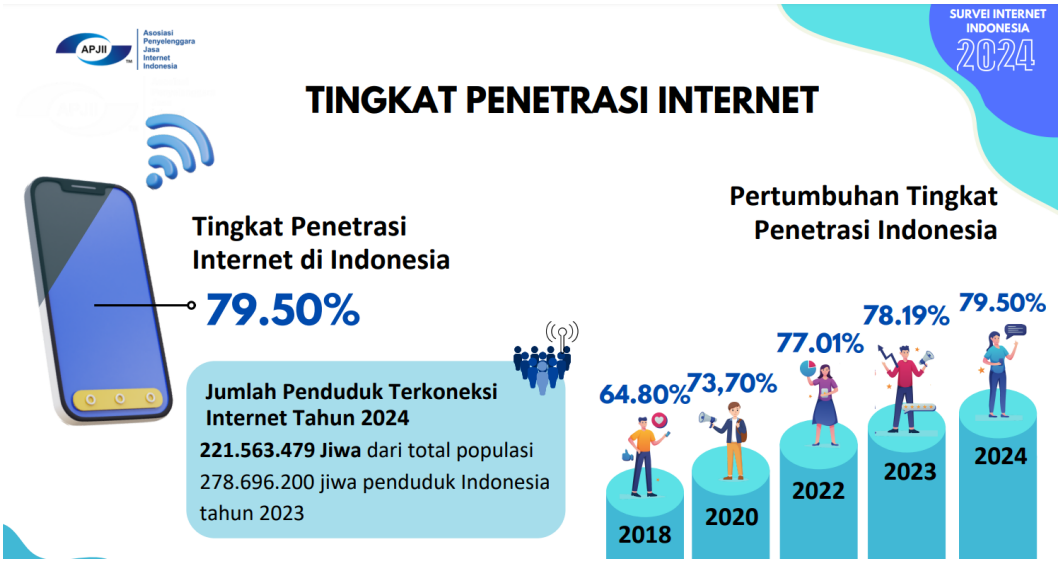
2.1 Pengembangan Aplikasi Berbasis Website

Aplikasi berbasis website telah menjadi pendekatan utama dalam membangun sistem informasi modern, karena memungkinkan pengguna mengakses layanan dan informasi melalui peramban (browser) tanpa perlu mengunduh dan memasang perangkat lunak khusus. Keunggulan utamanya mencakup minimnya biaya instalasi bagi pengguna, kemudahan bagi pengembang untuk merilis pembaruan fitur secara otomatis, serta aksesibilitas universal dari berbagai perangkat yang terhubung ke internet (Widya dkk., 2019).

Namun, tanpa fondasi yang kuat, pengembangan aplikasi website akan menghadapi berbagai kendala. Salah satu kekurangan aplikasi website adalah kesulitan dalam pengujian karena penggunaan teknologi server dan browser, yang membuatnya rentan terhadap kesalahan dan berpotensi menimbulkan masalah keandalan yang serius. Kesalahan pada aplikasi website berdampak pada kerugian finansial, penurunan pendapatan, dan hilangnya kredibilitas (Garousi dkk., 2013).

Dalam pengembangan sistem informasi, kualitas aplikasi memegang peran krusial. Sistem yang baik harus mampu mengakomodasi kebutuhan dan harapan penggunanya. Oleh karena itu, penting untuk mengukur bagaimana kemudahan penggunaan (*usability*), kualitas informasi (*information quality*), dan kualitas interaksi (*interaction quality*) memengaruhi pengalaman pengguna (Widodo dkk., 2019). Kecepatan pemuatan website sangat memengaruhi pengalaman pengguna, terutama dalam bisnis daring yang kompetitif seperti manajemen indekos. Website yang cepat memberikan keuntungan berupa peningkatan pengunjung, yang pada akhirnya berpengaruh positif pada jumlah pelanggan dan transaksi (Widya dkk., 2019).

Seiring dengan laju perkembangan teknologi dan perubahan selera pasar, para pengembang aplikasi menghadapi tren dan tantangan yang terus berevolusi. Berdasarkan data dari APJII (2024), jumlah penduduk Indonesia yang aktif menggunakan internet pada tahun 2024 mencapai lebih dari 220 juta jiwa, yang setara dengan lebih dari 70% populasi. Gambar 2.1 merupakan perkembangan Tingkat penetrasi internet di Indonesia 2024



Gambar 2.1 Data Tingkat penetrasi internet di Indonesia 2024

Sumber: APJII (Asosiasi Penyelenggara Jasa Internet Indonesia) 2024

Dengan maraknya aplikasi website, pengembang menghadapi tantangan untuk memperkuat aspek keamanan aplikasi tersebut. Ancaman peretasan terhadap aplikasi website masih terus ada. Beberapa serangan yang paling sering dieksploitasi terhadap aplikasi berbasis website antara lain adalah *SQL Injection, Phishing*, dan *Cross-Site Scripting* (XSS) (Yudha dkk., 2018). Pengembang website juga menghadapi tantangan akibat tuntutan klien yang beragam dan perkembangan teknologi yang pesat. Beberapa tantangan utama meliputi masalah desain, aksesibilitas website, dan kompatibilitas *browser* (Qayyum, 2012). Desainer menghadapi tantangan untuk memastikan website kompatibel dan berfungsi baik di berbagai browser yang ada. Ini mencakup pengujian menyeluruh dan aksesibilitas global melalui berbagai perangkat dan browser, serta menjamin fungsionalitas fitur (Saidu dkk., 2015).

Meskipun penelitian sebelumnya oleh Fauzi (2024)berfokus pada perancangan antarmuka untuk aplikasi mobile, penelitian ini secara strategis memilih untuk mengimplementasikan Nestify sebagai aplikasi berbasis website. Keputusan ini didasarkan pada beberapa pertimbangan utama. Pertama, platform website menawarkan aksesibilitas lintas perangkat yang lebih baik, memungkinkan baik pemilik maupun penyewa kos untuk mengelola aktivitas mereka melalui komputer, tablet, maupun *smartphone*. Hal ini sangat relevan karena pemilik kos mungkin lebih nyaman melakukan tugas administratif melalui layar yang lebih besar,

sementara penyewa membutuhkan fleksibilitas akses dari perangkat mana pun. Kedua, dari sisi pengembangan dan pemeliharaan, website memungkinkan pembaruan dan integrasi sistem dilakukan secara terpusat, yang secara signifikan mengurangi kompleksitas dan biaya operasional (Reddy dkk., 2024). Ketiga, pemanfaatan infrastruktur komputasi awan (*cloud computing*) menyediakan solusi yang skalabel dan efisien. Dengan model seperti *Software as a Service* (SaaS), aplikasi dapat diakses secara daring tanpa instalasi, sementara sumber daya server dapat disesuaikan dengan mudah untuk mengakomodasi pertumbuhan pengguna di masa depan, yang pada akhirnya meningkatkan efisiensi biaya dan keandalan sistem(Joe, 2023; Berenbrok dkk., 2022).

2.2 Indekos

Indekos (lebih dikenal dengan sebutan kos), merupakan sebuah tempat yang menyediakan jasa akomodasi sementara berupa beberapa kamar, di mana setiap kamar telah dilengkapi dengan fasilitas yang disediakan oleh pihak pemilik (Akbar dkk., 2022). Kata "kos" diadopsi dari bahasa Belanda, tepatnya dari frasa "*in de kos*". Terjemahan harfiahnya adalah "bersantap di dalam", tetapi maknanya meluas menjadi "bertempat tinggal dan makan" di kediaman orang lain (Triansah dkk., 2016). Seiring dengan meningkatnya mobilitas masyarakat di wilayah perkotaan yang padat dengan pusat pendidikan dan perkantoran, bisnis indekos menawarkan prospek yang sangat menarik dan terus bertumbuh secara signifikan di Indonesia (M.Ak dkk., 2021).

2.2.1 Tinjauan Pertumbuhan dan Celah Digitalisasi di Pasar Indekos

Pada masa kini, keberadaan fasilitas indekos di Indonesia menunjukkan peningkatan yang signifikan di berbagai daerah, mulai dari pusat-pusat kota hingga daerah-daerah kecil. Kondisi ini dipengaruhi oleh lonjakan peminat yang semakin tinggi. Fenomena ini juga diperkuat oleh data dari Rumah.com (2021) dalam Edityorini (2022), yang menunjukkan bahwa pada periode April-Juni 2021, aktivitas transaksi sewa indekos di Indonesia mengalami peningkatan luar biasa, mencapai 170% jika dibandingkan dengan periode Januari-Maret, dengan kenaikan jumlah *booking* sebesar 68%.

Dalam konteks pertumbuhan ini, digitalisasi berperan krusial, sebagaimana dicontohkan oleh Mamikos. Sebagai aplikasi pencari kos nomor satu di Indonesia, Mamikos telah berhasil mendominasi pasar dengan lebih dari 2 juta kamar kos yang tersebar di lebih dari 140 kota (Asih, 2021), dan terhubung dengan 6 juta penyewa setiap bulannya melalui platform

digitalnya (Damar, 2021). Penelitian oleh (Edityorini, 2022) secara spesifik mengkaji bagaimana intensitas mengakses akun *Instagram* @mamikosapp memengaruhi pemenuhan kebutuhan informasi *followers* mengenai rekomendasi kos. Temuan ini menegaskan bahwa platform digital seperti Instagram, yang digunakan oleh Mamikos, menjadi sarana vital bagi calon penyewa untuk mendapatkan informasi kos secara cepat dan mudah, serta bagi pemilik untuk memperluas jangkauan pemasaran.

Namun, keberhasilan platform pencarian ini sekaligus menyoroti adanya celah fungsional di pasar. Sebagian besar pemilik kos masih kembali mengandalkan metode konvensional untuk operasional internal setelah penyewa masuk. Proses manajemen seperti pencatatan data penghuni, pelacakan pembayaran, dan penanganan keluhan sering kali masih dilakukan secara manual menggunakan buku catatan atau spreadsheet (Suheri dkk., 2024). Metode konvensional ini memiliki sejumlah kelemahan yang dapat menghambat efisiensi, seperti risiko ketidakakuratan data, proses administrasi yang memakan waktu, dan komunikasi yang tidak terstruktur. Celah antara proses pencarian yang sudah digital dan proses manajemen yang masih manual inilah yang menjadi latar belakang utama pengembangan aplikasi Nestify.

2.2.2 Klasifikasi Indekos dan Implikasinya pada Desain Sistem

Untuk dapat merancang sistem yang relevan, penting untuk memahami keragaman produk di pasar indekos. Berdasarkan Ompusunggu, Ginting Munthe, Purba, & Sembiring (2022), indekos dapat dibedakan berdasarkan gender dan tipe kos berdasarkan harganya. Jenis kos berdasarkan gender terbagi menjadi tiga:

1. Kos putri, eksklusif untuk perempuan, umumnya dengan aturan jam malam yang ketat dan seringkali berintegrasi dengan rumah pemilik.
2. Kos putra, khusus diperuntukkan bagi laki-laki.
3. Kos campuran, dihuni oleh laki-laki dan perempuan, umumnya karyawan. Kos ini biasanya menerapkan aturan jam malam yang lebih fleksibel dan memperbolehkan penghuni memegang kunci gerbang sendiri.

Sedangkan pengelompokan tipe indekos berdasarkan harga menghasilkan beberapa kategori:

1. Tipe A (Eksklusif): Menawarkan fasilitas lengkap seperti tempat tidur, lemari, kamar mandi dalam, AC, dan TV. Harga sewanya paling tinggi.
2. Tipe B (Menengah): Menyediakan fasilitas standar dengan tambahan kamar mandi dalam dan terkadang token listrik pribadi. Harga sewanya menengah.

3. Tipe C (Standar): Hanya menyediakan fasilitas dasar seperti kasur dan lemari, dengan kamar mandi dan fasilitas lain yang digunakan bersama. Harga sewanya paling rendah.

Pemahaman terhadap klasifikasi ini menjadi krusial dalam perancangan sistem Nestify. Aplikasi harus memiliki arsitektur yang fleksibel untuk memungkinkan pemilik properti mengelola berbagai jenis kos dan tipe kamar yang berbeda dalam satu platform. Kemampuan sistem untuk mengakomodasi keragaman ini akan menentukan tingkat adopsi dan relevansi Nestify di pasar yang sesungguhnya.

2.2.3 Pemanfaatan Aplikasi Website sebagai Solusi Manajemen Indekos

Untuk menjawab tantangan operasional yang dihadapi pemilik kos, pemanfaatan aplikasi berbasis website menawarkan solusi yang efektif dan efisien. Berbeda dengan platform pencarian yang berfokus pada pemasaran, aplikasi manajemen seperti Nestify dirancang untuk mengotomatisasi proses bisnis internal. Pemilihan platform website didasari oleh kemudahan akses dari berbagai perangkat, efisiensi biaya pengembangan, serta kemudahan dalam pembaruan sistem secara terpusat (Brawijaya dkk., 2017).

Aplikasi website yang terintegrasi dapat secara langsung mengatasi kelemahan metode konvensional dengan menyediakan beberapa keunggulan fungsional (Abdul Jalil dkk., 2023):

a. Manajemen Data Terpusat

   Mengurangi risiko ketidakakuratan data dengan menyimpan informasi penghuni, kamar, dan riwayat transaksi secara terstruktur dalam sebuah basis data yang terorganisasi.

b. Otomatisasi Keuangan

   Memfasilitasi pembuatan tagihan otomatis, pemantauan pembayaran secara *real-time*, dan pelacakan tunggakan, sehingga meringankan beban administrasi pemilik dan meningkatkan efisiensi penagihan.

c. Komunikasi Terstruktur

   Menyediakan kanal khusus untuk pengajuan keluhan atau permintaan layanan, sehingga komunikasi antara pemilik dan penyewa menjadi lebih terdokumentasi, mudah dilacak, dan dapat dianalisis untuk perbaikan layanan di masa depan.

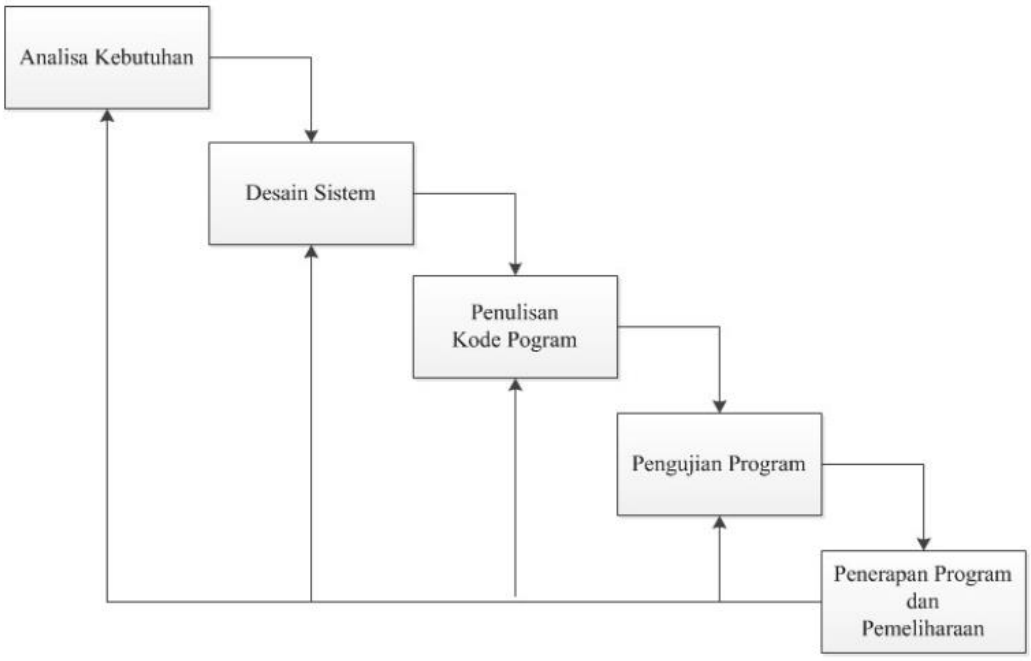
d. Peningkatan Transparansi

Memberikan akses bagi penyewa untuk melihat riwayat tagihan dan status pembayaran mereka, serta bagi pemilik untuk menyajikan laporan keuangan yang akurat, sehingga meningkatkan kepercayaan dan mengurangi potensi kesalahpahaman.

Dengan demikian, pengembangan aplikasi website Nestify dalam penelitian ini merupakan upaya untuk mengisi celah fungsional di pasar, dengan menyediakan sebuah alat bantu (*tool*) yang berfokus pada efisiensi, akurasi, dan transparansi dalam manajemen operasional indekos.

2.3 Model Pengembangan Perangkat Lunak *Waterfall*

Model *Waterfall* merupakan metode pengembangan perangkat lunak yang tahapannya berurutan, seperti air terjun yang mengalir dari atas ke bawah, di mana setiap fase harus diselesaikan untuk menghasilkan perangkat lunak yang sukses (Bassil, 2012). Model *Waterfall* menerapkan serangkaian fase yang saling bergantung dan harus diselesaikan secara berurutan. Perpindahan ke fase berikutnya hanya dimungkinkan setelah fase sebelumnya benar-benar tuntas. Karakteristik ini menjadikan model *Waterfall* bersifat rekursif, yang berarti setiap fase dapat diulang-ulang hingga mencapai hasil yang diinginkan (Bassil, 2012). Gambar 2.2 merupakan tahapan fase-fase dari metode *Waterfall*.



Gambar 2.2 Metode *Waterfall*

Sumber: (Kadir, 2014)

Pada dasarnya, model *Waterfall* terdiri dari lima fase; analisis, desain, implementasi atau penulisan, pengujian, dan pemeliharaan (Kadir, 2014). Penjelasan dari setiap fase tersebut;

a. Analisis Kebutuhan

Tahapan analisis kebutuhan sistem ini melibatkan pengumpulan data melalui penelitian, wawancara, atau studi literatur. Analis sistem bertugas mengumpulkan informasi sebanyak mungkin dari pengguna untuk mewujudkan sistem komputer yang sesuai dengan keinginan mereka. Hasil dari tahapan ini berupa dokumen *user requirement*, yang berisi data-data terkait kebutuhan pengguna dalam perancangan sistem. Dokumen inilah yang selanjutnya digunakan oleh analis sistem sebagai pedoman untuk menerjemahkan kebutuhan tersebut ke dalam kode pemrograman.

b. Desain Sistem

Dalam tahapan ini, gagasan-gagasan diolah dan sistem dirancang untuk menyelesaikan permasalahan yang dihadapi. Proses ini menggunakan perangkat pemodelan sistem, termasuk diagram alir data (*data flow diagram*), diagram relasi entitas (*entity relationship diagram*), serta struktur dan analisis data.

c. Penulisan Kode Program

Penulisan kode program atau implementasi coding melibatkan penerjemahan desain ke dalam bahasa yang dapat dimengerti oleh mesin. Programmer memegang peranan penting dalam menerjemahkan transaksi yang diminta pengguna ke dalam kode. Tahap ini merupakan tahapan inti dalam pengembangan sistem, di mana komputer digunakan secara intensif. Setelah proses coding rampung, pengujian (testing) dilakukan untuk mengidentifikasi dan memperbaiki potensi kesalahan pada sistem.

d. Pengujian Program

Pada tahap final, sistem baru diuji secara menyeluruh untuk mengungkap potensi kekurangan dan kelemahan dalam kemampuan dan efektivitasnya. Hasil pengujian ini kemudian digunakan sebagai dasar untuk pengkajian ulang dan perbaikan, dengan tujuan menyempurnakan aplikasi.

e. Penerapan Program dan Pemeliharaan

Perangkat lunak yang sudah digunakan oleh pelanggan pasti akan mengalami pembaruan. Pembaruan tersebut bisa berupa perbaikan kesalahan, adaptasi terhadap

lingkungan baru (perangkat keras atau sistem operasi yang berbeda), atau pengembangan fungsionalitas berdasarkan kebutuhan pengguna.

2.3.1 Justifikasi Pemilihan Model *Waterfall* untuk Nestify

Pemilihan model *Waterfall* untuk proyek aplikasi Nestify didasarkan pada beberapa pertimbangan strategis yang sangat relevan dengan konteks penelitian ini. Karena penelitian ini dibangun di atas fondasi analisis kebutuhan dan perancangan antarmuka yang telah dilakukan secara komprehensif oleh (Fauzi, 2024), maka persyaratan fungsional sistem telah terdefinisi dengan jelas. Kondisi ini sangat sesuai dengan keunggulan model *Waterfall* yang efektif untuk proyek dengan kebutuhan yang detail dan dapat diuji (Kramer, 2018). Pendekatan yang menuntut kejelasan di awal ini memungkinkan tim pengembang untuk fokus pada implementasi teknis tanpa terganggu oleh perubahan dinamis. Selain itu, sebagai sebuah penelitian akademik, kebutuhan akan dokumentasi yang terstruktur menjadi prioritas. Model *Waterfall*, dengan fokusnya yang kuat pada dokumentasi dan hasil akhir yang jelas di setiap fase, menyediakan kerangka kerja yang terorganisir dan terkontrol, yang sangat mendukung proses pelaporan dan evaluasi penelitian (Kramer, 2018).

2.3.2 Perbandingan dengan Metodologi *Agile*

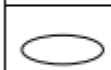
Sebagai perbandingan, metodologi populer lainnya seperti *Agile* menawarkan pendekatan yang lebih fleksibel, adaptif, dan berorientasi pada pelanggan (Al-Saqqa dkk., 2020). Metode *Agile* dengan berbagai kerangka kerjanya seperti *Scrum* atau *Kanban*, unggul dalam lingkungan proyek di mana kebutuhan pengguna terus berevolusi dan perubahan cepat perlu diakomodasi. Metode ini bekerja dalam siklus pendek yang disebut *sprint*, di mana tim menghasilkan bagian-bagian fungsional dari perangkat lunak secara berkala dan mendapatkan umpan balik langsung dari pengguna untuk perbaikan berkelanjutan (Daraojimba dkk., 2024).

Namun, untuk konteks pengembangan Nestify, pendekatan *Agile* dinilai kurang sesuai. Sifat proyek yang memiliki titik awal dan akhir yang jelas, dengan kebutuhan yang sudah ditentukan dari penelitian sebelumnya, tidak memerlukan fleksibilitas tinggi yang ditawarkan oleh *Agile*. Menggunakan *Waterfall* memastikan bahwa pengembangan tetap berada dalam jalur yang telah ditentukan, menghindari potensi penambahan fitur di luar ruang lingkup awal, dan menghasilkan produk akhir yang koheren sesuai dengan desain yang telah disetujui. Dengan demikian, *Waterfall* memberikan hasil dan kontrol yang lebih besar, yang merupakan faktor kunci keberhasilan untuk proyek Nestify.

2.4 *Entity Relationship Diagram*

Entity Relationship Diagram atau disingkat ERD, merupakan sebuah teknik yang dipergunakan pada tahapan awal perancangan basis data (Afiifah dkk., 2022). Para ahli berikut mendefinisikan ERD dari berbagai sudut pandang yang saling melengkapi. Joseph J. Frantiska Jr. menekankan ERD sebagai metodologi yang mendeskripsikan bagaimana data disimpan secara detail pada tingkat tinggi, dengan fokus pada struktur data, bukan fungsi sistem (Frantiska, 2018). Perspektif ini diperkuat oleh Syahidannur Mukminah Pulungan dkk., yang menyatakan ERD sebagai diagram utama yang merepresentasikan model data konseptual untuk mencerminkan kebutuhan data pengguna, menjadikannya tahap awal yang krusial dalam desain basis data (Pulungan dkk., 2023). Sementara itu, Qing Li dan Yuliu Chen melihatnya sebagai teknik umum untuk desain struktur data dan sistem basis data yang memperkenalkan sintaks dan semantik untuk pemodelan data (Li & Chen, 2009).

Sebagai salah satu teknik yang paling banyak diterapkan, ERD didasarkan pada model *entity relationship*. Data yang termuat dalam model tersebut direpresentasikan secara visual dalam bentuk ERD, yang menggambarkan relasi antar entitas dalam sebuah *database.*

Notasi	Keterangan
	Entitas, yaitu kumpulan dari objek yang dapat diidentifikasi secara unik.
	Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas.
	Atribut, yaitu karakteristik dari relasi yang merupakan penjelasan *detail* tentang entitas.
	Garis, hubungan antara *entity* dengan atributnya dan himpuna entitas dengan himpunan relasi.
	*Input* atau *output* data, Proses *input* atau *output* data, parameter, informasi.

Gambar 2.3 Simbol ERD

Sumber: (Rosa & Shalahuddin, 2013)

Dari penjelasan Gambar 2.3 di atas, dapat ditarik kesimpulan bahwa ERD adalah model konseptual yang memetakan relasi antar media penyimpanan data (Julianto, 2017).

2.5 *Prototype*

Prototype merupakan gambaran awal sebelum produksi yang mewakili beberapa bagian dari konsep atau desain final (Camburn dkk., 2017). Menurut pandangan Lauff, Kotys-Schwartz, & Rentschler (2018), *prototype* merupakan representasi fisik atau digital dari bagian-bagian penting sebuah desain, dan berfungsi sebagai alat iteratif untuk memperlancar komunikasi, memfasilitasi pembelajaran, dan memberikan informasi yang dibutuhkan dalam pengambilan keputusan di setiap tahapan proses desain. *Prototype* menjadi penting karena seringkali menentukan penggunaan sumber daya dalam pengembangan dan berdampak pada keberhasilan proyek desain (Camburn dkk., 2017). Dengan menggunakan *prototype*, representasi sederhana dari produk yang dikembangkan dapat diperoleh untuk keperluan eksplorasi, komunikasi, dan evaluasi ide, yang memberikan banyak keuntungan. Manfaat utama dari sisi bisnis terletak pada efisiensi biaya pengujian. Oleh karena itu, gagasan untuk "gagal lebih awal" dan secara berkelanjutan memikirkan kembali serta mengoptimalkan keputusan desain sebelum implementasi yang memakan biaya merupakan esensi dari pembuatan *prototype* (Christoforakos dkk., 2019).

2.6 *Black Box Testing*

Menurut Fatta dalam Setiyawati & Sardiarinto (2016), *black-box testing* merupakan proses pengujian melibatkan eksekusi unit sesuai alur sistem yang diinginkan. Apabila terdapat unit yang menghasilkan output yang salah atau tidak sesuai, maka akan dilakukan pengujian lebih lanjut menggunakan metode white box testing untuk menemukan dan memperbaiki kesalahan tersebut. Melengkapi pandangan yang berfokus pada alur sistem tersebut, Amalia, Putri Hamidah, & Kristanto, (2021) menyoroti bahwa pengujian *black-box* secara spesifik berfokus pada detail fungsional aplikasi seperti antarmuka dan *use case*, di mana pengujian dilakukan tanpa melihat kode program untuk memastikan fungsi yang dihasilkan sistem telah sesuai dengan spesifikasi yang dibutuhkan. Pendekatan *black-box* bersifat komplementer dan terbukti lebih ampuh dalam mengidentifikasi jenis kesalahan tertentu dibandingkan dengan metode *white-box*. Menurut Taufik & Ermawati (2017), pengujian *black-box* berupaya mendeteksi berbagai jenis kesalahan, antara lain:

1. Fungsi-fungsi yang tidak berjalan dengan benar atau bahkan tidak ada sama sekali (hilang). Ini berarti fitur-fitur yang seharusnya ada dan berfungsi, ternyata tidak bekerja sesuai harapan atau tidak tersedia sama sekali.

2. Kesalahan antarmuka (*interface*). Ini mencakup masalah pada tampilan pengguna (*user interface*/UI) atau interaksi antar modul atau sistem yang berbeda. Contohnya, kesalahan format tampilan, kesalahan navigasi, atau ketidaksesuaian data antar sistem.
3. Kesalahan dalam struktur data atau akses *database* eksternal. Hal ini merujuk pada kesalahan dalam cara data disimpan, diakses, atau dimanipulasi, termasuk masalah koneksi dan integritas data pada *database* eksternal.
4. Kesalahan kinerja (performance). Ini berkaitan dengan kecepatan, responsivitas, dan efisiensi sistem. Contohnya, waktu respon yang lambat, penggunaan memori yang berlebihan, atau *bottleneck* pada proses tertentu.
5. Inisialisasi dan kesalahan terminasi. Ini mencakup masalah pada saat sistem dimulai (inisialisasi) atau diakhiri (terminasi). Contohnya, kegagalan sistem untuk memulai dengan benar, kebocoran memori saat sistem ditutup, atau error yang terjadi saat proses *shutdown*.

2.7 Penelitian Terdahulu

Beberapa penelitian yang relevan dengan pengembangan aplikasi berbasis web untuk manajemen kos atau properti telah dilakukan sebelumnya. Penelitian-penelitian ini menjadi acuan dan pembanding dalam penyusunan sistem yang dikembangkan pada penelitian ini.

1. Suheri, Sulaeman, & Widaningsih (2024)

   Penelitian ini mengembangkan aplikasi manajemen kos berbasis web yang dilengkapi fitur pencarian lokasi kos menggunakan Google Maps API. Metode pengembangan yang digunakan adalah Waterfall, dan pengujian dilakukan dengan *Black Box Testing* untuk memastikan fungsi aplikasi berjalan sesuai kebutuhan. Hasilnya menunjukkan bahwa aplikasi dapat membantu calon penyewa menemukan lokasi kos secara lebih mudah serta memfasilitasi komunikasi antara pemilik kos dan penyewa melalui sistem daring.
2. Monteverde dkk. (2023)

   Penelitian ini merancang *Web-Based Rental House Smart Finder* menggunakan metode *Rapid Application Development* (RAD), yang ditujukan untuk mempermudah pencarian rumah sewa seperti kos, dormitori, dan hotel di daerah Bansud dan Bongabong, Filipina. Sistem ini dilengkapi integrasi Google Maps, fitur pemesanan daring, dan evaluasi menggunakan ISO/IEC 25010. Hasil evaluasi memperoleh skor rata-rata 3.67, yang menunjukkan tingkat penerimaan pengguna yang tinggi.

3. Mulya dkk. (2024)

   Penelitian ini mengembangkan aplikasi layanan kos berbasis web di area Universitas Ekasakti menggunakan metode SDLC Waterfall. Sistem dibangun dengan PHP dan MySQL, menyediakan informasi lokasi kos yang terhubung dengan Google Maps serta fitur penyewaan secara daring. Penelitian ini berhasil memberikan alternatif pencarian kos yang lebih praktis dibandingkan pencarian manual.

4. Ramanda dkk. (2025)

   Penelitian ini membangun sistem informasi manajemen rumah kos berbasis web yang berfokus pada pencatatan administrasi dan transaksi. Fitur yang disediakan mencakup manajemen data penghuni, pencatatan pembayaran, dan pembuatan laporan otomatis. Pengujian dilakukan untuk memastikan seluruh fungsi sistem berjalan sesuai kebutuhan pengguna.

Dari keempat penelitian tersebut, dapat diidentifikasi adanya beberapa persamaan dalam penggunaan teknologi berbasis web untuk mendukung pengelolaan atau pencarian kos. Namun, terdapat pula perbedaan yang cukup signifikan, baik dari sisi metode pengembangan, teknologi yang digunakan, maupun fokus fitur yang disediakan. Selain itu, masih terdapat celah penelitian yang belum terisi, khususnya dalam hal integrasi penuh antara manajemen data penghuni, pencatatan pembayaran, dan penentuan lokasi kos secara detail dalam satu platform web responsif. Ringkasan persamaan, perbedaan, dan celah penelitian tersebut disajikan pada Tabel berikut.

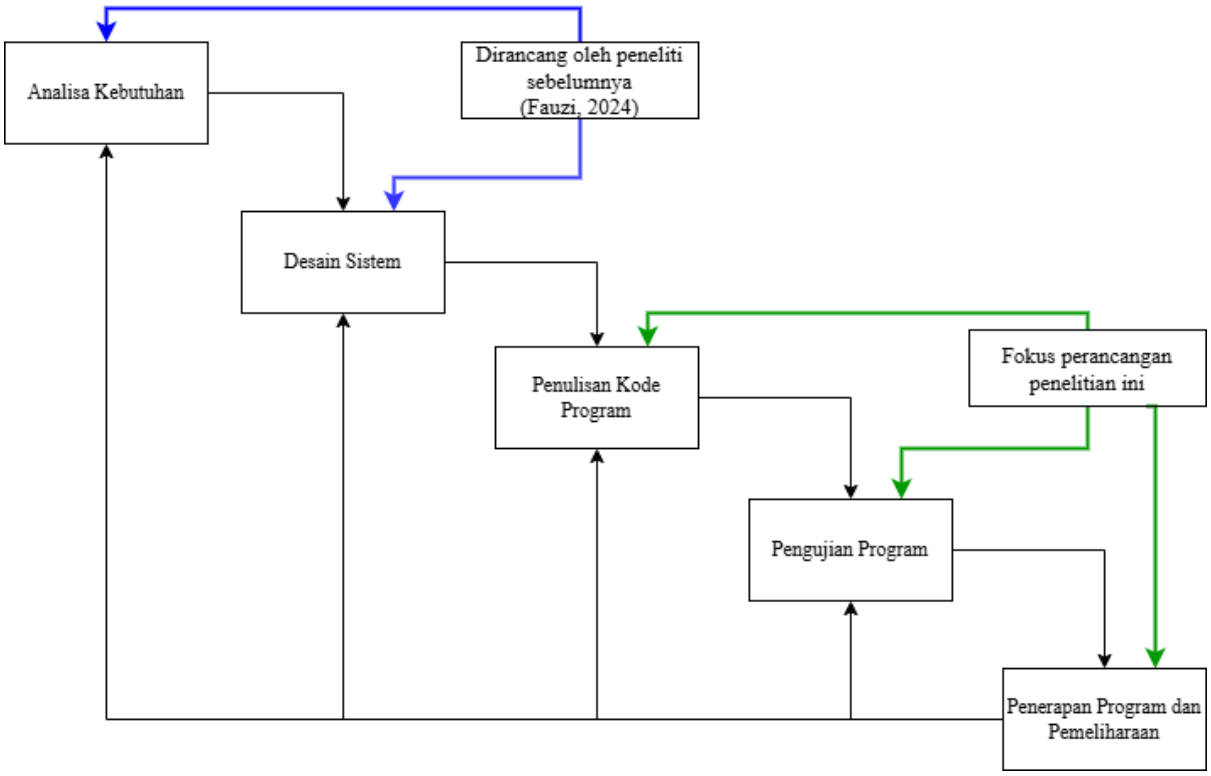
Aspek	**Penelitian**	**Detail**
Persamaan	Semua penelitian	Sama-sama berbasis web, membantu pencarian atau pengelolaan kos, dan memudahkan pengguna dalam mengakses informasi serta melakukan transaksi secara daring.
Perbedaan	Suheri, Sulaeman, & Widaningsih (2024)	Fokus pada pencarian kos dengan Google Maps API, metode Waterfall, pengujian *Black Box*. Tidak ada integrasi penuh untuk manajemen operasional kos.
Perbedaan	Monteverde dkk. (2023)	Fokus pada pencarian properti dan pemesanan daring, integrasi Google Maps, evaluasi ISO/IEC 25010, metode RAD. Tidak mengatur pengelolaan kos secara detail.
Perbedaan	Arta & Siregar (2024)	Pencarian kos berbasis web pada wilayah tertentu, metode Waterfall, PHP & MySQL. Tidak ada sistem pencatatan pembayaran otomatis.
Perbedaan	Ramanda (2022)	Fokus pada administrasi dan transaksi, tanpa integrasi peta atau sistem pencarian lokasi kos.
Celah Penelitian	Semua penelitian terdahulu	Belum ada yang menggabungkan manajemen data penghuni, pencatatan pembayaran, dan penentuan lokasi

		kos secara detail dalam satu platform web responsif. Penelitian ini mengisi celah tersebut dengan mengembangkan **Nestify**, sistem manajemen kos berbasis Laravel 12 dan Filament dengan pengujian *Boundary Value Analysis* (BVA).

Tabel 2.1 Persamaan, Perbedaan, dan Celah Penelitian

BAB III
METODOLOGI PENELITIAN

Penelitian sebelumnya oleh Fauzi (2024), melakukan pengembangan desain aplikasi *mobile* Nestify, yang mengadopsi pendekatan *User Centered Design* (UCD). Namun, penelitian ini berfokus pada implementasi aplikasi berbasis website dengan menggunakan metode *Research and Development* (R&D) dengan metode *Waterfall.* Penggunaan metode *Waterfall* didasarkan pada kebutuhan akan proses pengembangan yang sistematis dan terkendali, mengingat persyaratan aplikasi manajemen indekos yang telah didefinisikan dengan baik oleh penelitian sebelumnya. Metode *Waterfall* terdiri dari lima fase; analisis, desain, implementasi atau penulisan, pengujian, dan pemeliharaan.



Gambar 3.1 Bagan Kontribusi Penelitian

Sebagaimana ditunjukkan pada Gambar 3.1, penelitian ini dibangun di atas fondasi penelitian yang telah dilakukan oleh Fauzi (2024), yang telah berhasil menyelesaikan tahapan analisis kebutuhan dan desain sistem dalam kerangka metode Waterfall. Dengan demikian, penelitian ini akan melanjutkan proses pengembangan aplikasi dengan fokus pada

implementasi teknis melalui Penulisan Kode Program, verifikasi kualitas melalui Pengujian Program, implementasi di lingkungan pengguna melalui Penerapan Program, dan jaminan keberlanjutan melalui Pemeliharaan.

3.1 Analisis Kebutuhan

Fauzi (2024) dalam penelitiannya mengembangkan aplikasi Nestify dengan tujuan memenuhi kebutuhan pasar pengguna aplikasi manajemen indekos. Untuk mencapai tujuan tersebut, Fauzi menggunakan metode *User Centered Design* (UCD), yang secara khusus berorientasi pada kebutuhan, preferensi, dan perilaku pengguna sebagai tujuan utama dalam pengembangan desain produk atau layanan. Sebagai langkah awal dalam metode *Waterfall*, yaitu analisis kebutuhan, penelitian Fauzi (2024) mengadopsi tahapan *Understand Context of Use* yang bertujuan memahami konteks penggunaan aplikasi, dan *Specify User Requirements* yang berfokus pada perumusan kebutuhan pengguna secara spesifik, yang merupakan bagian dari metodologi UCD.

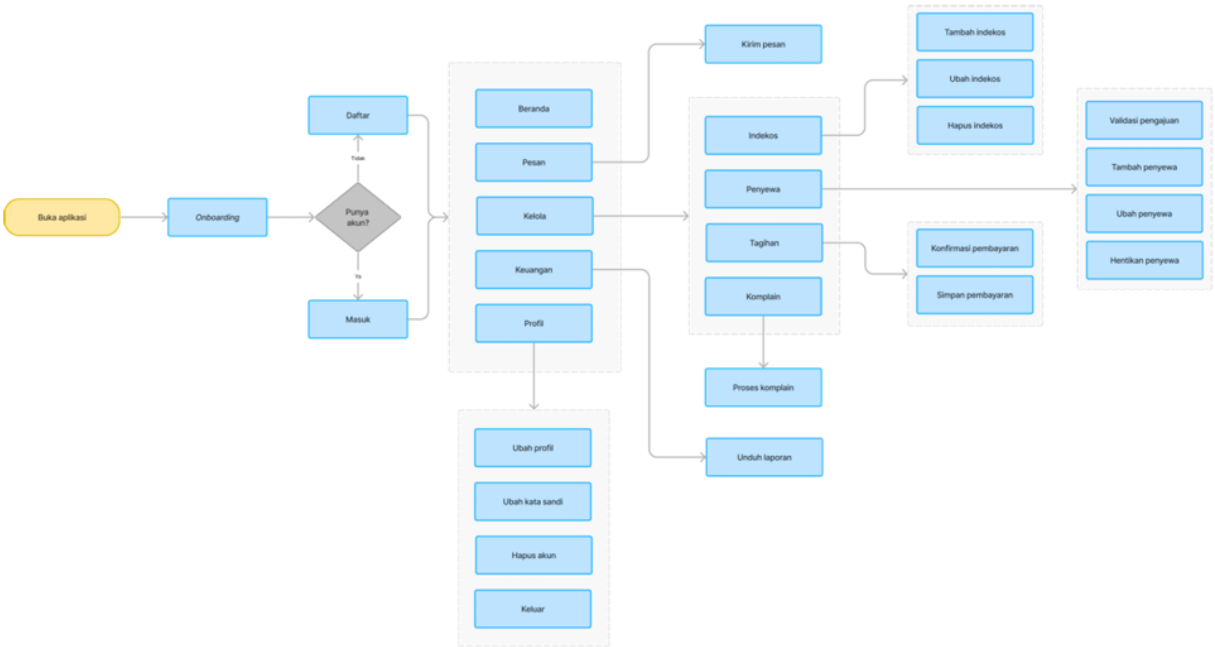
Dalam penelitian Fauzi (2024), telah dilakukan analisis konteks penggunaan aplikasi Nestify melalui identifikasi karakteristik pengguna potensial, yakni individu yang berusia di atas 17 tahun, memiliki pengalaman terkait indekos, dan mampu menggunakan teknologi. Data kualitatif dikumpulkan melalui wawancara semi-terstruktur dengan pemilik dan penyewa indekos yang mengeksplorasi aspek manajemen data, pelacakan pembayaran, komunikasi, dan penanganan keluhan. Informasi yang diperoleh disintesis menjadi *user persona* yang menggambarkan berbagai profil pengguna. Pemilik indekos ditampilkan mulai dari pengelola skala kecil yang membutuhkan sistem pencatatan keuangan sederhana hingga pengelola skala besar yang memerlukan fitur pelacakan pembayaran dan manajemen keluhan yang komprehensif. Sementara itu, penyewa indekos meliputi mahasiswi yang mengutamakan kenyamanan dan kebersihan, mahasiswa yang lebih memilih sistem pembayaran otomatis, serta mahasiswa dengan mobilitas tinggi yang menginginkan pencatatan pembayaran yang rapi. Pendekatan ini memberikan gambaran mendalam mengenai demografi, perilaku, tantangan, dan kebutuhan pengguna, sehingga aplikasi dapat dikembangkan secara tepat sasaran.

Selanjutnya dalam tahapan *Specify User Requirements*, Fauzi (2024) mengidentifikasi secara terperinci kebutuhan pengguna sebagai dasar klarifikasi persyaratan sistem aplikasi Nestify. Analisis dilakukan dengan memetakan *user needs* yang menggambarkan keinginan dan kebutuhan pengguna serta mengidentifikasi *pain points* untuk mengungkap masalah yang

mereka hadapi, dan merancang *user flow* yang menggambarkan alur interaksi pengguna dengan sistem. Berdasarkan hasil wawancara dan penyusunan *user persona*, kebutuhan spesifik bagi pemilik indekos meliputi fitur manajemen data yang efisien, kemampuan pencarian dan filter untuk pelacakan pembayaran, laporan pendapatan sederhana, serta pengelolaan komplain yang responsif. Sedangkan bagi penyewa, kebutuhan yang diidentifikasi mencakup komunikasi langsung dengan pemilik, pelaporan kerusakan yang mudah, pembayaran tagihan secara *online* yang aman, pencatatan riwayat tagihan, dan notifikasi otomatis sebagai pengingat jatuh tempo.

Pain points yang ditemukan menunjukkan bahwa pemilik sering mengalami kesulitan dalam pencatatan data, pelacakan pembayaran, penyusunan laporan keuangan, dan penanganan komplain, sementara penyewa menghadapi tantangan dalam komunikasi, pelaporan kerusakan, pengingat pembayaran, dan akses riwayat tagihan. Tahap *User flow* yang dikembangkan oleh (Fauzi, 2024) didasarkan pada analisis kebutuhan pengguna untuk mengatasi permasalahan calon pengguna aplikasi manajemen indekos, dengan pemetaan alur interaksi yang divisualisasikan pada Gambar 3.2 dan Gambar 3.3.

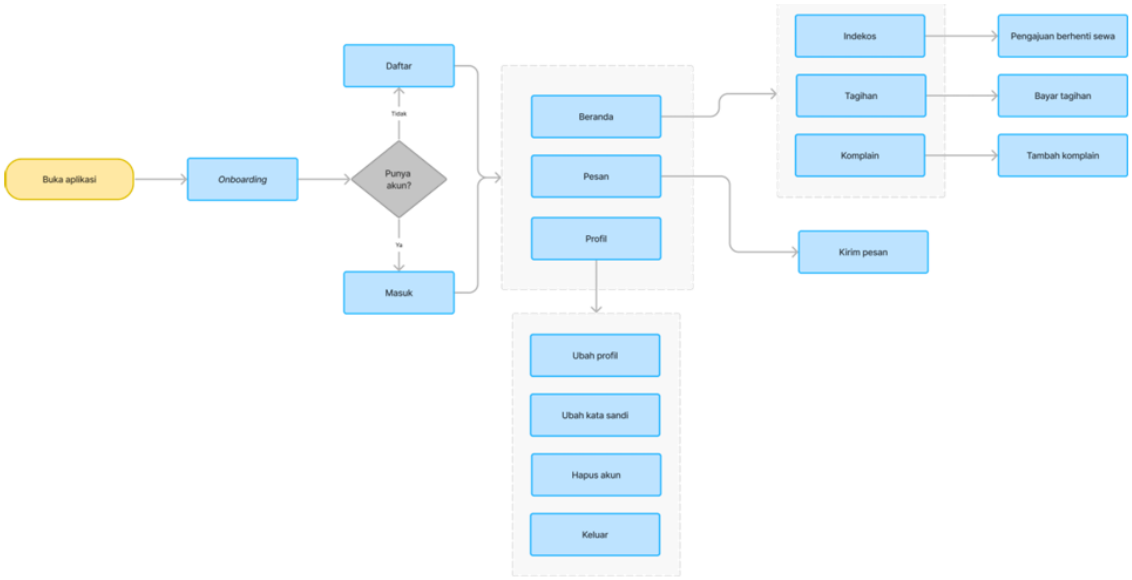
Pada Gambar 3.2, alur interaksi pemilik indekos dimulai dengan proses *login* menggunakan kredensial terdaftar, dilanjutkan dengan dashboard beranda yang menyajikan ringkasan aktivitas, notifikasi, dan akses cepat ke fitur utama seperti pengelolaan data indekos, penyewa, tagihan, komplain, serta pengaturan profil.



Gambar 3.2 *User Flow* Pemilik Indekos

Sumber: (Fauzi, 2024)

Sedangkan, Gambar 3.3 mengilustrasikan *user flow* penyewa indekos yang diawali dengan proses pendaftaran, diikuti dengan *login*, dan kemudian diarahkan ke dashboard yang menyediakan notifikasi serta akses ke fitur-fitur utama seperti tampilan rincian tagihan, pilihan metode pembayaran (otomatis atau manual), pengajuan komplain dengan deskripsi dan lampiran foto, serta pengelolaan informasi pribadi dan preferensi aplikasi.



Gambar 3.3 *User Flow* Penyewa Indekos

Sumber: (Fauzi, 2024)

Selain kebutuhan fungsional dan non-fungsional yang telah diidentifikasi oleh Fauzi (2024), terdapat celah analisis terkait pemilihan platform implementasi yang juga perlu diperhatikan. Meskipun desain awal aplikasi Nestify berfokus pada solusi *mobile*, terdapat alasan strategis untuk mempertimbangkan transisi ke aplikasi berbasis website. Platform website menawarkan akses lintas perangkat, sehingga pemilik dan penyewa indekos dapat mengakses sistem melalui komputer, tablet, maupun *smartphone* tanpa harus mengunduh aplikasi secara khusus. Selain itu, pengembangan berbasis website memungkinkan pemeliharaan, pembaruan, dan integrasi sistem dilakukan secara terpusat, yang dapat mengurangi biaya operasional dan kompleksitas pengembangan yang biasanya terjadi pada aplikasi *mobile* yang harus disesuaikan dengan berbagai sistem operasi.

3.2 Desain Sistem

Pada penelitian sebelumnya, Fauzi (2024) telah menyusun solusi desain yang mencakup perancangan *user flow*, *wireframe*, dan *prototype* untuk aplikasi berbasis *mobile* (Android).

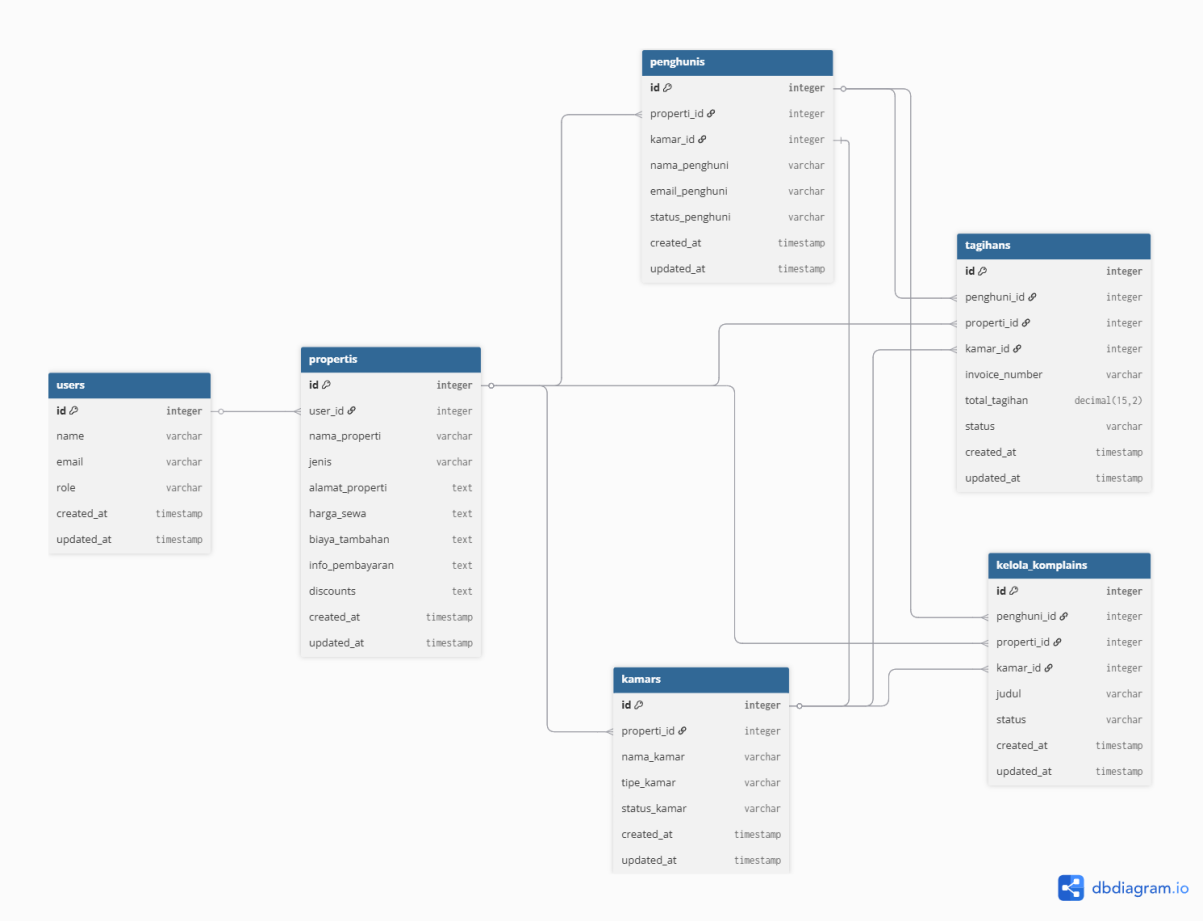
Hasil desain tersebut menjadi fondasi awal dalam memahami alur interaksi pengguna (*user interaction*) serta struktur antarmuka (*interface*) sistem.

Pada penelitian ini, penulis melakukan pengembangan lebih lanjut terhadap desain yang telah dirancang oleh Fauzi (2024) dengan melakukan adaptasi dari platform *mobile* ke platform berbasis website. Adaptasi ini mencakup penyesuaian tata letak (*layout*), navigasi, dan komponen antarmuka agar sesuai dengan karakteristik pengguna website, seperti responsivitas, skalabilitas, dan kompleksitas fungsional yang lebih tinggi. Selain itu, penulis juga melakukan evaluasi ulang terhadap *user flow* dan *wireframe* untuk memastikan konsistensi pengalaman pengguna (*user experience*) antar platform.

3.2.1 *Entity Relationship Diagram* (ERD)

Penelitian sebelumnya oleh Fauzi (2024) belum menyertakan perancangan *Entity Relationship Diagram* (ERD) sebagai representasi visual struktur *database*. Oleh karena itu, penulis juga menyusun ERD sesuai dengan kebutuhan fungsional sistem hasil adaptasi dari *mobile* ke website. ERD ini dirancang berdasarkan analisis *workflow* sistem, relasi antar-entitas data, serta kebutuhan *business process* yang tercermin dari hasil pengembangan desain antarmuka. Proses ini dilakukan dengan mengacu pada standar normalisasi *database* untuk memastikan integritas dan efisiensi struktur data.

Entity Relationship Diagram (ERD) pada Gambar 3.4 merupakan representasi visual dari struktur *database* aplikasi Nestify, yang dirancang untuk mendukung pengelolaan data yang efisien dan terintegrasi.



Gambar 3.4 *Entity Relationship Diagram* (ERD) Aplikasi Nestify

Perancangan ERD ini memastikan bahwa struktur *database* mendukung semua fungsionalitas aplikasi Nestify, mulai dari manajemen properti, kamar, dan penghuni, hingga pencatatan tagihan dan komplain, dengan integritas data yang optimal. ERD ini terdiri dari beberapa entitas utama dengan relasi antar-entitas sebagai berikut.

Entitas users

Merepresentasikan data pengguna yang terdaftar di aplikasi, baik pemilik kos maupun penyewa.

a. Atribut: id (*primary key*), name, email, role, created_at, updated_at.
b. Relasi: Berelasi satu-ke-banyak (*one-to-many*) dengan entitas propertis (satu pengguna dapat memiliki banyak properti).

Entitas propertis

Merepresentasikan data properti indekos yang dikelola oleh pemilik.

a. Atribut: id (*primary key*), user_id (*foreign key* ke users), nama_properti, jenis, alamat_properti, harga_sewa (JSON), biaya_tambahan (JSON), info_pembayaran (JSON), discounts (JSON), created_at, updated_at.
b. Relasi:
    - Berelasi satu-ke-banyak (*one-to-many*) dengan entitas kamars (satu properti dapat memiliki banyak kamar).
    - Berelasi satu-ke-banyak (*one-to-many*) dengan entitas penghunis (satu properti dapat memiliki banyak penghuni).
    - Berelasi satu-ke-banyak (*one-to-many*) dengan entitas kelola_komplains (komplain terkait properti).

Entitas kamars

Merepresentasikan data unit kamar dalam setiap properti.

a. Atribut: id (*primary key*), properti_id (*foreign key* ke propertis), nama_kamar, tipe_kamar, status_kamar, created_at, updated_at.
b. Relasi: Berelasi satu-ke-banyak (*one-to-many*) dengan entitas penghunis (satu kamar dapat ditempati oleh satu penghuni pada satu waktu).

Entitas penghunis

Merepresentasikan data penyewa yang terdaftar dan menempati kamar.

a. Atribut: id (*primary key*), properti_id (*foreign key* ke propertis), kamar_id (*foreign key* ke kamars), nama_penghuni, email_penghuni, status_penghuni, created_at, updated_at.
b. Relasi:
    - Berelasi satu-ke-banyak (*one-to-many*) dengan entitas tagihans (satu penghuni dapat memiliki banyak tagihan).
    - Berelasi satu-ke-banyak (*one-to-many*) dengan entitas kelola_komplains (satu penghuni dapat mengajukan banyak komplain).

Entitas tagihans

Merepresentasikan data tagihan sewa yang dibuat untuk penghuni.

a. Atribut: id (*primary key*), penghuni_id (*foreign key* ke penghunis), properti_id (*foreign key* ke propertis), kamar_id (*foreign key* ke kamars), invoice_number, total_tagihan, status, created_at, updated_at.

Entitas kelola_komplains

Merepresentasikan data komplain yang diajukan oleh penghuni.

a. Atribut: id (*primary key*), penghuni_id (*foreign key* ke penghunis), properti_id (*foreign key* ke propertis), kamar_id (*foreign key* ke kamars), judul, status, created_at, updated_at.

Entitas notifications

Merepresentasikan notifikasi sistem (status request berubah, tagihan dibuat/terbayar)

a. Atribut: id (UUID, primary key), type, notifiable_type, notifiable_id, data (JSON/string), read_at (opsional), created_at, updated_at.

3.2.2 *Wireframe*

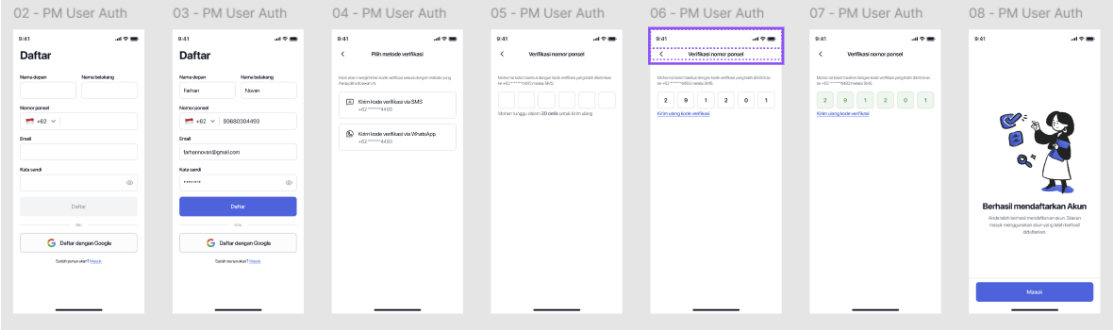
Penelitian ini mengadaptasi desain *wireframe* yang diacu dari Fauzi (2024), yang awalnya dirancang menggunakan aplikasi Figma untuk platform *mobile*. *Wireframe* ini berperan sebagai sketsa awal antarmuka yang memetakan struktur, tata letak, dan penempatan setiap elemen sebelum pembuatan *prototype* dilakukan. Proses perancangan ini dimaksudkan untuk menetapkan kerangka visual yang jelas, guna mendukung pengembangan sistem secara konsisten dan terstruktur.

Sebagai fondasi perancangan antarmuka aplikasi website Nestify, beberapa *wireframe* kunci dari penelitian Fauzi (2024) diadaptasi dan dijadikan acuan. Pemilihan *wireframe* ini berfokus pada alur interaksi utama pengguna yang relevan dengan fungsionalitas aplikasi website Nestify. Untuk detail *wireframe* lengkap dari penelitian Fauzi (2024), dapat merujuk langsung pada dokumen aslinya.

Berikut adalah *wireframe* utama dari penelitian Fauzi (2024) yang menjadi dasar adaptasi:

Wireframe Registrasi Pengguna

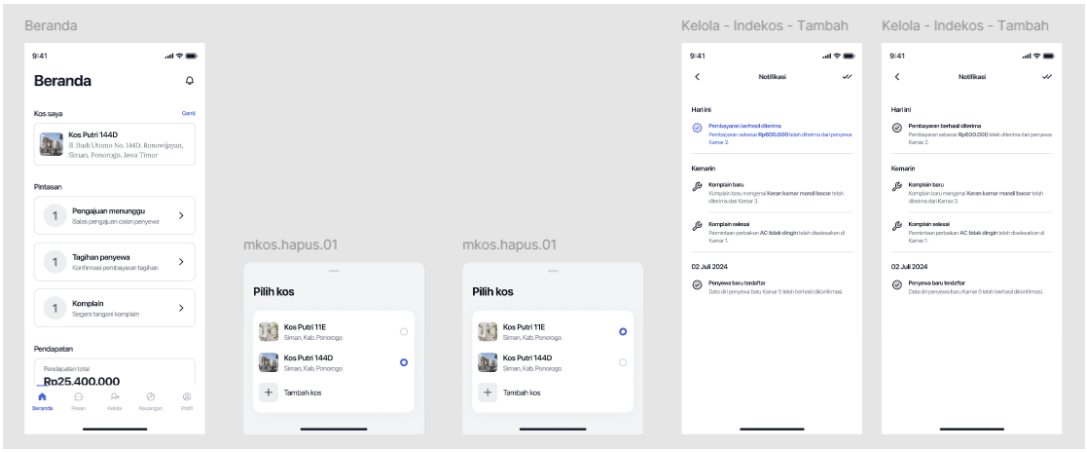
Gambar 3.5 menampilkan *wireframe* halaman registrasi pengguna dari penelitian Fauzi (2024). *Wireframe* ini menunjukkan alur pendaftaran akun baru, di mana pengguna diharuskan mengisi data diri seperti nama lengkap, nomor ponsel/email, dan kata sandi. Pemilihan peran pengguna (Pemilik kos atau Penyewa) merupakan fitur krusial yang perlu ditekankan, mengingat aspek ini belum diimplementasikan pada perancangan sebelumnya oleh Fauzi (2024), yang akan menentukan hak akses dan fungsionalitas yang tersedia setelah registrasi. Desain ini menjadi acuan untuk mengembangkan antarmuka registrasi pada aplikasi website Nestify, memastikan proses pembuatan akun yang terstruktur dan penentuan peran yang jelas.



Gambar 3.5 *Wireframe* Registrasi Pengguna oleh Fauzi (2024)

Wireframe Pemilik Kos

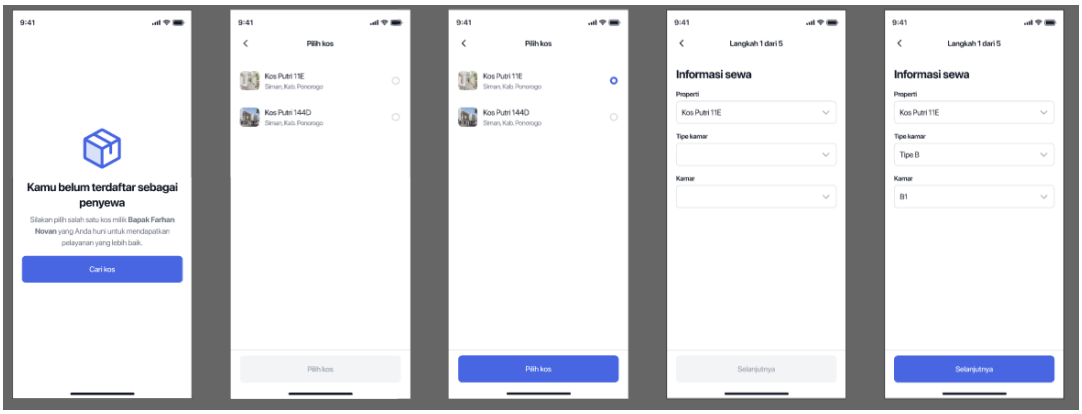
Gambar 3.6 menampilkan *wireframe* Dashboard Pemilik kos dari penelitian Fauzi (2024). *Wireframe* ini menyajikan gambaran umum tentang informasi penting dan pintasan fitur bagi pemilik, seperti ringkasan properti, pintasan manajemen penyewa, tagihan, dan komplain, serta visualisasi pendapatan. Desain ini menjadi inspirasi untuk menyusun tata letak dan prioritas informasi pada dashboard pemilik di aplikasi website Nestify. Penelitian Fauzi (2024) juga mencakup *wireframe* terperinci untuk modul-modul terkait dashboard ini, seperti pengelolaan properti (tambah, edit, hapus), manajemen unit kamar, pengelolaan data penyewa, dan detail keuangan, yang memberikan gambaran lengkap tentang fungsionalitas sisi pemilik.



Gambar 3.6 *Wireframe* Beranda Pemilik Kos oleh Fauzi (2024)

Wireframe Penyewa Kos

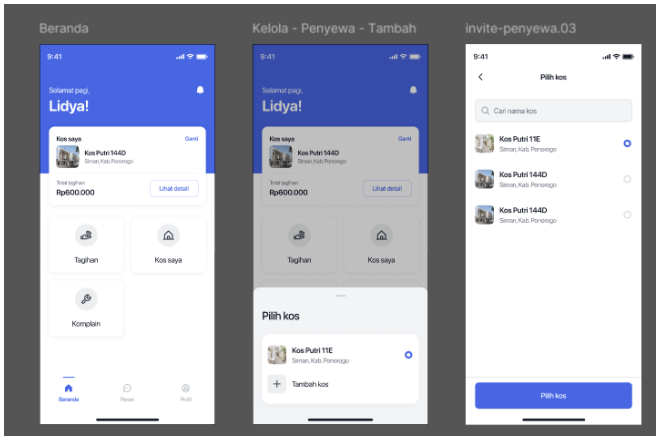
Gambar 3.7 menampilkan *wireframe* Dashboard Penyewa kos dari penelitian Fauzi (2024) pada kondisi awal ketika penyewa belum terdaftar di kos manapun. *Wireframe* ini mengilustrasikan tampilan utama bagi penyewa yang baru mengakses aplikasi atau belum memiliki kontrak sewa aktif, dengan penekanan pada ajakan untuk mencari dan mendaftar kos. Desain ini menjadi acuan untuk mengembangkan antarmuka dashboard penyewa yang informatif dan memandu pengguna baru di aplikasi website Nestify.



Gambar 3.7 *Wireframe* Beranda Penyewa oleh Fauzi (2024)

Gambar 3.8 menampilkan *wireframe* Dashboard Penyewa kos dari penelitian Fauzi (2024) setelah penyewa berhasil terdaftar di sebuah kos. *Wireframe* ini mengilustrasikan tampilan utama bagi penyewa yang sudah memiliki kontrak sewa aktif, menyajikan informasi kos yang ditempati, ringkasan tagihan, dan akses cepat ke fitur komplain. Desain ini menjadi acuan

untuk mengembangkan antarmuka dashboard penyewa yang informatif dan mudah diakses di aplikasi website Nestify, memberikan gambaran lengkap tentang status sewa dan keuangan mereka.



Gambar 3.8 *Wireframe* Beranda Penyewa oleh Fauzi (2024)

3.3 Penulisan Kode Program

Penulisan kode program dalam penelitian ini menggunakan *Visual Studio Code* (VSCode) sebagai *Integrated Development Environment* (IDE). Pengembangan aplikasi berbasis website dilakukan dengan memanfaatkan bahasa pemrograman PHP yang dikombinasikan dengan *framework* Laravel versi 12, dipilih karena kestabilannya dan kemampuannya untuk membangun aplikasi website yang aman serta mudah diskala. Dalam proses implementasinya, Laravel 12 diintegrasikan dengan Filament, sebuah paket *framework* yang menyediakan berbagai komponen antarmuka pengguna (UI) serta alat bantu untuk manajemen *back-end* yang efisien. Filament berperan penting dalam mempercepat pengembangan, terutama dalam pembuatan panel administrasi yang responsif dan mudah dikustomisasi.

Untuk pengelolaan basis data, penelitian ini menggunakan MySQL sebagai sistem manajemen basis data relasional (RDBMS). Lingkungan pengembangan *database* dijalankan melalui platform XAMPP yang menyediakan PHPMyAdmin sebagai alat visual untuk mempermudah pembuatan, pemeliharaan, dan pengelolaan *database*.

3.4 Pengujian Program

Tahap pengujian program merupakan fase fundamental dalam siklus hidup pengembangan perangkat lunak (*Software Development Life Cycle* atau SDLC) yang bertujuan untuk memverifikasi dan memvalidasi fungsionalitas aplikasi website Nestify. Fokus utama dari tahap ini adalah untuk memastikan bahwa setiap fitur yang telah dikembangkan dapat berjalan

sesuai dengan spesifikasi kebutuhan pengguna, andal, serta untuk mendeteksi dan mengidentifikasi potensi kesalahan (*bug*) sebelum aplikasi dirilis.

Sesuai dengan batasan dan tujuan penelitian, metode pengujian utama yang akan digunakan adalah *Black-box Testing*. Pendekatan ini dipilih karena evaluasi dilakukan berdasarkan spesifikasi fungsional dan perilaku eksternal aplikasi, tanpa memerlukan pengetahuan tentang struktur kode internalnya. Untuk meningkatkan efektivitas dan efisiensi, pendekatan ini akan diimplementasikan dengan menggunakan teknik perancangan kasus uji yang spesifik, yaitu *Boundary Value Analysis* (BVA). Teknik ini dipilih karena kemampuannya dalam mengoptimalkan pemilihan kasus uji, mengurangi redundansi, dan memfokuskan pengujian pada area-area yang paling rentan terhadap kesalahan, seperti pada nilai-nilai batas *input*.

3.4.1 Strategi dan Lingkungan Pengujian

Strategi pengujian dirancang secara sistematis untuk memastikan proses evaluasi fungsionalitas website Nestify berjalan secara terstruktur dan komprehensif.

Analisis Kebutuhan Fungsional

Langkah awal dalam tahap pengujian program adalah melakukan analisis kebutuhan fungsional untuk menentukan cakupan pengujian. Analisis ini dilakukan dengan meninjau kembali alur pengguna (*user flow*) dan spesifikasi kebutuhan dari studi sebelumnya, dengan fokus pada modul-modul yang paling fundamental dan memiliki risiko kegagalan tertinggi.

Untuk sisi Pemilik kos, cakupan pengujian difokuskan pada fungsionalitas inti yang meliputi Registrasi dan Login Akun, Kelola Properti, Data Penghuni, dan Manajemen Tagihan. Modul-modul ini dipilih karena secara kolektif merepresentasikan alur kerja yang paling krusial: autentikasi pengguna sebagai gerbang keamanan sistem, operasi dasar CRUD (*Create, Read, Update, Delete*) sebagai pola dasar manajemen data, serta manajemen tagihan sebagai jantung dari proses bisnis transaksional aplikasi.

Untuk memberikan evaluasi yang seimbang, pengujian juga mencakup fungsionalitas utama dari sisi Penyewa kos, yaitu Pendaftaran Sewa, Pembayaran Tagihan, dan Pengajuan Komplain. Ketiga modul ini dipilih karena merepresentasikan siklus hidup utama seorang penyewa, mulai dari proses pendaftaran, pemenuhan kewajiban finansial, hingga komunikasi terkait layanan. Dengan memvalidasi modul-modul representatif ini, dapat diperoleh tingkat

kepercayaan yang tinggi terhadap stabilitas dan keandalan fungsional sistem secara keseluruhan.

Perancangan Kasus Uji (*Test Case*)

Setelah ruang lingkup pengujian ditetapkan, langkah selanjutnya adalah merancang kasus uji secara detail. Proses perancangan ini dimulai dengan mengembangkan skenario uji (*test scenario*). Selanjutnya, akan disusun matriks kasus uji untuk memetakan berbagai kondisi pengujian beserta hasil yang diharapkan (*expected result*). Untuk merancang nilai data uji (*test data*), akan diimplementasikan teknik *Boundary Value Analysis* (BVA) untuk secara spesifik menguji nilai-nilai pada batas minimum dan maksimum dari suatu *field* masukan.

Pelaksanaan dan Pembahasan Analisis Kinerja

Tahap akhir dari strategi ini adalah pelaksanaan pengujian dan analisis terhadap hasilnya. Setiap kasus uji yang telah dirancang akan dieksekusi pada aplikasi website Nestify. Hasil observasi dari setiap eksekusi akan dicatat dan dibandingkan dengan hasil yang diharapkan untuk menentukan status pengujian, yaitu 'Berhasil' (*Pass*) atau 'Gagal' (*Fail*). Kinerja fungsionalitas sistem secara keseluruhan kemudian akan dianalisis secara kuantitatif dengan menghitung persentase keberhasilan menggunakan formula: N = (SP/SM) x 100%.

Lingkungan Pengujian

Proses pengujian akan dilaksanakan dalam lingkungan perangkat keras dan lunak yang terkontrol. Spesifikasi yang akan digunakan dalam pengujian ini adalah sebagai berikut:

- **Perangkat Keras**: Laptop ASUS Vivobook S14 M5406UA.
- **Sistem Operasi**: Microsoft Windows 11 Home Single Language (10.0.22631).
- **Peramban Website**: Google Chrome versi 136.0.7103.93 (64-bit).

3.4.2 Perancangan *Test Case*

Tahap ini merupakan penjabaran teknis dari strategi perancangan pengujian yang akan dilaksanakan. Perancangan dilakukan secara sistematis untuk memastikan setiap aspek fungsional dari modul yang diuji dapat dievaluasi secara menyeluruh. Proses perancangan ini mencakup beberapa langkah sebagai berikut:

a. Desain Nilai Uji *Boundary Value Analysis* (BVA)

Untuk memastikan cakupan pengujian yang efektif, perancangan nilai data uji (*test data*) akan dilakukan secara sistematis untuk setiap *field* masukan dengan mengimplementasikan *Boundary Value Analysis* (BVA). Teknik ini akan diterapkan untuk merancang kasus uji yang secara spesifik menargetkan nilai-nilai di batas (tepi) dari sebuah partisi data, seperti nilai minimum, sedikit di atas minimum, maksimum, dan sedikit di atas maksimum. Pendekatan ini sangat penting karena pengalaman dalam rekayasa perangkat lunak menunjukkan bahwa kesalahan sering kali terjadi pada nilai-nilai batas tersebut.

Skenario Validasi & *Boundary Value Analysis* Modul Registrasi

Tabel 3.1 menjelaskan kriteria validasi untuk setiap *field input* pada halaman Registrasi, mencakup nama lengkap, email, kata sandi, konfirmasi kata sandi, dan pilihan peran, termasuk batasan panjang, format, dan keunikan data.

ID Validasi	Data Input	Deskripsi Kriteria Validasi
VAL_REG_PK_001	Nama Lengkap	Nama Lengkap tidak boleh kosong.
VAL_REG_PK_002	Nama Lengkap	Panjang Nama Lengkap memiliki batas.
VAL_REG_PK_003	Nama Lengkap	Nama Lengkap hanya boleh mengandung huruf dan spasi.
VAL_REG_PK_004	Email	Email tidak boleh kosong.
VAL_REG_PK_005	Email	Email harus memiliki format yang valid.
VAL_REG_PK_006	Email	Panjang email memiliki batas.
VAL_REG_PK_007	Email	Email yang dimasukkan harus unik dan belum pernah dibuat di *database*.
VAL_REG_PK_008	Kata Sandi	Kata Sandi tidak boleh kosong.
VAL_REG_PK_009	Kata Sandi	Kata Sandi memiliki batas minimal karakter.
VAL_REG_PK_010	Kata Sandi	Kata Sandi memiliki batas maksimal karakter.
VAL_REG_PK_011	Kata Sandi	Kata Sandi harus memenuhi kriteria kompleksitas.
VAL_REG_PK_012	Konfirmasi Kata Sandi	Konfirmasi Kata Sandi tidak boleh kosong.
VAL_REG_PK_013	Konfirmasi Kata Sandi	Isi Konfirmasi Kata Sandi harus sama dengan isi Kata Sandi.
VAL_REG_PK_014	Pilihan Peran (*Role*)	Pemilihan peran harus dilakukan jika ingin masuk ke sistem.

Tabel 3.1 Cek Validasi Registrasi

Tabel 3.2 menyajikan nilai-nilai uji spesifik untuk *field* nama lengkap, email, dan kata sandi pada halaman Registrasi, yang difokuskan pada batas bawah (BLB, LB, ALB) dan batas atas (BUB, UB, AUB) berdasarkan kriteria validasi yang telah ditetapkan.

No.	**Data Input**	**Aturan Batas Terkait**	**Nilai Uji**	**Valid / Invalid**	**Keterangan**
1	Nama Lengkap	Minimal 3 karakter	An	Invalid	BLB (*Below Lower Bound*)
2	Nama Lengkap	Minimal 3 karakter	And	Valid	LB (*Lower Bound*)
3	Nama Lengkap	Minimal 3 karakter	Andi	Valid	ALB (*Above Lower Bound*)
4	Nama Lengkap	Maksimal 100 karakter	Andikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarak	Valid	BUB (*Below Upper Bound*)
5	Nama Lengkap	Maksimal 100 karakter	Andikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarak a	Valid	UB (*Upper Bound*)
6	Nama Lengkap	Maksimal 100 karakter	Andikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarakaandikarak aa	Invalid	AUB (*Above Upper Bound*)
7	Email	Minimal 6 karakter (VAL_REG_PK_006)	andik@gmail.com	Invalid	BLB (*Below Lower Bound*)
8	Email	Minimal 6 karakter (VAL_REG_PK_006)	andika@gmail.com	Valid	LB (*Lower Bound*)
9	Email	Minimal 6 karakter (VAL_REG_PK_006)	andikar@gmail.com	Valid	ALB (*Above Lower Bound*)

10	Email	Maksimal 100 karakter (VAL_REG_PK_006)	Andikaraka6969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969@gmail.com	Valid	BUB (*Below Upper Bound*)
11	Email	Maksimal 100 karakter (VAL_REG_PK_006)	Andikaraka6969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969@gmail.com	Valid	UB (*Upper Bound*)
12	Email	Maksimal 100 karakter (VAL_REG_PK_006)	Andikaraka6969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969696969@gmail.com	Invalid	AUB (*Above Upper Bound*)
13	Kata Sandi	Minimal 8 karakter (VAL_REG_PK_009)	Coba123	Invalid	BLB (*Below Lower Bound*)
14	Kata Sandi	Minimal 8 karakter (VAL_REG_PK_009)	Coba1234	Valid	LB (*Lower Bound*)
15	Kata Sandi	Minimal 8 karakter (VAL_REG_PK_009)	Coba12345	Valid	ALB (*Above Lower Bound*)
16	Kata Sandi	Maksimal 32 karakter (VAL_REG_PK_010)	Coba1234567891011121314151617 18	Valid	BUB (*Below Upper Bound*)
17	Kata Sandi	Maksimal 32 karakter (VAL_REG_PK_010)	Coba1234567891011121314151617 181	Valid	UB (*Upper Bound*)
18	Kata Sandi	Maksimal 32 karakter	Coba1234567891011121314151617 1819	Invalid	AUB (*Above Upper Bound*)

		(VAL_REG_PK_0 10)			

Tabel 3.2 *Boundary Value Analysis* Registrasi

Skenario Validasi & *Boundary Value Analysis* Modul Login

Tabel 3.3 merinci kriteria validasi untuk setiap *field input* pada halaman *Login*, seperti email tidak boleh kosong, format email yang valid, panjang karakter email, dan kesesuaian kata sandi dengan data di basis data.

ID Validasi	**Data Input**	**Deskripsi Kriteria Validasi**
VAL_LOGIN_PK_001	Email	Email tidak boleh kosong.
VAL_LOGIN_PK_002	Email	Email harus memiliki format yang valid.
VAL_LOGIN_PK_003	Email	Panjang email memiliki batas minimal 6 dan maksimal 100 karakter.
VAL_LOGIN_PK_004	Email	Email yang dimasukkan harus sudah terdaftar di dalam basis data sistem Nestify.
VAL_LOGIN_PK_005	Kata Sandi	Kata Sandi tidak boleh kosong.
VAL_LOGIN_PK_006	Kata Sandi	Kata Sandi memiliki batas minimal 8 dan maksimal 32 karakter.
VAL_LOGIN_PK_007	Kata Sandi	Kata Sandi yang dimasukkan harus sesuai dengan kata sandi yang terasosiasi dengan email di basis data.

Tabel 3.3 Cek Validasi Login

Tabel 3.4 menyajikan nilai-nilai uji spesifik untuk *field* email dan kata sandi pada halaman *Login*, yang difokuskan pada batas bawah (BLB, LB, ALB) dan batas atas (BUB, UB, AUB) berdasarkan kriteria validasi yang telah ditetapkan.

No.	**Data Input**	**Aturan Batas Terkait**	**Nilai Uji**	**Valid / Invalid**	**Keterangan**
1	Email	Minimal 6 karakter	andik@gmail.com	Invalid	BLB (*Below Lower Bound*)
2	Email	Minimal 6 karakter	andika@gmail.com	Valid	LB (*Lower Bound*)

3	Email	Minimal 6 karakter	andikar@gmail.com	Valid	ALB (*Above Lower Bound*)
4	Email	Maksimal 100 karakter	Andikaraka6969696969696969696969696969696969696969696969696969696969696969696969696969696969@gmail.com	Valid	BUB (*Below Upper Bound*)
5	Email	Maksimal 100 karakter	Andikaraka696969696969696969696969696969696969696969696969696969696969696969696969696969696969@gmail.com	Valid	UB (*Upper Bound*)
6	Email	Maksimal 100 karakter	Andikaraka6969696969696969696969696969696969696969696969696969696969696969696969696969696969696@gmail.com	Invalid	AUB (*Above Upper Bound*)
7	Kata Sandi	Minimal 8 karakter	Coba123	Invalid	BLB (*Below Lower Bound*)
8	Kata Sandi	Minimal 8 karakter	Coba1234	Valid	LB (*Lower Bound*)
9	Kata Sandi	Minimal 8 karakter	Coba12345	Valid	ALB (*Above Lower Bound*)
10	Kata Sandi	Maksimal 32 karakter	Coba123456789101112131415161718	Valid	BUB (*Below Upper Bound*)
11	Kata Sandi	Maksimal 32 karakter	Coba1234567891011121314151617181	Valid	UB (*Upper Bound*)
12	Kata Sandi	Maksimal 32 karakter	Coba12345678910111213141516171819	Invalid	AUB (*Above Upper Bound*)

Tabel 3.4 Kelas Boundary Value Analysis Login

Skenario Validasi & *Boundary Value Analysis* Modul Kelola Properti

Tabel 3.13 menguraikan kriteria validasi untuk setiap *field input* pada modul Kelola Properti, seperti nama properti, jenis properti, alamat, foto properti (termasuk format dan ukuran), jumlah kamar, dan harga sewa.

ID Validasi	Data Input	Deskripsi Kriteria Validasi
VAL_PROPERTI_PK_001	Nama Properti	Nama Properti tidak boleh kosong.
VAL_PROPERTI_PK_002	Nama Properti	Panjang Nama Properti memiliki batas.
VAL_PROPERTI_PK_003	Jenis Properti	Jenis Properti harus dipilih dari opsi yang valid.
VAL_PROPERTI_PK_004	Alamat Properti	Alamat Properti tidak boleh kosong.
VAL_PROPERTI_PK_005	Alamat Properti	Detail Alamat harus valid jika ada pemisahan *field*.
VAL_PROPERTI_PK_006	Foto Properti	Minimal satu foto properti harus diunggah.
VAL_PROPERTI_PK_007	Foto Properti	Format *file* foto harus sesuai.
VAL_PROPERTI_PK_008	Foto Properti	Ukuran *file* foto tidak boleh melebihi batas maksimal yang ditentukan.
VAL_PROPERTI_PK_009	Jumlah Foto	Jumlah maksimal foto yang dapat diunggah per properti dibatasi 5.
VAL_PROPERTI_PK_010	Informasi Tambahan	Jika *field* Informasi Tambahan memiliki batasan panjang, maka harus divalidasi.
VAL_PROPERTI_PK_011	Harga Sewa	Harga Sewa harus berupa angka positif dan tidak boleh kosong.
VAL_PROPERTI_PK_012	Jumlah Kamar	Jumlah Kamar harus berupa angka bulat positif.

Tabel 3.5 Cek Validasi Kelola Properti

Tabel 3.14 menyajikan nilai-nilai uji spesifik untuk *field* nama properti, jumlah kamar, harga sewa, dan jumlah foto pada modul Kelola Properti, yang difokuskan pada batas bawah (BLB, LB, ALB) dan batas atas (BUB, UB, AUB) berdasarkan kriteria validasi yang telah ditetapkan.

No.	Data Input	Aturan Batas Terkait	Nilai Uji	Valid / Invalid	Keterangan
1	Nama Properti	Minimal 5 karakter (VAL_PROPERTI_PK_002)	Kire	Invalid	BLB (*Below Lower Bound*)
2	Nama Properti	Minimal 5 karakter (VAL_PROPERTI_PK_002)	Kirei	Valid	LB (*Lower Bound*)
3	Nama Properti	Minimal 5 karakter (VAL_PROPERTI_PK_002)	Kirein	Valid	ALB (*Above Lower Bound*)

4	Nama Properti	Maksimal 150 karakter (VAL_PROPERTI_PK_002)	Kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinapu	Valid	BUB (*Below Upper Bound*)
5	Nama Properti	Maksimal 150 karakter (VAL_PROPERTI_PK_002)	Kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaput	Valid	UB (*Upper Bound*)
6	Nama Properti	Maksimal 150 karakter (VAL_PROPERTI_PK_002)	Kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputrimajujaya kireinaputr	Invalid	AUB (*Above Upper Bound*)
7	Jumlah Kamar	Minimal 1 kamar	0	Invalid	BLB (*Below Lower Bound*)
8	Jumlah Kamar	Minimal 1 kamar	1	Valid	LB (*Lower Bound*)
9	Jumlah Kamar	Minimal 1 kamar	2	Valid	ALB (*Above Lower Bound*)
10	Jumlah Kamar	Maksimal 999 kamar	998	Valid	BUB (*Below Upper Bound*)
11	Jumlah Kamar	Maksimal 999 kamar	999	Valid	UB (*Upper Bound*)

12	Jumlah Kamar	Maksimal 999 kamar	1000	Invalid	AUB (*Above Upper Bound*)
13	Harga Sewa	Minimal Rp1.000	999	Invalid	BLB (*Below Lower Bound*)
14	Harga Sewa	Minimal Rp1.000	1000	Valid	LB (*Lower Bound*)
15	Harga Sewa	Minimal Rp1.000	1001	Valid	ALB (*Above Lower Bound*)
16	Harga Sewa	Maksimal Rp100.000.000	99999999	Valid	BUB (*Below Upper Bound*)
17	Harga Sewa	Maksimal Rp10.000.0000	100000000	Valid	UB (*Upper Bound*)
18	Harga Sewa	Maksimal Rp10.000.0000	100000001	Invalid	AUB (*Above Upper Bound*)
19	Jumlah Foto	Minimal 1 foto (untuk Tambah Baru)	0	Invalid	BLB (*Below Lower Bound*)
20	Jumlah Foto	Minimal 1 foto	1	Valid	LB (*Lower Bound*)
21	Jumlah Foto	Minimal 1 foto	2	Valid	ALB (*Above Lower Bound*)
22	Jumlah Foto	Maksimal 10 foto	9	Valid	BUB (*Below Upper Bound*)
23	Jumlah Foto	Maksimal 10 foto	10	Valid	UB (*Upper Bound*)
24	Jumlah Foto	Maksimal 10 foto	11	Invalid	AUB (*Above Upper Bound*)

Tabel 3.6 Kelas Boundary Value Analysis Kelola Properti

Skenario Validasi & *Boundary Value Analysis* Modul Data Penghuni

Tabel 3.15 mengidentifikasi kriteria validasi untuk setiap *field input* pada modul Data Penghuni, termasuk nama, alamat, pekerjaan, jenis kelamin, foto KTP, nomor HP, email, kontak darurat, properti, kamar, durasi sewa, total tagihan, dan tanggal.

ID Validasi	**Data Input**	**Deskripsi Kriteria Validasi**
VAL_DP_PK_001	Nama Penghuni	Nama Penghuni tidak boleh kosong.

VAL_DP_PK_002	Nama Penghuni	Panjang Nama Penghuni memiliki batas.
VAL_DP_PK_003	Nama Penghuni	Nama Penghuni hanya boleh mengandung huruf dan spasi.
VAL_DP_PK_004	Alamat Penghuni	Alamat Penghuni tidak boleh kosong.
VAL_DP_PK_005	Alamat Penghuni	Panjang Alamat Penghuni memiliki batas.
VAL_DP_PK_006	Pekerjaan Penghuni	Pekerjaan Penghuni tidak boleh kosong.
VAL_DP_PK_007	Jenis Kelamin Penghuni	Jenis Kelamin Penghuni harus dipilih dari opsi yang valid.
VAL_DP_PK_008	Foto KTP	Foto KTP harus diunggah.
VAL_DP_PK_009	Foto KTP	Format *file* foto KTP harus sesuai (JPG/PNG).
VAL_DP_PK_010	Foto KTP	Ukuran *file* foto KTP tidak boleh melebihi batas maksimal yang ditentukan.
VAL_DP_PK_011	Nomor HP Penghuni	Nomor HP Penghuni tidak boleh kosong.
VAL_DP_PK_012	Nomor HP Penghuni	Nomor HP Penghuni harus berupa angka.
VAL_DP_PK_013	Nomor HP Penghuni	Panjang Nomor HP Penghuni memiliki batas minimal 10 dan maksimal 15 digit.
VAL_DP_PK_014	Email Penghuni	Email Penghuni tidak boleh kosong.
VAL_DP_PK_015	Email Penghuni	Email Penghuni harus memiliki format yang valid.
VAL_DP_PK_016	Email Penghuni	Email Penghuni harus unik dan belum terdaftar.
VAL_DP_PK_017	Nama Kontak Darurat	Nama Kontak Darurat tidak boleh kosong.
VAL_DP_PK_018	Nomor HP Kontak Darurat	Nomor HP Kontak Darurat tidak boleh kosong.
VAL_DP_PK_019	Nomor HP Kontak Darurat	Nomor HP Kontak Darurat harus berupa angka.
VAL_DP_PK_020	Nomor HP Kontak Darurat	Panjang Nomor HP Kontak Darurat memiliki batas minimal 10 dan maksimal 15 digit.
VAL_DP_PK_021	Properti	Properti harus dipilih dari daftar properti yang dimiliki.
VAL_DP_PK_022	Kamar	Kamar harus dipilih dari daftar kamar yang tersedia di properti yang dipilih.
VAL_DP_PK_023	Durasi Sewa	Durasi Sewa harus berupa angka positif.
VAL_DP_PK_024	Satuan Durasi	Satuan Durasi harus dipilih dari opsi yang valid (hari, minggu, bulan, tahun).

VAL_DP_PK_025	Tanggal Mulai Sewa	Tanggal Mulai Sewa tidak boleh kosong dan harus dalam format tanggal yang valid.
VAL_DP_PK_026	Total Tagihan	Total Tagihan harus berupa angka positif.
VAL_DP_PK_027	Tanggal Jatuh Tempo	Tanggal Jatuh Tempo tidak boleh kosong dan harus dalam format tanggal yang valid.
VAL_DP_PK_028	Status Penghuni	Status Penghuni harus dipilih dari opsi yang valid (Aktif, Pengajuan, Tidak Aktif).

Tabel 3.7 Cek Validasi Data Penghuni

Tabel 3.16 menyajikan nilai-nilai uji spesifik untuk *field* nama penghuni, nomor HP penghuni, durasi sewa, dan total tagihan pada modul Data Penghuni, yang difokuskan pada batas bawah (BLB, LB, ALB) dan batas atas (BUB, UB, AUB) berdasarkan kriteria validasi yang telah ditetapkan.

No.	**Data Input**	**Aturan Batas Terkait**	**Nilai Uji**	**Valid / Invalid**	**Keterangan**
1	Nama Penghuni	Minimal 3 karakter (VAL_DP_PK_002)	Bu	Invalid	BLB (*Below Lower Bound*)
2	Nama Penghuni	Minimal 3 karakter (VAL_DP_PK_002)	Bud	Valid	LB (*Lower Bound*)
3	Nama Penghuni	Minimal 3 karakter (VAL_DP_PK_002)	Budi	Valid	ALB (*Above Lower Bound*)
4	Nama Penghuni	Maksimal 100 karakter (VAL_DP_PK_002)	Budiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosireg	Valid	BUB (*Below Upper Bound*)
5	Nama Penghuni	Maksimal 100 karakter (VAL_DP_PK_002)	Budiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosirega	Valid	UB (*Upper Bound*)

6	Nama Penghuni	Maksimal 100 karakter (VAL_DP_PK_002)	Budiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarbudiyonosiregarb	Invalid	AUB (*Above Upper Bound*)
7	Nomor HP Penghuni	Minimal 10 digit (VAL_DP_PK_013)	0812345678	Valid	LB (*Lower Bound*)
8	Nomor HP Penghuni	Minimal 10 digit (VAL_DP_PK_013)	081234567	Invalid	BLB (*Below Lower Bound*)
9	Nomor HP Penghuni	Minimal 10 digit (VAL_DP_PK_013)	08123456789	Valid	ALB (*Above Lower Bound*)
10	Nomor HP Penghuni	Maksimal 15 digit (VAL_DP_PK_013)	081234567890123	Valid	UB (*Upper Bound*)
11	Nomor HP Penghuni	Maksimal 15 digit (VAL_DP_PK_013)	08123456789012	Valid	BUB (*Below Upper Bound*)
12	Nomor HP Penghuni	Maksimal 15 digit (VAL_DP_PK_013)	0812345678901234	Invalid	AUB (*Above Upper Bound*)
13	Durasi Sewa	Minimal 1 (VAL_DP_PK_023)	0	Invalid	BLB (*Below Lower Bound*)
14	Durasi Sewa	Minimal 1 (VAL_DP_PK_023)	1	Valid	LB (*Lower Bound*)
15	Durasi Sewa	Minimal 1 (VAL_DP_PK_023)	2	Valid	ALB (*Above Lower Bound*)
16	Durasi Sewa	Maksimal 365 (VAL_DP_PK_023)	364	Valid	BUB (*Below Upper Bound*)
17	Durasi Sewa	Maksimal 365 (VAL_DP_PK_023)	365	Valid	UB (*Upper Bound*)
18	Durasi Sewa	Maksimal 365 (VAL_DP_PK_023)	366	Invalid	AUB (*Above Upper Bound*)
19	Total Tagihan	Minimal Rp1.000 (VAL_DP_PK_026)	999	Invalid	BLB (*Below Lower Bound*)
20	Total Tagihan	Minimal Rp1.000 (VAL_DP_PK_026)	1000	Valid	LB (*Lower Bound*)

21	Total Tagihan	Minimal Rp1.000 (VAL_DP_PK_026)	1001	Valid	ALB (*Above Lower Bound*)
22	Total Tagihan	Maksimal Rp100.000.000 (VAL_DP_PK_026)	99999999	Valid	BUB (*Below Upper Bound*)
23	Total Tagihan	Maksimal Rp100.000.000 (VAL_DP_PK_026)	100000000	Valid	UB (*Upper Bound*)
24	Total Tagihan	Maksimal Rp100.000.000 (VAL_DP_PK_026)	100000001	Invalid	AUB (*Above Upper Bound*)

Tabel 3.8 Kelas Boundary Value Analysis Data Penghuni

Skenario Validasi & *Boundary Value Analysis* Modul Manajemen Tagihan

Tabel 3.17 mengidentifikasi kriteria validasi untuk setiap *field input* pada modul Manajemen Tagihan, termasuk status tagihan, bukti pembayaran (format dan ukuran), tanggal jatuh tempo, nominal tagihan, periode tagihan, diskon, dan kode promo.

ID Validasi	Data Input	Deskripsi Kriteria Validasi
VAL_MT_PK_001	Status Tagihan	Status Tagihan harus valid (Belum Bayar, Butuh Konfirmasi, Lunas).
VAL_MT_PK_002	Bukti Pembayaran	Bukti Pembayaran harus diunggah jika status "Butuh Konfirmasi".
VAL_MT_PK_003	Bukti Pembayaran	Format *file* bukti pembayaran harus sesuai (JPG/PNG).
VAL_MT_PK_004	Bukti Pembayaran	Ukuran *file* bukti pembayaran tidak boleh melebihi batas maksimal yang ditentukan.
VAL_MT_PK_005	Tanggal Jatuh Tempo	Tanggal Jatuh Tempo tidak boleh kosong dan harus dalam format tanggal yang valid.
VAL_MT_PK_006	Nominal Tagihan	Nominal Tagihan harus berupa angka positif.
VAL_MT_PK_007	Periode Tagihan	Periode Tagihan tidak boleh kosong dan harus dalam format yang valid.
VAL_MT_PK_008	Diskon	Diskon harus berupa angka positif atau nol.
VAL_MT_PK_009	Kode Promo	Kode Promo harus valid dan belum kedaluwarsa jika digunakan.

Tabel 3. 9 Cek Validasi Manajemen Tagihan

Tabel 3.18 menyajikan nilai-nilai uji spesifik untuk *field* nominal tagihan dan diskon pada modul Manajemen Tagihan, yang difokuskan pada batas bawah (BLB, LB, ALB) dan batas atas (BUB, UB, AUB) berdasarkan kriteria validasi yang telah ditetapkan.

No.	**Data Input**	**Aturan Batas Terkait**	**Nilai Uji**	**Valid / Invalid**	**Keterangan**
1	Nominal Tagihan	Minimal Rp1.000 (VAL_MT_PK_006)	999	Invalid	BLB (*Below Lower Bound*)
2	Nominal Tagihan	Minimal Rp1.000 (VAL_MT_PK_006)	1000	Valid	LB (*Lower Bound*)
3	Nominal Tagihan	Minimal Rp1.000 (VAL_MT_PK_006)	1001	Valid	ALB (*Above Lower Bound*)
4	Nominal Tagihan	Maksimal Rp100.000.000 (VAL_MT_PK_006)	99999999	Valid	BUB (*Below Upper Bound*)
5	Nominal Tagihan	Maksimal Rp100.000.000 (VAL_MT_PK_006)	100000000	Valid	UB (*Upper Bound*)
6	Nominal Tagihan	Maksimal Rp100.000.000 (VAL_MT_PK_006)	100000001	Invalid	AUB (*Above Upper Bound*)
7	Diskon	Minimal 0 (VAL_MT_PK_008)	-1	Invalid	BLB (*Below Lower Bound*)
8	Diskon	Minimal 0 (VAL_MT_PK_008)	0	Valid	LB (*Lower Bound*)
9	Diskon	Minimal 0 (VAL_MT_PK_008)	1	Valid	ALB (*Above Lower Bound*)
10	Diskon	Maksimal 100% (VAL_MT_PK_008)	99	Valid	BUB (*Below Upper Bound*)
11	Diskon	Maksimal 100% (VAL_MT_PK_008)	100	Valid	UB (*Upper Bound*)
12	Diskon	Maksimal 100% (VAL_MT_PK_008)	101	Invalid	AUB (*Above Upper Bound*)

Tabel 3.10 Kelas Boundary Value Analysis Manajemen Tagihan

Skenario Validasi & *Boundary Value Analysis* Modul Pendaftaran Penyewa Kos

Tabel 3.19 mengidentifikasi kriteria validasi untuk setiap *field input* pada modul Pendaftaran Sewa, meliputi nama lengkap, email, nomor telepon/WA, jenis kelamin, pekerjaan/status, alamat asal, foto KTP, kamar, rencana tanggal masuk, durasi sewa, total tagihan, tanggal jatuh tempo, dan kontak darurat.

ID Validasi	Data Input	Deskripsi Kriteria Validasi
VAL_REG_PY_001	Nama Lengkap	Nama Lengkap tidak boleh kosong.
VAL_REG_PY_002	Nama Lengkap	Panjang Nama Lengkap memiliki batas.
VAL_REG_PY_003	Nama Lengkap	Nama Lengkap hanya boleh mengandung huruf dan spasi.
VAL_REG_PY_004	Email	Email tidak boleh kosong.
VAL_REG_PY_005	Email	Email harus memiliki format yang valid.
VAL_REG_PY_006	Email	Panjang email memiliki batas.
VAL_REG_PY_007	Email	Email yang dimasukkan harus unik dan belum pernah dibuat di *database*.
VAL_REG_PY_008	Nomor Telepon/WA	Nomor Telepon/WA tidak boleh kosong.
VAL_REG_PY_009	Nomor Telepon/WA	Nomor Telepon/WA harus berupa angka.
VAL_REG_PY_010	Nomor Telepon/WA	Panjang Nomor Telepon/WA memiliki batas minimal 10 dan maksimal 15 digit.
VAL_REG_PY_011	Jenis Kelamin	Jenis Kelamin harus dipilih dari opsi yang valid.
VAL_REG_PY_012	Pekerjaan/Status	Pekerjaan/Status tidak boleh kosong.
VAL_REG_PY_013	Alamat Asal (sesuai KTP)	Alamat Asal tidak boleh kosong.
VAL_REG_PY_014	Foto KTP	Foto KTP harus diunggah.
VAL_REG_PY_015	Foto KTP	Format *file* foto KTP harus sesuai (JPG/PNG).
VAL_REG_PY_016	Foto KTP	Ukuran *file* foto KTP tidak boleh melebihi batas maksimal yang ditentukan.
VAL_REG_PY_017	Kamar	Kamar harus dipilih dari daftar kamar yang tersedia di properti yang dipilih.
VAL_REG_PY_018	Rencana Tanggal Masuk	Rencana Tanggal Masuk tidak boleh kosong dan harus dalam format tanggal yang valid.
VAL_REG_PY_019	Durasi Sewa	Durasi Sewa harus berupa angka positif.
VAL_REG_PY_020	Satuan Durasi	Satuan Durasi harus dipilih dari opsi yang valid (hari, minggu, bulan, tahun).
VAL_REG_PY_021	Total Tagihan	Total Tagihan harus berupa angka positif.

VAL_REG_PY_022	Tanggal Jatuh Tempo	Tanggal Jatuh Tempo tidak boleh kosong dan harus dalam format tanggal yang valid.
VAL_REG_PY_023	Nama Kontak Darurat	Nama Kontak Darurat tidak boleh kosong.
VAL_REG_PY_024	Nomor HP Kontak Darurat	Nomor HP Kontak Darurat tidak boleh kosong.
VAL_REG_PY_025	Nomor HP Kontak Darurat	Nomor HP Kontak Darurat harus berupa angka.
VAL_REG_PY_026	Nomor HP Kontak Darurat	Panjang Nomor HP Kontak Darurat memiliki batas minimal 10 dan maksimal 15 digit.

Tabel 3.11 Cek Validasi Pendaftaran Sewa

Tabel 3.20 menyajikan nilai-nilai uji spesifik untuk *field* durasi sewa, total tagihan, nomor telepon/WA, dan nomor HP kontak darurat pada modul Pendaftaran Sewa, yang difokuskan pada batas bawah (BLB, LB, ALB) dan batas atas (BUB, UB, AUB) berdasarkan kriteria validasi yang telah ditetapkan.

No.	**Data Input**	**Aturan Batas Terkait**	**Nilai Uji**	**Valid / Invalid**	**Keterangan**
1	Durasi Sewa	Minimal 1 (VAL_REG_PY_019)	0	Invalid	BLB (*Below Lower Bound*)
2	Durasi Sewa	Minimal 1 (VAL_REG_PY_019)	1	Valid	LB (*Lower Bound*)
3	Durasi Sewa	Minimal 1 (VAL_REG_PY_019)	2	Valid	ALB (*Above Lower Bound*)
4	Durasi Sewa	Maksimal 365 (VAL_REG_PY_019)	364	Valid	BUB (*Below Upper Bound*)
5	Durasi Sewa	Maksimal 365 (VAL_REG_PY_019)	365	Valid	UB (*Upper Bound*)
6	Durasi Sewa	Maksimal 365 (VAL_REG_PY_019)	366	Invalid	AUB (*Above Upper Bound*)
7	Total Tagihan	Minimal Rp1.000 (VAL_REG_PY_021)	999	Invalid	BLB (*Below Lower Bound*)
8	Total Tagihan	Minimal Rp1.000 (VAL_REG_PY_021)	1000	Valid	LB (*Lower Bound*)

9	Total Tagihan	Minimal Rp1.000 (VAL_REG_PY_021)	1001	Valid	ALB (*Above Lower Bound*)
10	Total Tagihan	Maksimal Rp100.000.000 (VAL_REG_PY_021)	99999999	Valid	BUB (*Below Upper Bound*)
11	Total Tagihan	Maksimal Rp100.000.000 (VAL_REG_PY_021)	100000000	Valid	UB (*Upper Bound*)
12	Total Tagihan	Maksimal Rp100.000.000 (VAL_REG_PY_021)	100000001	Invalid	AUB (*Above Upper Bound*)
13	Nomor Telepon/WA	Minimal 10 digit (VAL_REG_PY_010)	0812345678	Valid	LB (*Lower Bound*)
14	Nomor Telepon/WA	Minimal 10 digit (VAL_REG_PY_010)	081234567	Invalid	BLB (*Below Lower Bound*)
15	Nomor Telepon/WA	Minimal 10 digit (VAL_REG_PY_010)	08123456789	Valid	ALB (*Above Lower Bound*)
16	Nomor Telepon/WA	Maksimal 15 digit (VAL_REG_PY_010)	081234567890123	Valid	UB (*Upper Bound*)
17	Nomor Telepon/WA	Maksimal 15 digit (VAL_REG_PY_010)	08123456789012	Valid	BUB (*Below Upper Bound*)
18	Nomor Telepon/WA	Maksimal 15 digit (VAL_REG_PY_010)	0812345678901234	Invalid	AUB (*Above Upper Bound*)
19	Nomor HP Kontak Darurat	Minimal 10 digit (VAL_REG_PY_026)	0812345678	Valid	LB (*Lower Bound*)
20	Nomor HP Kontak Darurat	Minimal 10 digit (VAL_REG_PY_026)	081234567	Invalid	BLB (*Below Lower Bound*)
21	Nomor HP Kontak Darurat	Minimal 10 digit (VAL_REG_PY_026)	08123456789	Valid	ALB (*Above Lower Bound*)
22	Nomor HP Kontak Darurat	Maksimal 15 digit (VAL_REG_PY_026)	081234567890123	Valid	UB (*Upper Bound*)

23	Nomor HP Kontak Darurat	Maksimal 15 digit (VAL_REG_PY_026)	08123456789012	Valid	BUB (*Below Upper Bound*)
24	Nomor HP Kontak Darurat	Maksimal 15 digit (VAL_REG_PY_026)	0812345678901234	Invalid	AUB (*Above Upper Bound*)

Tabel 3.12 Kelas Boundary Value Analysis Pendaftaran Sewa

Skenario Validasi & *Boundary Value Analysis* Modul Pembayaran Tagihan Penyewa

Tabel 3.21 mengidentifikasi kriteria validasi untuk setiap *field input* pada modul Pembayaran Tagihan, termasuk bukti pembayaran (harus diunggah, format, dan ukuran) serta kode promo (opsional, validitas, dan kedaluwarsa).

ID Validasi	**Data Input**	**Deskripsi Kriteria Validasi**
VAL_PAY_PY_001	Bukti Pembayaran	Bukti Pembayaran harus diunggah.
VAL_PAY_PY_002	Bukti Pembayaran	Format *file* bukti pembayaran harus sesuai (JPG/PNG).
VAL_PAY_PY_003	Bukti Pembayaran	Ukuran *file* bukti pembayaran tidak boleh melebihi batas maksimal yang ditentukan.
VAL_PAY_PY_004	Kode Promo	Kode Promo bersifat opsional.
VAL_PAY_PY_005	Kode Promo	Kode Promo harus valid dan belum kedaluwarsa jika digunakan.

Tabel 3.13 Cek Validasi Pembayaran Tagihan

Tabel 3.22 menyajikan nilai-nilai uji spesifik untuk *field* ukuran *file* bukti pembayaran pada modul Pembayaran Tagihan, yang difokuskan pada batas bawah (BLB, LB, ALB) dan batas atas (BUB, UB, AUB) berdasarkan kriteria validasi yang telah ditetapkan.

No.	**Data Input**	**Aturan Batas Terkait**	**Nilai Uji**	**Valid / Invalid**	**Keterangan**
1	Ukuran File Bukti Pembayaran	Minimal 1 KB	0 KB	Invalid	BLB (*Below Lower Bound*)
2	Ukuran File Bukti Pembayaran	Minimal 1 KB	1 KB	Valid	LB (*Lower Bound*)

3	Ukuran File Bukti Pembayaran	Minimal 1 KB	2 KB	Valid	ALB (*Above Lower Bound*)
4	Ukuran File Bukti Pembayaran	Maksimal 5 MB	4.9 MB	Valid	BUB (*Below Upper Bound*)
5	Ukuran File Bukti Pembayaran	Maksimal 5 MB	5 MB	Valid	UB (*Upper Bound*)
6	Ukuran File Bukti Pembayaran	Maksimal 5 MB	5.1 MB	Invalid	AUB (*Above Upper Bound*)

Tabel 3.14 Kelas Boundary Value Analysis Pembayaran Tagihan

Skenario Validasi & *Boundary Value Analysis* Modul Pengajuan Komplain Penyewa Kos

Tabel 3.23 mengidentifikasi kriteria validasi untuk setiap *field input* pada modul Pengajuan Komplain, termasuk judul komplain, deskripsi komplain, dan lampiran foto (format dan ukuran).

ID Validasi	Data Input	Deskripsi Kriteria Validasi
VAL_COMP_PY_001	Judul Komplain	Judul Komplain tidak boleh kosong.
VAL_COMP_PY_002	Judul Komplain	Panjang Judul Komplain memiliki batas minimal 5 dan maksimal 100 karakter.
VAL_COMP_PY_003	Deskripsi Komplain	Deskripsi Komplain tidak boleh kosong.
VAL_COMP_PY_004	Deskripsi Komplain	Panjang Deskripsi Komplain memiliki batas minimal 10 dan maksimal 500 karakter.
VAL_COMP_PY_005	Lampiran Foto	Lampiran Foto bersifat opsional, namun jika diunggah harus valid.
VAL_COMP_PY_006	Lampiran Foto	Format *file* lampiran foto harus sesuai (JPG/PNG).
VAL_COMP_PY_007	Lampiran Foto	Ukuran *file* lampiran foto tidak boleh melebihi batas maksimal yang ditentukan.

Tabel 3.15 Cek Validasi Pengajuan Komplain

Tabel 3.24 menyajikan nilai-nilai uji spesifik untuk *field* judul komplain dan deskripsi komplain pada modul Pengajuan Komplain, yang difokuskan pada batas bawah (BLB, LB, ALB) dan batas atas (BUB, UB, AUB) berdasarkan kriteria validasi yang telah ditetapkan.

No.	Data Input	Aturan Batas Terkait	Nilai Uji	Valid / Invalid	Keterangan

1	Judul Komplain	Minimal 5 karakter (VAL_COMP_PY_002)	Pintu	Invalid	BLB (*Below Lower Bound*)
2	Judul Komplain	Minimal 5 karakter (VAL_COMP_PY_002)	Pintu rusak	Valid	LB (*Lower Bound*)
3	Judul Komplain	Minimal 5 karakter (VAL_COMP_PY_002)	Pintu rusak parah	Valid	ALB (*Above Lower Bound*)
4	Judul Komplain	Maksimal 100 karakter (VAL_COMP_PY_002)	Judul komplain sangat panjang sekali dan ini adalah contoh pengujian batas maksimal karakter untuk judul komplain yang sangat panjang	Valid	BUB (*Below Upper Bound*)
5	Judul Komplain	Maksimal 100 karakter (VAL_COMP_PY_002)	Judul komplain sangat panjang sekali dan ini adalah contoh pengujian batas maksimal karakter untuk judul komplain yang sangat panjang.	Valid	UB (*Upper Bound*)
6	Judul Komplain	Maksimal 100 karakter (VAL_COMP_PY_002)	Judul komplain sangat panjang sekali dan ini adalah contoh pengujian batas maksimal karakter untuk judul komplain yang sangat panjang sekali.	Invalid	AUB (*Above Upper Bound*)
7	Deskripsi Komplain	Minimal 10 karakter (VAL_COMP_PY_004)	Rusak ringan	Invalid	BLB (*Below Lower Bound*)
8	Deskripsi Komplain	Minimal 10 karakter (VAL_COMP_PY_004)	Rusak total.	Valid	LB (Lower Lower)

9	Deskripsi Komplain	Minimal 10 karakter (VAL_COMP_PY_004)	Rusak total di bagian pintu.	Valid	ALB (*Above Lower Bound*)
10	Deskripsi Komplain	Maksimal 500 karakter (VAL_COMP_PY_004)	Deskripsi komplain yang sangat panjang ini digunakan untuk menguji batasan karakter maksimal pada *field* deskripsi. Pengujian ini memastikan bahwa sistem dapat menangani *input* teks yang panjang tanpa *error*. Ini adalah contoh deskripsi yang mendekati batas atas karakter yang diizinkan oleh sistem untuk sebuah komplain. Deskripsi ini harus dapat disimpan dengan benar dan ditampilkan sepenuhnya tanpa terpotong.	Valid	BUB (*Below Upper Bound*)
11	Deskripsi Komplain	Maksimal 500 karakter (VAL_COMP_PY_004)	Deskripsi komplain yang sangat panjang ini digunakan untuk menguji batasan karakter maksimal pada *field* deskripsi. Pengujian ini memastikan bahwa sistem dapat menangani *input* teks yang panjang tanpa *error*. Ini adalah contoh deskripsi yang mendekati batas atas karakter yang diizinkan oleh sistem untuk sebuah komplain. Deskripsi ini	Valid	UB (*Upper Bound*)

			harus dapat disimpan dengan benar dan ditampilkan sepenuhnya tanpa terpotong.A		
12	Deskripsi Komplain	Maksimal 500 karakter (VAL_COMP_PY_004)	Deskripsi komplain yang sangat panjang ini digunakan untuk menguji batasan karakter maksimal pada *field* deskripsi. Pengujian ini memastikan bahwa sistem dapat menangani *input* teks yang panjang tanpa *error*. Ini adalah contoh deskripsi yang mendekati batas atas karakter yang diizinkan oleh sistem untuk sebuah komplain. Deskripsi ini harus dapat disimpan dengan benar dan ditampilkan sepenuhnya tanpa terpotong.AB	Invalid	AUB (*Above Upper Bound*)

Tabel 3.16 Kelas Boundary Value Analysis Pengajuan Komplain

b. Pengembangan Skenario Uji (*Test Scenario*)

Pengembangan skenario uji dilakukan dengan menyusun ID Test Case, merumuskan skenario uji berdasarkan alur interaksi pengguna dengan sistem, serta menetapkan hasil yang diharapkan dari setiap tindakan yang dilakukan. Setiap test case dirancang untuk memverifikasi apakah sistem website merespons sesuai dengan fungsionalitas yang diharapkan, baik dalam kondisi input valid maupun ketika terjadi kesalahan atau input tidak valid.

Skenario Pengujian Modul Registrasi

Tabel 3.17 merupakan perancangan kasus uji fungsionalitas Registrasi Akun Pemilik Kos, yang disusun berdasarkan kriteria validasi dan nilai batas yang telah didefinisikan pada Tabel 3.1 dan Tabel 3.2.

ID Test Case	Skenario Uji	Hasil yang Diharapkan
TC_REGIS_PK_R01	Registrasi dengan semua data valid.	Berhasil registrasi, email verifikasi terkirim.
TC_REGIS_PK_R02	Registrasi dengan *field* Nama Lengkap kosong.	Gagal, muncul pesan validasi "Nama Lengkap tidak boleh kosong."
TC_REGIS_PK_R03	Registrasi dengan *field* Email kosong.	Gagal, muncul pesan validasi "Email tidak boleh kosong."
TC_REGIS_PK_R04	Registrasi dengan *field* Kata Sandi kosong.	Gagal, muncul pesan validasi "Kata Sandi tidak boleh kosong."
TC_REGIS_PK_R05	Registrasi dengan Konfirmasi Kata Sandi kosong.	Gagal, pesan validasi "Konfirmasi Kata Sandi tidak boleh kosong."
TC_REGIS_PK_R06	Registrasi dengan format email salah.	Gagal, pesan validasi "Format email tidak valid."
TC_REGIS_PK_R07	Registrasi dengan email yang sudah terdaftar.	Gagal, muncul pesan "Email sudah terdaftar."
TC_REGIS_PK_R08	Registrasi dengan kata sandi lemah (kurang dari 8 karakter).	Gagal, pesan validasi panjang atau kompleksitas kata sandi.
TC_REGIS_PK_R09	Registrasi dengan Konfirmasi Kata Sandi tidak cocok.	Gagal, pesan "Konfirmasi kata sandi tidak cocok."
TC_REGIS_PK_R10	Registrasi dengan Nama Lengkap di bawah batas minimal.	Gagal, pesan validasi panjang Nama Lengkap.
TC_REGIS_PK_R11	Registrasi dengan Email di bawah batas minimal.	Gagal, pesan validasi panjang Email.
TC_REGIS _PK_R12	Registrasi dengan Kata Sandi di atas batas maksimal.	Gagal, pesan validasi panjang Kata Sandi.
TC_REGIS_PK_R13	Registrasi tanpa memilih peran.	Gagal, pesan validasi "Pilihan Peran harus dipilih."

Tabel 3.17 Perancangan Kasus Uji Fungsionalitas Registrasi Akun Pemilik Kos

Skenario Pengujian Modul Login Akun

Tabel 3.18 menyajikan perancangan kasus uji fungsionalitas Login Akun Pemilik Kos, yang disusun berdasarkan kriteria validasi dan nilai batas yang telah didefinisikan pada Tabel 3.3 dan Tabel 3.4.

ID Test Case	Skenario Uji	Hasil yang Diharapkan
TC_LOGIN_PK_R01	Login dengan email dan kata sandi valid.	Berhasil Login, diarahkan ke dashboard.
TC_LOGIN_PK_R02	Login dengan kata sandi salah.	Gagal, pesan "Email atau Kata Sandi salah."
TC_LOGIN_PK_R03	Login dengan email tidak terdaftar.	Gagal, pesan "Email atau Kata Sandi salah."
TC_LOGIN_PK_R04	Login dengan email kosong.	Gagal, pesan validasi "Email tidak boleh kosong."
TC_LOGIN_PK_R05	Login dengan kata sandi kosong.	Gagal, pesan validasi "Kata Sandi tidak boleh kosong."
TC_LOGIN_PK_R06	Login dengan format email salah.	Gagal, pesan validasi "Format email tidak valid."
TC_LOGIN_PK_R07	Login dengan email di bawah batas minimal.	Gagal, pesan validasi panjang email.
TC_LOGIN_PK_R08	Login dengan kata sandi di bawah batas minimal.	Gagal, pesan validasi panjang Kata Sandi.
TC_LOGIN_PK_R09	Login dengan email di atas batas maksimal.	Gagal, pesan validasi panjang email.
TC_LOGIN_PK_R10	Login dengan kata sandi di atas batas maksimal.	Gagal, pesan validasi panjang Kata Sandi.

Tabel 3.18 Perancangan Kasus Uji Fungsionalitas Login Akun Pemilik Kos

Skenario Pengujian Modul Tampilan Pemilik Kelola Properti

Tabel 3.19 mengidentifikasi perancangan kasus uji fungsionalitas Kelola Properti, yang disusun berdasarkan kriteria validasi dan nilai batas yang telah didefinisikan pada Tabel 3.5 dan Tabel 3.6.

ID Test Case	Skenario Uji	Hasil yang Diharapkan
TC_KP_PK_R01	Tambah properti dengan semua data valid.	Properti baru berhasil disimpan dan tampil di daftar.
TC_KP_PK_R02	Tambah properti dengan Nama Properti kosong.	Gagal, pesan validasi "Nama Properti tidak boleh kosong."
TC_KP_PK_R03	Tambah properti dengan Alamat Properti kosong.	Gagal, pesan validasi "Alamat Properti tidak boleh kosong."
TC_KP_PK_R04	Tambah properti dengan format foto salah.	Gagal, pesan validasi "Format file foto tidak didukung."

TC_KP_PK_R05	Lihat detail properti yang valid.	Berhasil menampilkan halaman detail dengan informasi benar.
TC_KP_PK_R06	Edit properti dengan data valid.	Perubahan berhasil disimpan, data ter-update.
TC_KP_PK_R07	Edit properti dan mengosongkan Nama Properti.	Gagal, pesan validasi "Nama Properti tidak boleh kosong."
TC_KP_PK_R08	Edit properti dengan format Harga Sewa salah (input teks).	Gagal, pesan validasi "Format Harga Sewa tidak valid."
TC_KP_PK_R09	Hapus properti yang valid dengan konfirmasi "Ya".	Properti berhasil dihapus dari daftar.
TC_KP_PK_R10	Hapus properti namun memilih "Batal" pada konfirmasi.	Penghapusan dibatalkan, properti tetap ada.

Tabel 3.19 Perancangan Kasus Uji Fungsionalitas Kelola Properti

Skenario Pengujian Modul Data Penghuni

Tabel 3.20 menyajikan perancangan kasus uji fungsionalitas Data Penghuni, yang disusun berdasarkan kriteria validasi dan nilai batas yang telah didefinisikan pada Tabel 3.7 dan Tabel 3.8.

ID Test Case	Skenario Uji	Hasil yang Diharapkan
TC_DP_PK_01	Pemilik menambahkan data penghuni baru secara manual dengan semua data valid.	Penghuni baru berhasil dibuat dan muncul di daftar "Penghuni Aktif".
TC_DP_PK_02	Pemilik menambahkan data penghuni baru namun mengosongkan *field* wajib (Nama Penghuni).	Gagal, sistem menampilkan pesan validasi "Nama Penghuni tidak boleh kosong."
TC_DP_PK_03	Pemilik menerima pengajuan pendaftar baru yang valid.	Status pendaftar berubah menjadi "Aktif" dan datanya pindah ke modul "Data Penghuni".
TC_DP_PK_04	Pemilik mengedit data penghuni dengan informasi yang valid.	Data penghuni berhasil diperbarui di dalam sistem.
TC_DP_PK_05	Pemilik menghapus data penghuni yang sudah tidak aktif.	Data penghuni berhasil dihapus dari daftar.
TC_DP_PK_06	Pemilik mencoba menambahkan penghuni baru ke kamar yang statusnya sudah "Terisi".	Gagal, sistem menampilkan pesan error "Kamar sudah terisi dan tidak tersedia."
TC_DP_PK_07	Pemilik mengedit nomor telepon penghuni dengan format tidak valid (mengandung huruf).	Gagal, sistem menampilkan pesan validasi "Format nomor telepon tidak valid."
TC_DP_PK_08	Pemilik menggunakan filter untuk menampilkan hanya penghuni "Aktif".	Tabel hanya menampilkan data penghuni dengan status "Aktif".
TC_DP_PK_09	Pemilik menggunakan fitur pencarian untuk menemukan penghuni berdasarkan nama.	Tabel menampilkan hasil pencarian yang sesuai dengan nama yang diinput.

TC_DP_PK_10	Pemilik mengonfirmasi pengajuan berhenti sewa dari penghuni.	Status penghuni berubah menjadi "Tidak Aktif" dan status kamar menjadi "Kosong".

Tabel 3.20 Perancangan Kasus Uji Fungsionalitas Data Penghuni

Skenario Pengujian Modul Manajemen Tagihan

Tabel 3.21 mengidentifikasi perancangan kasus uji fungsionalitas Manajemen Tagihan, yang disusun berdasarkan kriteria validasi dan nilai batas yang telah didefinisikan pada Tabel 3.9 dan Tabel 3.10.

ID Test Case	Skenario Uji	Hasil yang Diharapkan
TC_MT_PK_01	Pemilik mengonfirmasi masuknya penghuni baru.	Tagihan pertama untuk penghuni tersebut otomatis dibuat dengan status "Belum Bayar".
TC_MT_PK_02	Pemilik mengubah status tagihan dari "Butuh Konfirmasi" menjadi "Lunas".	Status tagihan berhasil diubah menjadi "Lunas" dan pindah ke tab "Lunas".
TC_MT_PK_03	Pemilik mengubah status tagihan "Lunas" untuk penyewa dengan durasi sewa >1 bulan.	Tagihan untuk periode berikutnya otomatis dibuat dengan status "Belum Bayar".
TC_MT_PK_04	Pemilik menggunakan filter untuk menampilkan tagihan dengan status "Belum Bayar".	Hanya tagihan dengan status "Belum Bayar" yang ditampilkan di tabel.
TC_MT_PK_05	Pemilik menggunakan filter untuk menampilkan tagihan dengan status "Lunas".	Hanya tagihan dengan status "Lunas" yang ditampilkan di tabel.
TC_MT_PK_06	Pemilik mencoba mengubah status tagihan "Belum Bayar" langsung menjadi "Lunas".	Aksi ditolak atau opsi tidak tersedia, karena harus melalui status "Butuh Konfirmasi" terlebih dahulu.

Tabel 3.21 Perancangan Kasus Uji Fungsionalitas Manajemen Tagihan

Skenario Pengujian Modul Pendaftaran Sewa

Tabel 3.22 menyajikan perancangan kasus uji fungsionalitas Pendaftaran Sewa, yang disusun berdasarkan kriteria validasi dan nilai batas yang telah didefinisikan pada Tabel 3.11 dan Tabel 3.12.

ID Test Case	Skenario Uji	Hasil yang Diharapkan
TC_PY_REG_R01	Penyewa mengajukan pendaftaran dengan mengisi semua data formulir secara valid.	Pengajuan pendaftaran berhasil terkirim ke pemilik dan penyewa menerima notifikasi sukses.

TC_PY_REG_R02	Penyewa mengajukan pendaftaran namun mengosongkan salah satu *field* wajib (contoh: durasi sewa).	Pengajuan gagal terkirim, sistem menampilkan pesan validasi "Durasi sewa wajib diisi."
TC_PY_REG_R03	Penyewa memasukkan data dengan format tidak valid pada *field* durasi sewa (contoh: "satu bulan" bukan angka).	Pengajuan gagal, sistem menampilkan pesan validasi "Durasi harus berupa angka."
TC_PY_REG_R04	Penyewa mencoba mendaftar ke kamar yang statusnya sudah "Terisi" atau "Tidak Tersedia".	Sistem menolak pendaftaran dan menampilkan pesan "Kamar yang dipilih tidak tersedia."
TC_PY_REG_R05	Penyewa mengunggah foto KTP dengan format yang tidak didukung (contoh: .gif).	Pengajuan gagal, sistem menampilkan pesan validasi "Format foto KTP tidak valid."

Tabel 3.22 Perancangan Kasus Uji Fungsionalitas Pendaftaran Sewa

Skenario Pengujian Modul Pembayaran Tagihan Penyewa

Tabel 3.23 mengidentifikasi perancangan kasus uji fungsionalitas Pembayaran Tagihan Penyewa Kos, yang disusun berdasarkan kriteria validasi dan nilai batas yang telah didefinisikan pada Tabel 3.13 dan Tabel 3.14.

ID Test Case	Skenario Uji	Hasil yang Diharapkan
TC_PY_PAY_01	Penyewa membayar tagihan dengan mengunggah bukti bayar berformat valid (JPG/PNG).	Bukti pembayaran berhasil diunggah, status tagihan berubah menjadi "Butuh Konfirmasi".
TC_PY_PAY_02	Penyewa mencoba mengunggah bukti bayar dengan format tidak valid.	Sistem menolak file dan menampilkan pesan error "Format file tidak didukung."
TC_PY_PAY_03	Penyewa mencoba mengirim pembayaran tanpa mengunggah bukti bayar.	Proses gagal, sistem menampilkan pesan validasi "Bukti pembayaran wajib diunggah."
TC_PY_PAY_04	Penyewa mengunggah file bukti pembayaran yang ukurannya melebihi batas maksimal yang ditentukan.	Unggah gagal, sistem menampilkan pesan error "Ukuran file terlalu besar."
TC_PY_PAY_05	Penyewa menerapkan kode promo yang valid saat pembayaran.	Total tagihan berhasil dikurangi sesuai nilai promo, dan pembayaran berhasil diproses.
TC_PY_PAY_06	Penyewa mencoba menerapkan kode promo yang tidak valid atau sudah kedaluwarsa.	Sistem menampilkan pesan "Kode promo tidak valid atau sudah tidak berlaku."

Tabel 3.23 Perancangan Kasus Uji Fungsionalitas Pembayaran Tagihan Penyewa

Skenario Pengujian Modul Pengajuan Komplain Penyewa

Tabel 3.24 menyajikan perancangan kasus uji fungsionalitas Pengajuan Komplain Penyewa Kos, yang disusun berdasarkan kriteria validasi dan nilai batas yang telah didefinisikan pada Tabel 3.15 dan Tabel 3.16.

ID Test Case	Skenario Uji	Hasil yang Diharapkan
TC_PY_COMP_01	Penyewa mengajukan komplain dengan mengisi judul dan deskripsi secara valid.	Komplain berhasil terkirim dan muncul di daftar komplain aktif penyewa serta di dashboard pemilik.
TC_PY_COMP_02	Penyewa mengajukan komplain namun mengosongkan *field* deskripsi yang wajib diisi.	Pengajuan komplain gagal, sistem menampilkan pesan validasi "Deskripsi komplain tidak boleh kosong."
TC_PY_COMP_03	Penyewa mengajukan komplain dengan judul dan deskripsi yang sangat panjang (melebihi batas karakter).	Pengajuan gagal, sistem menampilkan pesan validasi "Jumlah karakter melebihi batas."
TC_PY_COMP_04	Penyewa mengajukan komplain dengan melampirkan file gambar yang valid (JPG/PNG).	Komplain berhasil terkirim beserta lampirannya, dan dapat dilihat oleh pemilik.
TC_PY_COMP_05	Penyewa mencoba melampirkan file dengan format tidak valid pada komplain.	Sistem menolak lampiran dan menampilkan pesan error "Format lampiran tidak didukung."

Tabel 3.24 Perancangan Kasus Uji Fungsionalitas Pengajuan Komplain Penyewa

3.4.3 Kriteria Keberhasilan dan Penanganan Hasil

Untuk memastikan objektivitas dan akurasi dalam penilaian fungsionalitas aplikasi website Nestify, kriteria keberhasilan pengujian ditetapkan pada dua tingkatan analisis yang berbeda: tingkat kasus uji individu dan tingkat sistem secara keseluruhan. Penetapan kriteria ini esensial untuk mengidentifikasi keberhasilan maupun kegagalan secara sistematis, serta untuk memberikan dasar interpretasi kinerja aplikasi.

Tingkat Kasus Uji

Pada tingkatan ini, setiap kasus uji dievaluasi secara terpisah untuk menentukan status kelulusannya.

a. **Berhasil (*Pass*):** Sebuah kasus uji akan dinyatakan *Berhasil (Pass)* apabila hasil aktual yang diobservasi dari eksekusi sistem sepenuhnya sesuai dengan hasil yang diharapkan (*expected result*) yang telah didefinisikan dalam rancangan kasus uji. Ini berarti tidak

ditemukan adanya penyimpangan fungsional, perilaku yang tidak sesuai, atau pesan kesalahan yang tidak semestinya selama eksekusi kasus uji tersebut.

b. **Gagal (*Fail*):** Sebaliknya, sebuah kasus uji akan dinyatakan *Gagal (Fail)* jika ditemukan adanya penyimpangan sekecil apapun antara hasil aktual dan hasil yang diharapkan. Penyimpangan ini dapat berupa fungsionalitas tidak berjalan sama sekali, hasil *output* yang tidak sesuai, pesan kesalahan yang muncul tidak relevan atau tidak informatif, dan kondisi akhir sistem tidak tercapai sesuai ekspektasi.

Tingkat *Success Rate*

Kinerja fungsionalitas sistem secara keseluruhan dievaluasi berdasarkan hasil dari seluruh kasus uji yang telah dieksekusi. Metrik utama yang digunakan adalah persentase keberhasilan total, yang dihitung menggunakan formula sebagai berikut:

$$N = \frac{SP}{SM} \times 100\% \quad (1)$$

Keterangan:

- N = Persentase keberhasilan fungsionalitas sistem.
- SP = Jumlah kasus uji yang dinyatakan Berhasil *(Pass)*.
- SM = Total seluruh kasus uji yang dieksekusi.

Persentase keberhasilan ini kemudian diinterpretasikan menggunakan skala penilaian berikut untuk memberikan gambaran kualitatif terhadap kualitas fungsional aplikasi:

a. 89% – 100%: Sangat Baik

Kategori ini menunjukkan bahwa fungsionalitas aplikasi sangat stabil dan hampir sepenuhnya memenuhi semua persyaratan yang ditetapkan, dengan *bug* yang sangat minimal atau tidak ada sama sekali.

b. 70% – 88%: Baik

Menunjukkan bahwa sebagian besar fungsionalitas inti aplikasi berjalan dengan baik, meskipun mungkin terdapat beberapa *bug* minor yang tidak mengganggu operasional utama secara signifikan, namun perlu diperbaiki.

c. 31% – 69%: Kurang

Mengindikasikan adanya sejumlah *bug* fungsional yang signifikan, yang mungkin memengaruhi pengalaman pengguna atau menghambat operasional aplikasi. Perbaikan mendesak diperlukan.

d. 0% – 30%: Sangat Kurang

Menunjukkan bahwa aplikasi memiliki banyak *bug* fungsional kritis yang membuatnya tidak dapat digunakan atau sangat tidak stabil. Pengembangan ulang atau perbaikan besar sangat diperlukan.

Persentase ini menjadi indikator utama untuk menilai tingkat kesesuaian fungsionalitas website Nestify terhadap kebutuhan yang telah ditetapkan, memberikan dasar yang kuat untuk pengambilan keputusan terkait kelayakan rilis atau kebutuhan perbaikan lebih lanjut.

3.5 Pemeliharaan/ *Maintenance*

Tahap pemeliharaan (*maintenance*) merupakan fase terakhir dalam model pengembangan *Waterfall* yang dimulai setelah aplikasi diimplementasikan dan digunakan oleh pengguna. Tujuan utama dari tahap ini adalah untuk memastikan aplikasi dapat terus beroperasi dengan baik dan relevan seiring waktu. Kegiatan dalam tahap ini meliputi perbaikan kesalahan yang mungkin baru ditemukan, penyesuaian terhadap perubahan lingkungan seperti peramban atau sistem operasi baru, serta pengembangan atau penambahan fungsionalitas baru berdasarkan kebutuhan pengguna yang berkembang.

Pemeliharaan Korektif (*Corrective Maintenance*)

Aktivitas ini berfokus pada identifikasi dan perbaikan *bug* atau kegagalan fungsional yang ditemukan di seluruh modul aplikasi, baik pada antarmuka Pemilik kos maupun Penyewa. Proses ini mencakup perbaikan pada alur kerja utama seperti modul autentikasi (Login dan Registrasi), pengelolaan properti, manajemen kamar, data penyewa, dan tagihan. Selain itu, perbaikan juga akan dilakukan pada sisi Penyewa kos, meliputi fitur pendaftaran sewa, pembayaran, pengajuan komplain, hingga pengelolaan profil. Setelah setiap perbaikan dilakukan, akan dilaksanakan siklus pengujian ulang (*re-testing*) dan pengujian regresi untuk memastikan semua kesalahan telah teratasi tanpa menimbulkan masalah baru pada fitur lainnya.

Pemeliharaan Perfektif (*Perfective Maintenance*)

Aktivitas ini bertujuan untuk menyempurnakan dan memperluas fungsionalitas aplikasi. Rencana pengembangan lebih lanjut meliputi perluasan cakupan pengujian ke fitur-fitur yang belum diuji dalam penelitian ini, seperti pengelolaan detail kamar, manajemen data penyewa, dan pengelolaan tagihan. Selain itu, pemeliharaan ini juga merencanakan implementasi

pengujian non-fungsional, seperti pengujian kinerja dan keamanan, untuk meningkatkan kualitas dan keandalan sistem secara menyeluruh.

Pemeliharaan Adaptif (*Adaptive Maintenance*)

Aktivitas ini bertujuan untuk memastikan aplikasi tetap kompatibel dengan perkembangan teknologi di masa depan. Contohnya termasuk penyesuaian kode agar aplikasi dapat berjalan dengan baik pada versi terbaru dari peramban *browser*, pembaruan *framework* Laravel, atau perubahan lain pada lingkungan server.

Rancangan kegiatan pemeliharaan ini menjadi landasan untuk siklus hidup aplikasi Nestify di masa depan, memastikan bahwa prototipe yang dihasilkan dari penelitian ini dapat dikembangkan menjadi produk yang matang dan berkelanjutan.

BAB IV
HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil implementasi dari tahapan pengembangan aplikasi website Nestify. Bab ini juga memuat analisis mendalam terhadap hasil pengujian fungsional yang telah dilaksanakan. Pembahasan secara komprehensif akan mencakup implementasi antarmuka pengguna (*user interface*), detail hasil eksekusi kasus uji, serta interpretasi dan analisis terhadap temuan-temuan krusial selama proses pengujian.

4.1 Implementasi Antarmuka Pengguna

Bagian ini secara spesifik menampilkan hasil implementasi visual dari perancangan antarmuka yang telah disusun secara sistematis pada Bab III. Setiap tampilan visual dari halaman-halaman utama aplikasi website Nestify disajikan untuk memberikan gambaran konkret dan detail mengenai fungsionalitas yang telah berhasil dikembangkan.

4.1.1 Halaman Publik dan Autentikasi

Halaman *Landing Page*

Pada bagian ini, implementasi *Landing Page* disajikan sebagai tampilan awal aplikasi *Nestify* berbasis website. Berbeda dengan penelitian sebelumnya oleh (Fauzi, 2024) yang mengadopsi *splash-screen* untuk *onboarding* pada aplikasi bergerak, dalam implementasi berbasis website ini, *splash-screen* tersebut digantikan oleh *Landing Page*. Adaptasi ini didasarkan pada pertimbangan bahwa *Landing Page* lebih efektif dalam memberikan impresi awal yang kuat serta menyajikan informasi esensial mengenai aplikasi *Nestify* kepada calon pengguna website. *Landing Page* ini secara strategis memaparkan berbagai informasi penting, sebagaimana dijelaskan pada gambar-gambar berikut.

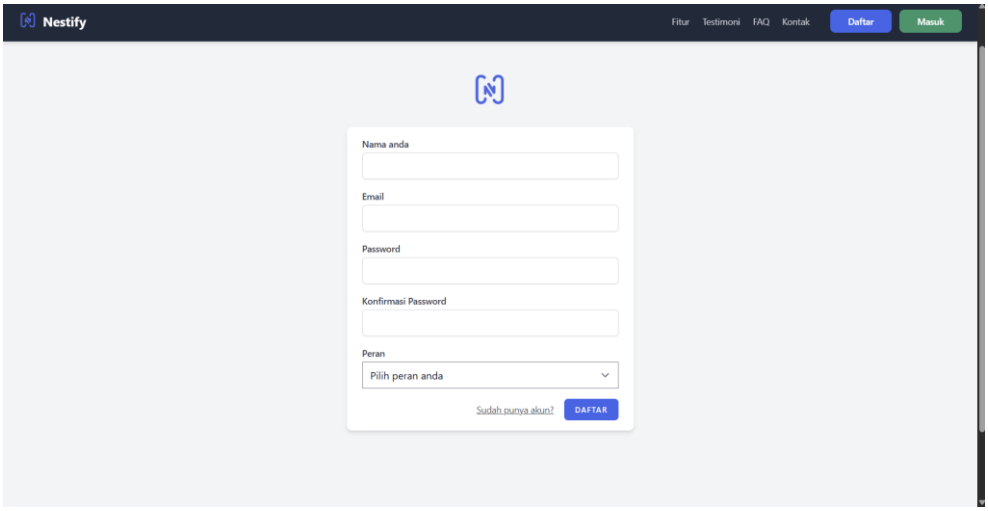
Gambar 4.1 menampilkan tampilan awal *Landing Page* aplikasi Nestify. Halaman ini berfungsi sebagai gerbang utama yang menyajikan informasi pengenalan aplikasi, fitur-fitur utama, testimoni, FAQ, dan informasi kontak, dirancang untuk memberikan impresi pertama yang kuat kepada calon pengguna.



Gambar 4.1 Tampilan *Landing page* Aplikasi Nestify

Halaman Registrasi Pengguna

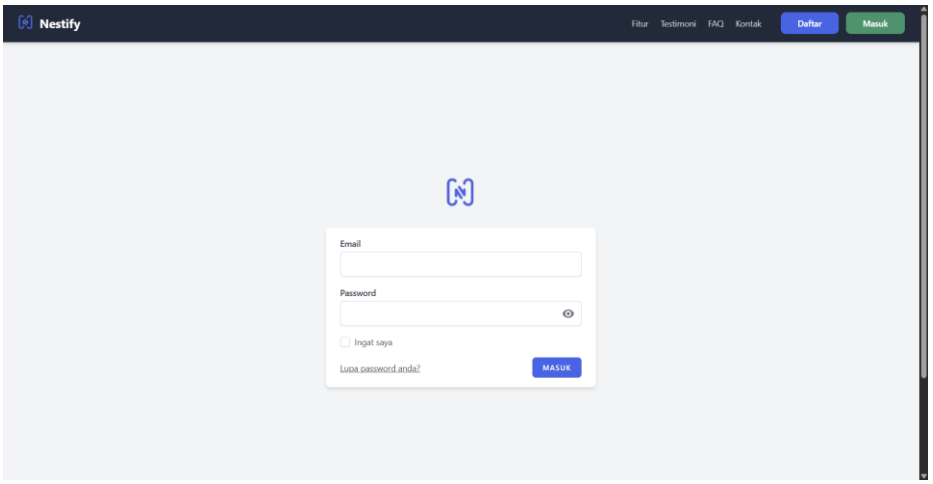
Halaman ini didesain secara spesifik untuk memfasilitasi pengguna baru dalam pembuatan akun pada aplikasi *Nestify*, baik yang berperan sebagai Pemilik kos maupun Penyewa. Gambar 4.2 menampilkan antarmuka awal halaman registrasi pengguna. Halaman ini dirancang untuk memungkinkan pengguna baru (Pemilik kos atau Penyewa) membuat akun pada aplikasi *Nestify*. Formulir pendaftaran mewajibkan pengguna untuk mengisi informasi esensial berupa nama lengkap, alamat email, kata sandi, dan konfirmasi kata sandi, dan memilih peran (Pemilik kos atau Penyewa), yang krusial dalam menentukan alur fungsionalitas selanjutnya.



Gambar 4.2 Tampilan Halaman Registrasi Pengguna

Halaman *Login* Pengguna

Gambar 4.3 menampilkan antarmuka halaman *login* pengguna. Halaman ini berfungsi sebagai gerbang akses utama bagi pengguna yang telah terdaftar untuk masuk ke dalam sistem *Nestify*. Pengguna diwajibkan untuk memasukkan alamat email dan kata sandi yang telah terdaftar sebelumnya. Untuk meningkatkan kenyamanan pengguna, halaman ini juga menyediakan opsi "Ingat saya" yang memungkinkan sistem untuk menyimpan kredensial *login* pengguna pada perangkat yang digunakan. Selain itu, terdapat fitur "Lupa password Anda?" yang dirancang untuk memfasilitasi pengguna dalam proses pengaturan ulang kata sandi apabila lupa, dengan mengarahkan mereka ke halaman khusus untuk prosedur tersebut.



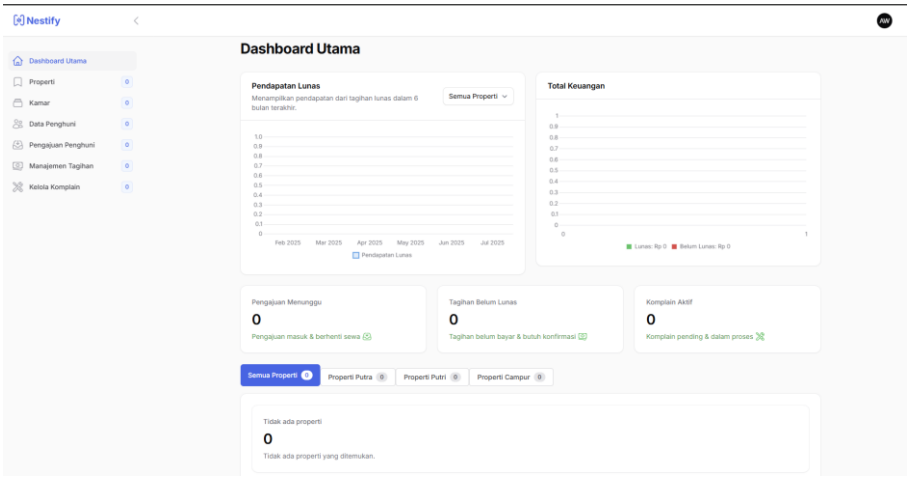
Gambar 4.3 Tampilan Halaman *Login* Pengguna

4.1.2 Antarmuka Fungsionalitas Pemilik Kos

Dashboard Utama

Gambar 4.4 menunjukkan tampilan Dashboard Utama bagi Pemilik kos setelah login. Dasbor ini menyajikan ringkasan informasi penting, seperti grafik “Pendapatan Lunas” enam bulan terakhir dan “Total Keuangan”. Di bawahnya, tersedia *widget* yang berisi jumlah pengajuan yang menunggu, tagihan belum lunas, dan komplain aktif. Pengguna juga bisa memfilter dan melihat daftar properti berdasarkan jenisnya. Antarmuka ini dirancang untuk

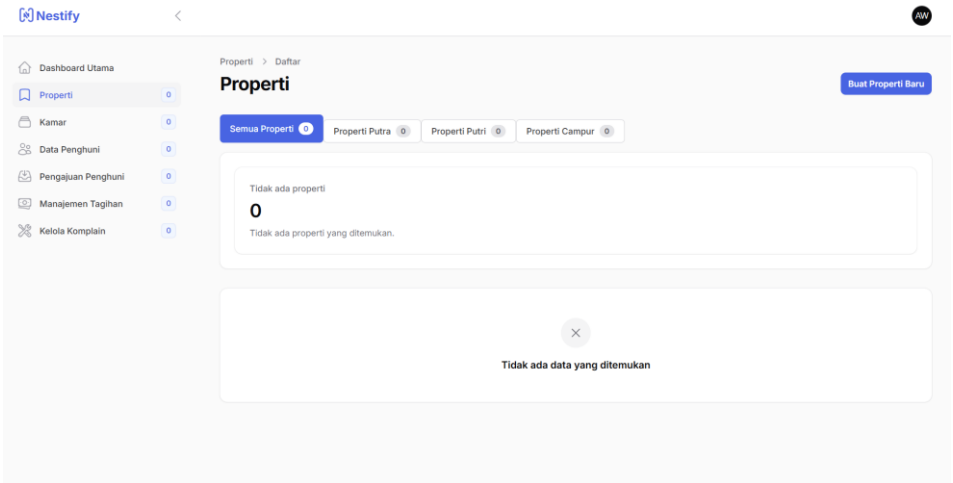
memberi gambaran operasional secara *real-time* dan memudahkan akses ke fitur manajemen lainnya.



Gambar 4.4 Tampilan Dashboard Utama Pemilik Kos

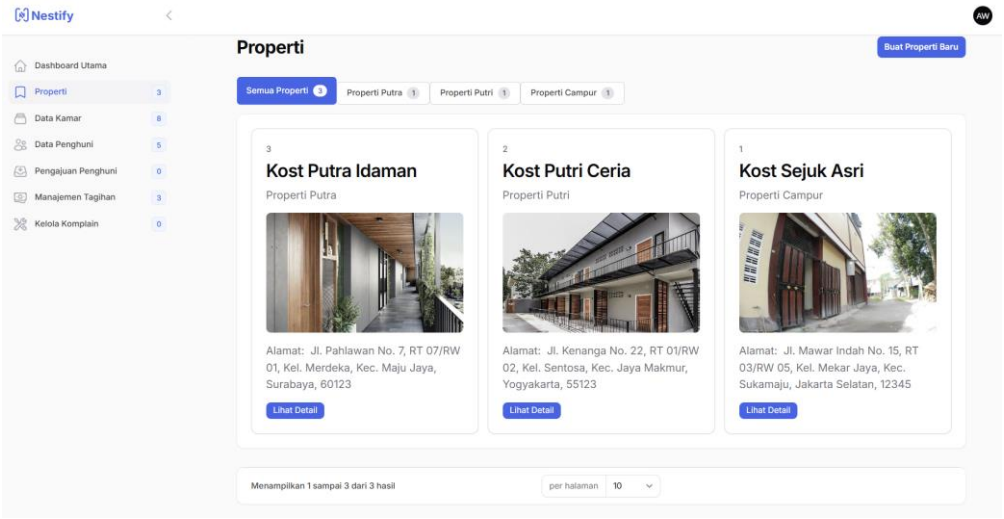
Halaman Kelola Properti

Gambar 4.5 menampilkan antarmuka halaman "Properti" yang didesain untuk memudahkan Pemilik kos dalam mengelola data properti mereka. Halaman ini menyajikan daftar properti yang dimiliki, dilengkapi dengan fitur pencarian dan filter berdasarkan jenis properti (Semua Properti, Properti Putra, Properti Putri, Properti Campur). Pemilik dapat dengan mudah menambahkan properti baru melalui tombol "Buat Properti Baru".



Gambar 4.5 Tampilan Halaman Kelola Properti

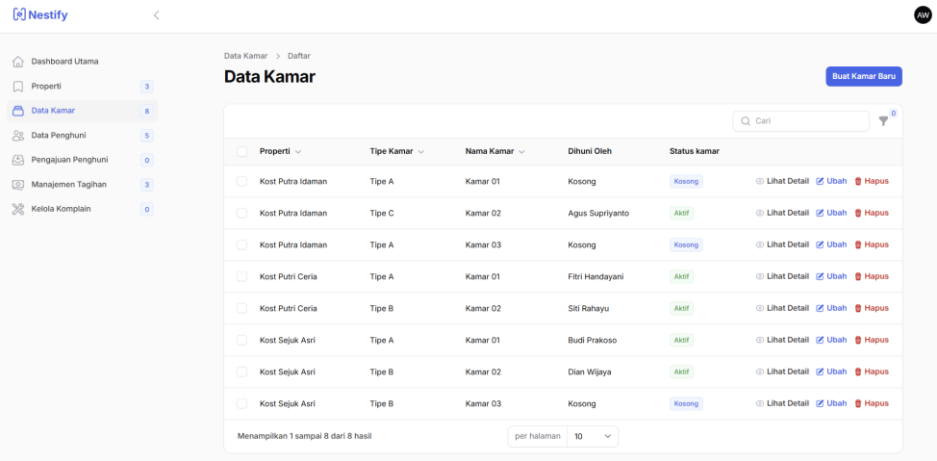
Gambar 4.6 merupakan tampilan halaman "Properti" ketika sebuah kos/properti baru pertama kali berhasil ditambahkan ke dalam sistem. Pada kondisi ini, daftar properti akan menampilkan entri tunggal dari kos yang baru saja dibuat, seperti "Kos Putra Adhiwangsa". Tampilan ini mengonfirmasi bahwa proses penambahan properti telah sukses dan properti tersebut kini siap untuk dikelola lebih lanjut oleh pemilik. Dari halaman ini, pemilik juga dapat melihat detail setiap properti, melakukan perubahan data, atau menghapus properti yang tidak lagi diperlukan.



Gambar 4.6 Tampilan Halaman Properti Setelah Penambahan Kos Pertama

Halaman Kelola Kamar

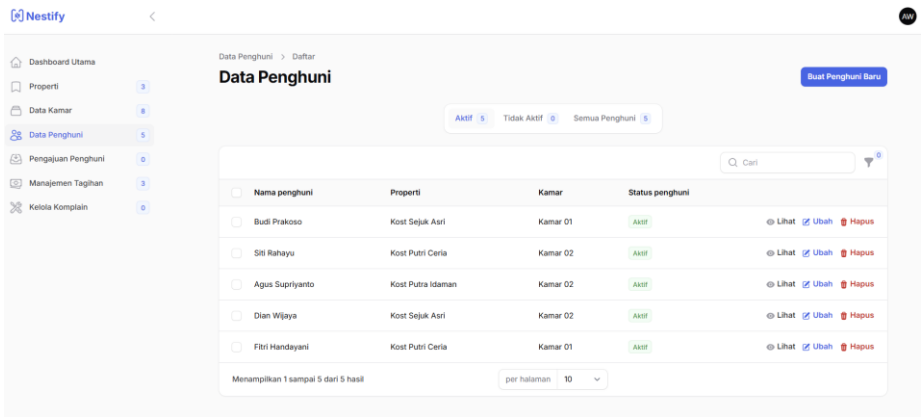
Gambar 4.7 menunjukkan tampilan awal halaman "Kamar". Halaman ini menampilkan daftar unit kamar yang terdaftar di properti pemilik, dilengkapi dengan fitur pencarian dan filter untuk mempermudah navigasi. Dari halaman ini, pemilik dapat melakukan berbagai tindakan manajemen kamar secara terpusat, seperti menambah kamar baru, melihat detail kamar, mengubah data kamar, menghapus kamar, dan memfilter kamar berdasarkan properti tertentu. Fitur-fitur ini dirancang untuk memberikan kontrol penuh kepada pemilik dalam mengelola unit kamar mereka.



Gambar 4.7 Tampilan Halaman Kamar

Halaman Data Penghuni

Gambar 4.8 menampilkan halaman "Data Penghuni" yang menyajikan daftar penyewa dengan fitur pencarian dan filter berdasarkan status. Pemilik dapat menambah penghuni baru lewat tombol "Buat Penghuni Baru", melihat detail, mengubah, atau menghapus data penghuni langsung dari halaman ini. Setelah data ditambahkan, sistem otomatis memperbarui daftar serta indikator jumlah penghuni di sidebar. Aksi hapus disertai pop-up konfirmasi untuk mencegah kesalahan.

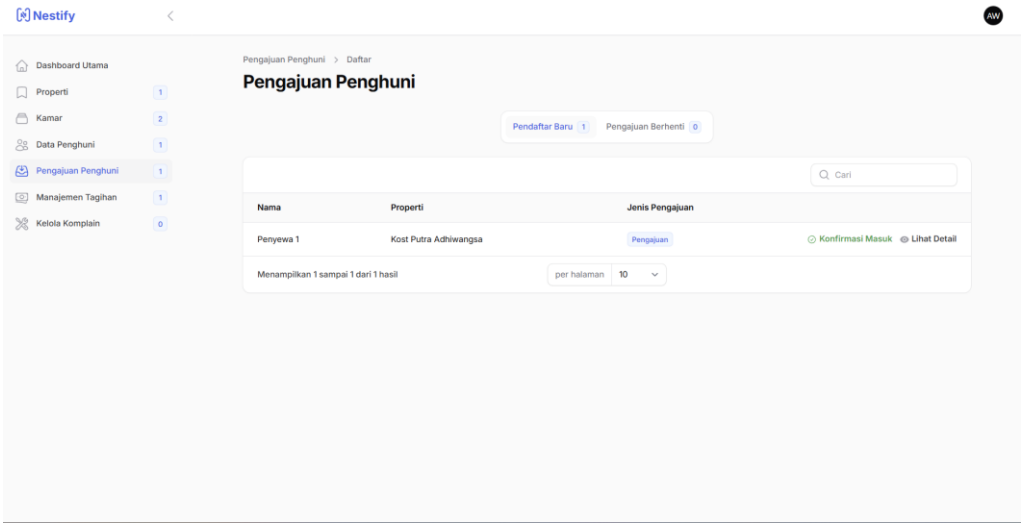


Gambar 4.8 Halaman Data Penghuni

Halaman Pengajuan Penghuni

Gambar 4.9 menampilkan halaman "Pengajuan Penghuni" yang membantu pemilik mengelola permohonan masuk dari calon atau penghuni lama. Halaman ini terbagi dua tab:

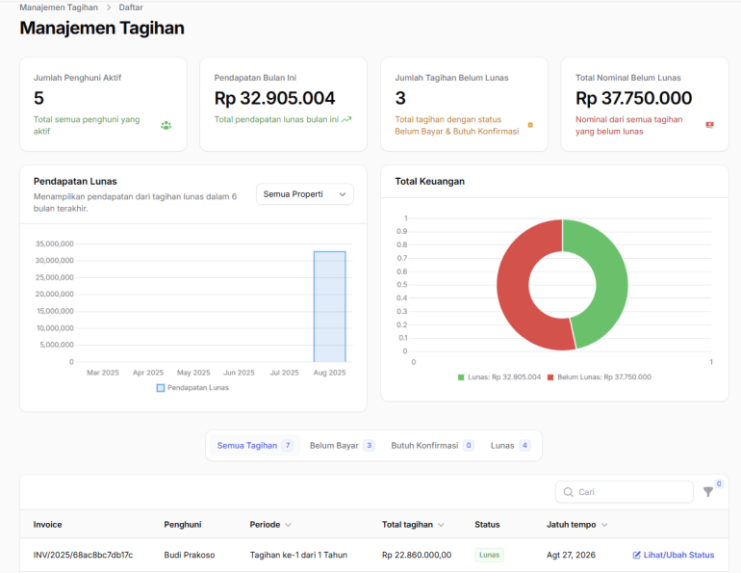
“Pendaftar Baru” dan “Pengajuan Berhenti”, masing-masing dilengkapi indikator angka untuk menunjukkan jumlah pengajuan belum diproses. Pemilik dapat melihat detail pengajuan, menyetujui masuknya penghuni baru, atau mengonfirmasi penghentian sewa melalui pop-up khusus. Setiap aksi berhasil akan memicu notifikasi dan memperbarui tampilan indikator angka secara otomatis.



Gambar 4.9 Tampilan Halaman Pengajuan Penghuni

Halaman Manajemen Tagihan

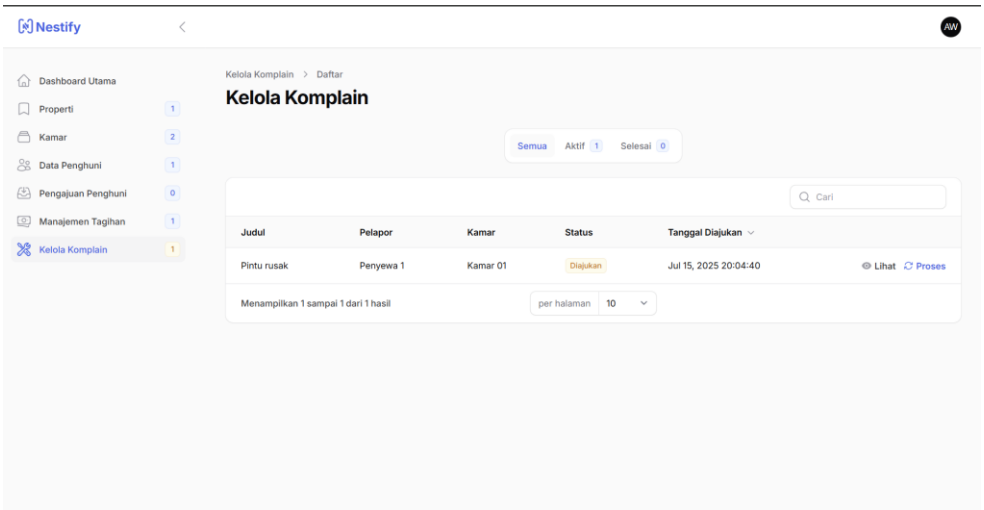
Gambar 4.10 menampilkan halaman “Manajemen Tagihan” saat terdapat tagihan yang menunggu konfirmasi. Halaman ini memuat ringkasan status keuangan dan tagihan, mulai dari “Semua Tagihan”, “Belum Bayar”, “Butuh Konfirmasi”, hingga “Lunas”. Pemilik dapat melihat detail tagihan, mengonfirmasi pelunasan, serta memantau sistem tagihan berulang yang aktif otomatis setelah pembayaran sebelumnya selesai. Tersedia juga fitur pencarian dan filter untuk mempermudah pencatatan dan pelacakan tagihan secara efisien.



Gambar 4.10 Tampilan Halaman Manajemen Tagihan

Halaman Kelola Komplain

Gambar 4.11 menampilkan halaman “Kelola Komplain” saat terdapat keluhan yang diajukan oleh penyewa. Daftar komplain ditampilkan dalam tiga tab: “Semua”, “Aktif”, dan “Selesai”, yang memudahkan pemilik dalam melacak status penanganan. Pemilik dapat melihat detail, memproses, dan menyelesaikan komplain melalui pop-up yang tersedia. Setiap tindakan akan secara otomatis memperbarui status komplain serta menampilkan notifikasi berupa indikator angka pada masing-masing tab. Fitur ini dirancang untuk membantu pemilik menangani keluhan secara cepat dan terstruktur.

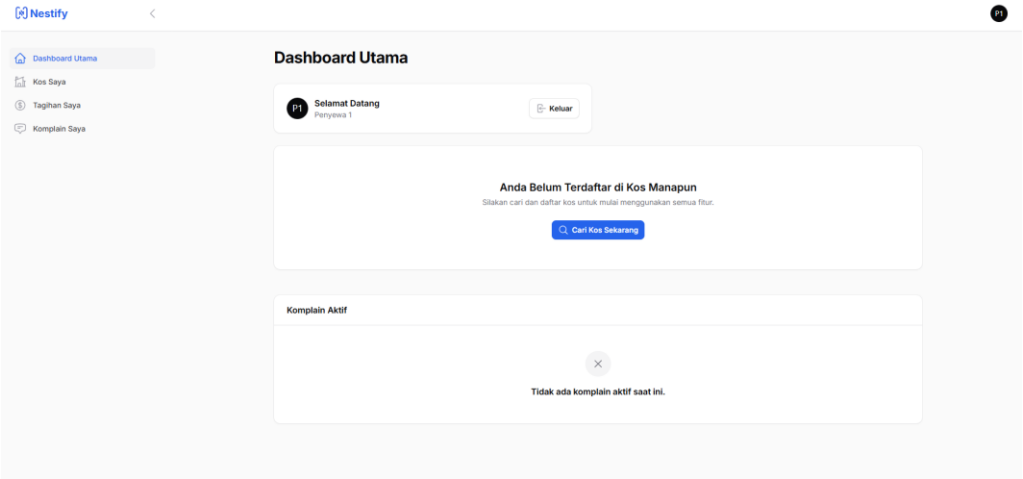


Gambar 4.11 Tampilan Halaman Kelola Komplain

4.1.3 Antarmuka Fungsionalitas Penyewa Kos

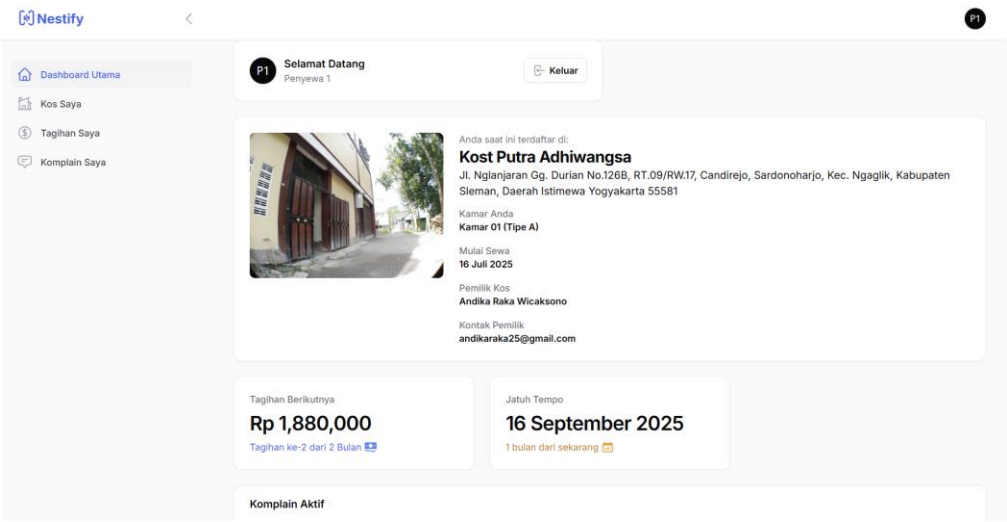
Dashboard Utama Penyewa

Gambar 4.12 menampilkan antarmuka Dashboard Utama bagi Penyewa kos setelah berhasil masuk ke dalam sistem. Pada kondisi awal, jika penyewa belum terdaftar di kos manapun, dasbor akan menampilkan pesan "Anda Belum Terdaftar di Kos Manapun" dan menyediakan tombol "Cari Kos Sekarang" untuk memulai pencarian. Bagian bawah dasbor juga menampilkan ringkasan "Komplain Aktif". Desain ini bertujuan untuk memandu penyewa yang baru bergabung untuk menemukan dan mendaftar ke kos, serta memberikan gambaran cepat tentang status komplain mereka.



Gambar 4.12 Tampilan Dasbor Utama Penyewa (Belum Terdaftar di Kos)

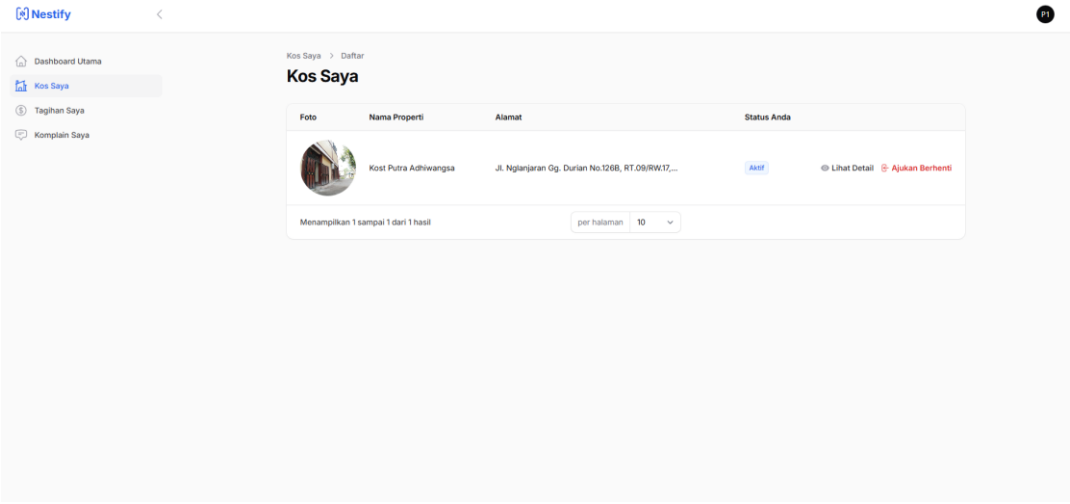
Gambar 4.13 menampilkan antarmuka Dashboard Utama bagi Penyewa kos setelah mereka berhasil terdaftar dan dikonfirmasi di sebuah kos. Dasbor ini menyajikan informasi penting mengenai kos yang sedang ditempati, termasuk nama kos, alamat lengkap, kamar yang dihuni, tanggal mulai sewa, serta kontak pemilik kos. Selain itu, dasbor juga menampilkan ringkasan "Tagihan Berikutnya" dengan nominal dan tanggal jatuh tempo, serta bagian "Komplain Aktif". Tampilan ini memberikan penyewa gambaran menyeluruh tentang status sewa dan keuangan mereka secara *real-time*.



Gambar 4.13 Tampilan Dasbor Penyewa (Terdaftar di Kos)

Halaman Kos Saya

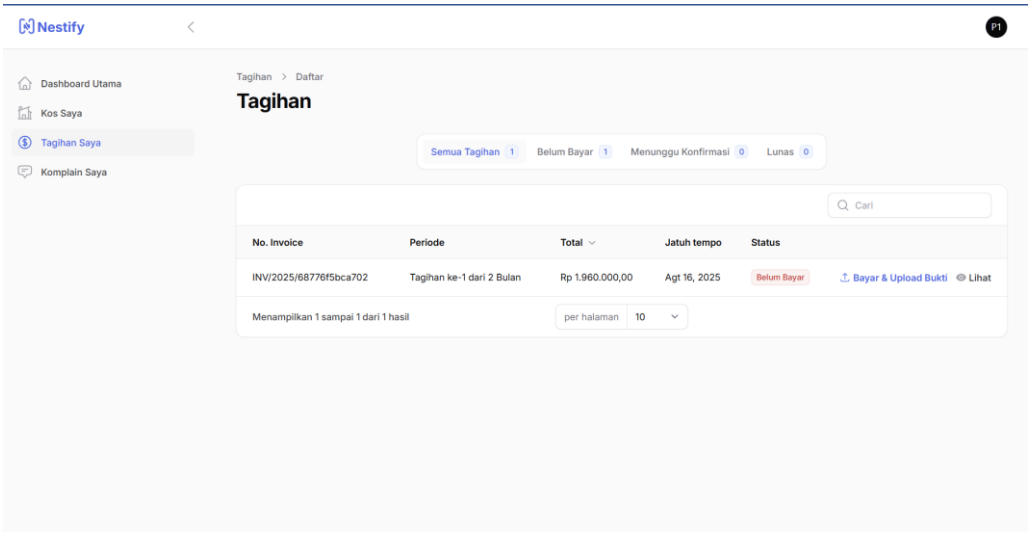
Gambar 4.14 menampilkan halaman awal “Kos Saya” bagi penyewa. Jika sudah menempati kos, halaman ini menampilkan detail kos yang dihuni. Dan jika belum terdaftar, tersedia opsi untuk mencari kos. Penyewa dapat melihat detail kos melalui pop-up, serta mengajukan permohonan berhenti sewa jika tidak memiliki tagihan tertunda. Pengajuan dilakukan lewat pop-up yang meminta alasan dan tanggal keluar. Setelah terkirim, status akan diperbarui, dan bila disetujui oleh pemilik, tampilan akan berubah menjadi “Belum Menempati Kos”.



Gambar 4.14 Tampilan Halaman Kos Saya

Halaman Tagihan Saya

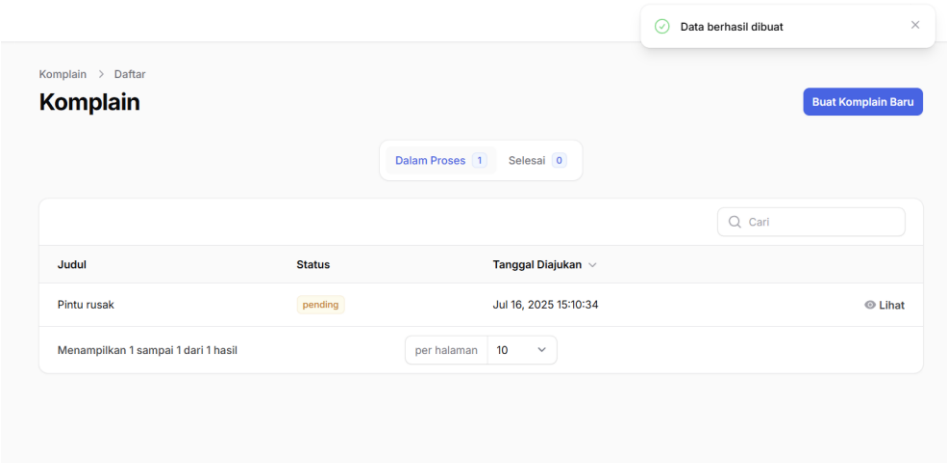
Gambar 4.15 menampilkan halaman “Tagihan Saya” bagi penyewa, yang memuat daftar tagihan dalam empat tab: “Semua Tagihan”, “Belum Bayar”, “Menunggu Konfirmasi”, dan “Lunas”. Setelah terdaftar di kamar kos tertentu, penyewa dapat melihat detail tagihan, melakukan pembayaran, mengunggah bukti, serta menggunakan promo jika tersedia. Setelah bukti diunggah, status tagihan akan diperbarui secara otomatis, termasuk untuk tagihan berulang yang aktif setelah pelunasan. Fitur ini membantu penyewa memantau dan menyelesaikan kewajiban pembayaran dengan mudah dan transparan.



Gambar 4.15 Tampilan Halaman Tagihan Saya

Halaman Komplain Saya

Gambar 4.16 menunjukkan halaman “Komplain Saya” bagi penyewa yang menampilkan daftar keluhan dalam dua tab: “Dalam Proses” dan “Selesai”, serta opsi untuk membuat komplain baru. Penyewa dapat mengajukan komplain melalui tombol “Buat Komplain Baru”, yang setelah berhasil akan langsung muncul di daftar dan memperbarui indikator angka. Selain itu, penyewa bisa melihat detail komplain serta memantau statusnya saat diproses atau setelah diselesaikan oleh pemilik. Fitur ini memudahkan penyewa mengelola keluhan secara langsung dan terpantau.

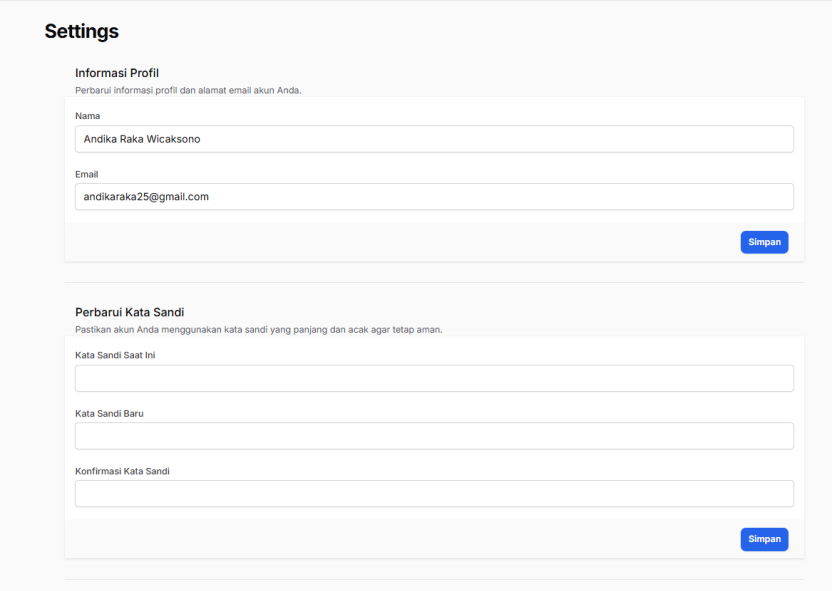


Gambar 4.16 Tampilan Halaman Komplain Saya

4.1.4 Antarmuka Pengaturan Akun

Halaman *Settings*

Gambar 4.17 menampilkan halaman "*Settings*" yang diakses melalui menu *drop-down* di pojok kanan atas. Halaman ini menyediakan opsi bagi pemilik dan penyewa untuk mengelola profil, memperbarui nama dan email, serta mengganti kata sandi dengan verifikasi. Pengguna juga dapat mengelola sesi aktif di perangkat lain, keluar dari sesi tertentu demi keamanan, dan menghapus akun secara permanen setelah konfirmasi. Selain itu, tersedia pusat bantuan yang memuat FAQ dan kontak dukungan jika diperlukan.



Gambar 4.17 Tampilan Halaman Settings

4.2 Hasil Pengujian Fungsional

Pada tahap ini, dilakukan pengujian fungsional dengan metode *Black-box Testing* untuk memverifikasi bahwa setiap fitur aplikasi berjalan sesuai dengan spesifikasi kebutuhan yang telah dirumuskan. Penelitian ini menerapkan pendekatan strategis dalam memilih modul yang akan diuji, berdasarkan prinsip prioritas fungsional dan analisis risiko.

Untuk sisi Pemilik kos, modul yang diuji adalah Registrasi Akun, Login Akun, Kelola Properti, Data Penghuni, dan Manajemen Tagihan. Untuk memberikan evaluasi yang seimbang, pengujian juga dilakukan pada fungsionalitas utama dari sisi Penyewa kos, modul yang dipilih adalah Pendaftaran Sewa, Pembayaran Tagihan, dan Pengajuan Komplain.

Pengujian dilakukan dengan teknik *Boundary Value Analysis* (BVA) untuk merancang kasus uji yang efisien dan efektif. Berikut adalah hasil eksekusi dari setiap kasus uji yang telah dirancang.

4.2.1 Hasil Pengujian Modul Registrasi Dan *Login* Akun

Hasil Pengujian Modul Registrasi

ID Test Case	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Status
TC_REGIS_PK_R01	Registrasi dengan semua data valid.	Berhasil registrasi, email verifikasi terkirim.	Berhasil registrasi, email verifikasi terkirim.	Pass
TC_REGIS_PK_R02	Registrasi dengan *field* Nama Lengkap kosong.	Gagal, muncul pesan validasi "Nama Lengkap tidak boleh kosong."	Gagal, muncul pesan validasi "Nama Lengkap tidak boleh kosong."	Pass
TC_REGIS_PK_R03	Registrasi dengan *field* Email kosong.	Gagal, muncul pesan validasi "Email tidak boleh kosong."	Gagal, muncul pesan validasi "Email tidak boleh kosong."	Pass
TC_REGIS_PK_R04	Registrasi dengan *field* Kata Sandi kosong.	Gagal, muncul pesan validasi "Kata Sandi tidak boleh kosong."	Gagal, muncul pesan validasi "Kata Sandi tidak boleh kosong."	Pass
TC_REGIS_PK_R05	Registrasi dengan Konfirmasi Kata Sandi kosong.	Gagal, pesan validasi "Konfirmasi Kata Sandi tidak boleh kosong."	Gagal, pesan validasi "Konfirmasi Kata Sandi tidak boleh kosong."	Pass
TC_REGIS_PK_R06	Registrasi dengan format email salah.	Gagal, pesan validasi "Format email tidak valid."	Gagal, pesan validasi "Format email tidak valid."	Pass
TC_REGIS_PK_R07	Registrasi dengan email yang sudah terdaftar.	Gagal, muncul pesan "Email sudah terdaftar."	Gagal, muncul pesan validasi "Email sudah terdaftar."	Pass

TC_REGIS_PK_R08	Registrasi dengan kata sandi lemah (kurang dari 8 karakter).	Gagal, pesan validasi panjang atau kompleksitas kata sandi.	Gagal, pesan validasi "Kata sandi minimal 8 karakter."	Pass
TC_REGIS_PK_R09	Registrasi dengan Konfirmasi Kata Sandi tidak cocok.	Gagal, pesan "Konfirmasi kata sandi tidak cocok."	Gagal, pesan "Konfirmasi kata sandi tidak cocok."	Pass
TC_REGIS_PK_R10	Registrasi dengan Nama Lengkap di bawah batas minimal.	Gagal, pesan validasi panjang Nama Lengkap.	Gagal, pesan validasi "Nama Lengkap minimal 3 karakter."	Pass
TC_REGIS_PK_R11	Registrasi dengan Email di bawah batas minimal.	Gagal, pesan validasi panjang Email.	Gagal, pesan validasi "Email minimal 6 karakter."	Pass
TC_REGIS_PK_R12	Registrasi dengan Kata Sandi di atas batas maksimal.	Gagal, pesan validasi panjang Kata Sandi.	Gagal, pesan validasi "Kata Sandi maksimal 32 karakter."	Pass
TC_REGIS_PK_R13	Registrasi tanpa memilih peran.	Gagal, pesan validasi "Pilihan Peran harus dipilih."	Gagal, pesan validasi "Silakan pilih peran Anda."	Pass

Tabel 4.1 Hasil Pengujian Modul Registrasi Akun

Hasil pengujian pada modul Registrasi Akun menunjukkan bahwa seluruh 13 kasus uji yang dieksekusi berhasil (Pass). Hal ini menunjukkan bahwa sistem berhasil menangani seluruh skenario registrasi, baik untuk input valid maupun kasus negatif seperti data kosong, format tidak sesuai, hingga validasi panjang karakter dan kecocokan kata sandi. Validasi pesan kesalahan ditampilkan secara konsisten dan sesuai spesifikasi, sementara proses registrasi berjalan baik saat seluruh data valid dimasukkan.

Hasil Pengujian Modul Login Akun.

ID Test Case	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Status
TC_LOGIN_PK_R01	Login dengan email dan kata sandi valid.	Berhasil Login, diarahkan ke dashboard.	Berhasil Login, diarahkan ke dashboard.	Pass
TC_LOGIN_PK_R02	Login dengan kata sandi salah.	Gagal, pesan "Email atau Kata Sandi salah."	Gagal, muncul pesan "Invalid password".	Pass
TC_LOGIN_PK_R03	Login dengan email tidak terdaftar.	Gagal, pesan "Email atau Kata Sandi salah."	Gagal, muncul pesan "Invalid email".	Pass
TC_LOGIN_PK_R04	Login dengan email kosong.	Gagal, pesan validasi "Email	Gagal, muncul pesan validasi.	Pass

		tidak boleh kosong."		
TC_LOGIN_PK_R05	Login dengan kata sandi kosong.	Gagal, pesan validasi "Kata Sandi tidak boleh kosong."	Gagal, muncul pesan validasi.	Pass
TC_LOGIN_PK_R06	Login dengan format email salah.	Gagal, pesan validasi "Format email tidak valid."	Gagal, muncul pesan validasi.	Pass
TC_LOGIN_PK_R07	Login dengan email di bawah batas minimal.	Gagal, pesan validasi panjang email.	Gagal, muncul pesan validasi.	Pass
TC_LOGIN_PK_R08	Login dengan kata sandi di bawah batas minimal.	Gagal, pesan validasi panjang Kata Sandi.	Gagal, muncul pesan validasi.	Pass
TC_LOGIN_PK_R09	Login dengan email di atas batas maksimal.	Gagal, pesan validasi panjang email.	Tidak ada pesan validasi, proses *loading* terus-menerus.	Fail
TC_LOGIN_PK_R10	Login dengan kata sandi di atas batas maksimal.	Gagal, pesan validasi panjang Kata Sandi.	Gagal, muncul pesan validasi.	Pass

Tabel 4.2 Hasil Pengujian Modul Login

Hasil pengujian pada modul Login menunjukkan 9 dari 10 kasus uji berhasil (Pass). Kegagalan pada TC_LOGIN_PK_R09 mengindikasikan kelemahan dalam menangani input yang melebihi batas maksimal, yang dapat memengaruhi stabilitas sistem.

4.2.2 Hasil Pengujian Modul Tampilan Pemilik

Hasil Pengujian Modul Kelola Properti

ID Test Case	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Status
TC_KP_PK_R01	Tambah properti dengan semua data valid.	Properti baru berhasil disimpan dan tampil di daftar.	Properti baru berhasil disimpan dan tampil di daftar.	Pass
TC_KP_PK_R02	Tambah properti dengan Nama Properti kosong.	Gagal, pesan validasi "Nama Properti tidak boleh kosong."	Gagal, pesan validasi "Nama Properti tidak boleh kosong."	Pass
TC_KP_PK_R03	Tambah properti dengan Alamat Properti kosong.	Gagal, pesan validasi "Alamat Properti tidak boleh kosong."	Gagal, pesan validasi "Alamat Properti tidak boleh kosong."	Pass
TC_KP_PK_R04	Tambah properti dengan format foto salah.	Gagal, pesan validasi "Format file foto tidak didukung."	Sistem menerima file, properti tersimpan namun foto rusak.	Fail

TC_KP_PK_R05	Lihat detail properti yang valid.	Berhasil menampilkan halaman detail dengan informasi benar.	Berhasil menampilkan halaman detail dengan informasi benar.	Pass
TC_KP_PK_R06	Edit properti dengan data valid.	Perubahan berhasil disimpan, data ter-update.	Perubahan berhasil disimpan, data ter-update.	Pass
TC_KP_PK_R07	Edit properti dan mengosongkan Nama Properti.	Gagal, pesan validasi "Nama Properti tidak boleh kosong."	Perubahan berhasil disimpan meskipun Nama Properti kosong.	Fail
TC_KP_PK_R08	Edit properti dengan format Harga Sewa salah (input teks).	Gagal, pesan validasi "Format Harga Sewa tidak valid."	Gagal, pesan "Masukkan angka yang valid untuk Harga Sewa."	Pass
TC_KP_PK_R09	Hapus properti yang valid dengan konfirmasi "Ya".	Properti berhasil dihapus dari daftar.	Properti berhasil dihapus dari daftar.	Pass
TC_KP_PK_R10	Hapus properti namun memilih "Batal" pada konfirmasi.	Penghapusan dibatalkan, properti tetap ada.	Penghapusan dibatalkan, properti tetap ada.	Pass

Tabel 4.3 Hasil Pengujian Modul Tampilan Pemilik Kelola Properti

Hasil pengujian pada modul Kelola Properti menunjukkan 8 dari 10 kasus uji berhasil (Pass). Dua kegagalan pada TC_KP_PK_R04 dan TC_KP_PK_R07 menyoroti kelemahan pada validasi sisi server untuk format file dan *field* wajib, yang dapat mengganggu integritas data properti.

Hasil Pengujian Modul Data Penghuni

ID Test Case	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Status
TC_DP_PK_01	Pemilik menambahkan data penghuni baru secara manual dengan semua data valid.	Penghuni baru berhasil dibuat dan muncul di daftar "Penghuni Aktif".	Penghuni baru berhasil dibuat dan tercatat dengan benar.	Pass
TC_DP_PK_02	Pemilik menambahkan data penghuni baru namun mengosongkan *field* wajib (Nama Penghuni).	Gagal, sistem menampilkan pesan validasi "Nama Penghuni tidak boleh kosong."	Gagal, sistem menampilkan pesan validasi yang sesuai.	Pass

TC_DP_PK_03	Pemilik menerima pengajuan pendaftar baru yang valid.	Status pendaftar berubah menjadi "Aktif" dan datanya pindah ke modul "Data Penghuni".	Proses konfirmasi berjalan lancar, data penghuni ter-update.	Pass
TC_DP_PK_04	Pemilik mengedit data penghuni dengan informasi yang valid.	Data penghuni berhasil diperbarui di dalam sistem.	Data penghuni berhasil diperbarui.	Pass
TC_DP_PK_05	Pemilik menghapus data penghuni yang sudah tidak aktif.	Data penghuni berhasil dihapus dari daftar.	Data penghuni berhasil dihapus.	Pass
TC_DP_PK_06	Pemilik mencoba menambahkan penghuni baru ke kamar yang statusnya sudah "Terisi".	Gagal, sistem menampilkan pesan error "Kamar sudah terisi dan tidak tersedia."	Gagal, sistem tidak menampilkan kamar yang statusnya “Terisi” pada pilihan.	Pass
TC_DP_PK_07	Pemilik mengedit nomor telepon penghuni dengan format tidak valid (mengandung huruf).	Gagal, sistem menampilkan pesan validasi "Format nomor telepon tidak valid."	Gagal, sistem menampilkan pesan validasi yang sesuai.	Pass
TC_DP_PK_08	Pemilik menggunakan filter untuk menampilkan hanya penghuni "Aktif".	Tabel hanya menampilkan data penghuni dengan status "Aktif".	Filter berfungsi dengan benar.	Pass
TC_DP_PK_09	Pemilik menggunakan fitur pencarian untuk menemukan penghuni berdasarkan nama.	Tabel menampilkan hasil pencarian yang sesuai dengan nama yang diinput.	Fitur pencarian berfungsi dengan benar.	Pass
TC_DP_PK_10	Pemilik mengonfirmasi pengajuan berhenti sewa dari penghuni.	Status penghuni berubah menjadi "Tidak Aktif" dan status kamar menjadi "Kosong".	Status penghuni dan kamar berhasil diperbarui.	Pass

Tabel 4.4 Hasil Pengujian Modul Tampilan Pemilik Data Penghuni

Hasil pengujian pada modul Data Penghuni menunjukkan bahwa seluruh 10 kasus uji yang dieksekusi berhasil (Pass). Hal ini mengindikasikan bahwa fungsionalitas manajemen data penghuni, mulai dari penambahan, validasi input, konfirmasi pengajuan, hingga penanganan logika bisnis seperti ketersediaan kamar, telah berjalan dengan sangat baik dan sesuai dengan spesifikasi yang diharapkan.

Hasil Pengujian Modul Manajemen Tagihan

ID Test Case	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Status
TC_MT_PK_01	Pemilik mengonfirmasi masuknya penghuni baru.	Tagihan pertama untuk penghuni tersebut otomatis dibuat dengan status "Belum Bayar".	Tagihan pertama berhasil dibuat secara otomatis dan muncul di daftar.	Pass
TC_MT_PK_02	Pemilik mengubah status tagihan dari "Butuh Konfirmasi" menjadi "Lunas".	Status tagihan berhasil diubah menjadi "Lunas" dan pindah ke tab "Lunas".	Status tagihan berhasil diubah dan berpindah tab dengan benar.	Pass
TC_MT_PK_03	Pemilik mengubah status tagihan "Lunas" untuk penyewa dengan durasi sewa >1 bulan.	Tagihan untuk periode berikutnya otomatis dibuat dengan status "Belum Bayar".	Tagihan periode berikutnya berhasil dibuat secara otomatis.	Pass
TC_MT_PK_04	Pemilik menggunakan filter untuk menampilkan tagihan dengan status "Belum Bayar".	Hanya tagihan dengan status "Belum Bayar" yang ditampilkan di tabel.	Filter berfungsi dengan benar, hanya menampilkan tagihan yang relevan.	Pass
TC_MT_PK_05	Pemilik menggunakan filter untuk menampilkan tagihan dengan status "Lunas".	Hanya tagihan dengan status "Lunas" yang ditampilkan di tabel.	Filter berfungsi dengan benar, hanya menampilkan tagihan yang relevan.	Pass
TC_MT_PK_06	Pemilik mencoba mengubah status tagihan "Belum Bayar" langsung menjadi "Lunas".	Aksi ditolak atau opsi tidak tersedia, karena harus melalui status "Butuh Konfirmasi" terlebih dahulu.	Status berhasil diubah langsung menjadi "Lunas" tanpa verifikasi bukti bayar.	Fail

Tabel 4.5 Hasil Pengujian Modul Tampilan Pemilik Manajemen Tagihan

Hasil pengujian pada modul Manajemen Tagihan menunjukkan 5 dari 6 kasus uji berhasil (Pass). Kegagalan pada TC_MT_PK_06 mengindikasikan adanya celah dalam alur logika bisnis, di mana sistem memperbolehkan perubahan status yang tidak semestinya, berisiko pada akurasi pencatatan keuangan.

4.2.3 Hasil Pengujian Modul Tampilan Penyewa

Hasil Pengujian Modul Pendaftaran Properti

ID Test Case	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Status
TC_PY_REG_R01	Penyewa mengajukan pendaftaran dengan mengisi semua data formulir secara valid.	Pengajuan pendaftaran berhasil terkirim ke pemilik dan penyewa menerima notifikasi sukses.	Pengajuan pendaftaran berhasil terkirim, muncul notifikasi sukses.	Pass
TC_PY_REG_R02	Penyewa mengajukan pendaftaran namun mengosongkan salah satu *field* wajib (contoh: durasi sewa).	Pengajuan gagal terkirim, sistem menampilkan pesan validasi "Durasi sewa wajib diisi."	Gagal, sistem menampilkan pesan validasi yang sesuai pada *field* yang kosong.	Pass
TC_PY_REG_R03	Penyewa memasukkan data dengan format tidak valid pada *field* durasi sewa (contoh: "satu bulan" bukan angka).	Pengajuan gagal, sistem menampilkan pesan validasi "Durasi harus berupa angka."	Gagal, sistem menampilkan pesan validasi format yang sesuai.	Pass
TC_PY_REG_R04	Penyewa mencoba mendaftar ke kamar yang statusnya sudah "Terisi" atau "Tidak Tersedia".	Sistem menolak pendaftaran dan menampilkan pesan "Kamar yang dipilih tidak tersedia."	Sistem berhasil menolak pendaftaran dan menampilkan pesan ketersediaan kamar.	Pass
TC_PY_REG_R05	Penyewa mengunggah foto KTP dengan format yang tidak didukung (contoh: .gif).	Pengajuan gagal, sistem menampilkan pesan validasi "Format foto KTP tidak valid."	Gagal, sistem menampilkan pesan validasi format file.	Pass

Tabel 4.6 Hasil Pengujian Modul Tampilan Penyewa Pendaftaran Properti

Hasil pengujian pada modul Pendaftaran Sewa menunjukkan bahwa seluruh (5 dari 5) kasus uji berhasil (Pass). Hal ini mengindikasikan bahwa alur pendaftaran penyewa, termasuk validasi input dasar dan pengecekan ketersediaan kamar, telah berfungsi sesuai dengan yang diharapkan.

Hasil Pengujian Modul Pembayaran Tagihan Penyewa

ID Test Case	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Status
TC_PY_PAY_01	Penyewa membayar tagihan dengan mengunggah bukti bayar berformat valid (JPG/PNG).	Bukti pembayaran berhasil diunggah, status tagihan berubah menjadi "Butuh Konfirmasi".	Berhasil, bukti terunggah dan status tagihan ter-update di sisi penyewa dan pemilik.	Pass
TC_PY_PAY_02	Penyewa mencoba mengunggah bukti bayar dengan format tidak valid.	Sistem menolak file dan menampilkan pesan error "Format file tidak didukung."	Sistem menerima file tanpa validasi. File yang terunggah rusak/tidak bisa dibuka oleh pemilik.	Fail
TC_PY_PAY_03	Penyewa mencoba mengirim pembayaran tanpa mengunggah bukti bayar.	Proses gagal, sistem menampilkan pesan validasi "Bukti pembayaran wajib diunggah."	Gagal, sistem menampilkan pesan validasi yang sesuai.	Pass
TC_PY_PAY_04	Penyewa mengunggah file bukti pembayaran yang ukurannya melebihi batas maksimal yang ditentukan.	Unggah gagal, sistem menampilkan pesan error "Ukuran file terlalu besar."	Unggah gagal, sistem menampilkan pesan error ukuran file.	Pass
TC_PY_PAY_05	Penyewa menerapkan kode promo yang valid saat pembayaran.	Total tagihan berhasil dikurangi sesuai nilai promo, dan pembayaran berhasil diproses.	Total tagihan terpotong dengan benar dan pembayaran berhasil.	Pass
TC_PY_PAY_06	Penyewa mencoba menerapkan kode promo yang tidak valid atau sudah kedaluwarsa.	Sistem menampilkan pesan "Kode promo tidak valid atau sudah tidak berlaku."	Sistem berhasil menampilkan pesan error kode promo.	Pass

Tabel 4.7 Hasil Pengujian Modul Tampilan Penyewa Pembayaran Tagihan Penyewa

Hasil pengujian pada modul Pembayaran Tagihan menunjukkan 5 dari 6 kasus uji berhasil (Pass). Kegagalan pada TC_PY_PAY_02, yang serupa dengan temuan pada modul Kelola Properti, menegaskan adanya kelemahan sistemik pada validasi format unggahan file di seluruh aplikasi.

Hasil Pengujian Modul Pengajuan Komplain Penyewa

ID Test Case	Skenario Uji	Hasil yang Diharapkan	Hasil Aktual	Status
TC_PY_COMP_01	Penyewa mengajukan komplain dengan mengisi judul dan deskripsi secara valid.	Komplain berhasil terkirim dan muncul di daftar komplain aktif penyewa serta di dashboard pemilik.	Komplain berhasil terkirim dan tercatat dengan benar.	Pass
TC_PY_COMP_02	Penyewa mengajukan komplain namun mengosongkan *field* deskripsi yang wajib diisi.	Pengajuan komplain gagal, sistem menampilkan pesan validasi "Deskripsi komplain tidak boleh kosong."	Gagal, sistem menampilkan pesan validasi yang sesuai.	Pass
TC_PY_COMP_03	Penyewa mengajukan komplain dengan judul dan deskripsi yang sangat panjang (melebihi batas karakter).	Pengajuan gagal, sistem menampilkan pesan validasi "Jumlah karakter melebihi batas."	Gagal, sistem menampilkan pesan validasi batas karakter.	Pass
TC_PY_COMP_04	Penyewa mengajukan komplain dengan melampirkan file gambar yang valid (JPG/PNG).	Komplain berhasil terkirim beserta lampirannya, dan dapat dilihat oleh pemilik.	Komplain dan lampiran berhasil terkirim.	Pass
TC_PY_COMP_05	Penyewa mencoba melampirkan file dengan format tidak valid pada komplain.	Sistem menolak lampiran dan menampilkan pesan error "Format lampiran tidak didukung."	Sistem berhasil menolak lampiran dan menampilkan pesan error yang sesuai.	Pass

Tabel 4.8 Hasil Pengujian Modul Tampilan Penyewa Pengajuan Komplain Penyewa

Hasil pengujian pada modul Pengajuan Komplain menunjukkan bahwa seluruh (5 dari 5) kasus uji berhasil (Pass), mengindikasikan bahwa fungsionalitas ini telah solid dan berjalan sesuai spesifikasi.

4.3 Pembahasan

Bagian ini berisi analisis dan interpretasi terhadap data hasil pengujian fungsional yang telah disajikan sebelumnya pada sub-bab 4.2. Pembahasan ini bertujuan untuk mengevaluasi kinerja aplikasi website Nestify berdasarkan kriteria keberhasilan yang telah ditetapkan, serta mengidentifikasi area-area yang memerlukan perbaikan lebih lanjut.

4.3.1 Ringkasan Hasil Pengujian Modul

Pengujian fungsional aplikasi website Nestify telah dilaksanakan secara komprehensif pada delapan modul utama, baik dari sisi Pemilik kos maupun Penyewa. Berikut adalah ringkasan hasil pengujian untuk setiap modul:

1. Modul Registrasi Akun

   Seluruh 13 kasus uji berhasil (*Pass*). Ini menunjukkan fungsionalitas Registrasi Akun telah berjalan dengan sangat baik dan sesuai spesifikasi.

2. Modul Login Akun

   Dari 10 kasus uji, 9 kasus berhasil (*Pass*) dan 1 kasus gagal (*Fail*). Kegagalan teridentifikasi pada skenario TC_LOGIN_PK_R09, yaitu penanganan *input* email yang melebihi batas maksimal karakter.

3. Modul Kelola Properti

   Dari 10 kasus uji, 8 kasus berhasil (*Pass*) dan 2 kasus gagal (*Fail*). Kegagalan terjadi pada skenario TC_KP_PK_R04 (format foto salah) dan TC_KP_PK_R07 (pengosongan *field* Nama Properti saat edit).

4. Modul Data Penghuni

   Seluruh 10 kasus uji berhasil (*Pass*). Ini menunjukkan fungsionalitas manajemen data penghuni telah berjalan dengan sangat baik dan sesuai spesifikasi.

5. Modul Manajemen Tagihan

   Dari 6 kasus uji, 5 kasus berhasil (*Pass*) dan 1 kasus gagal (*Fail*). Kegagalan terjadi pada skenario TC_MT_PK_06, di mana sistem memperbolehkan perubahan status tagihan "Belum Bayar" langsung menjadi "Lunas" tanpa verifikasi bukti bayar.

6. Modul Pendaftaran Sewa (Sisi Penyewa)

   Seluruh 5 kasus uji berhasil (*Pass*). Ini mengindikasikan alur pendaftaran penyewa telah berfungsi sesuai harapan.

7. Modul Pembayaran Tagihan Penyewa Kos

   Dari 6 kasus uji, 5 kasus berhasil (*Pass*) dan 1 kasus gagal (*Fail*). Kegagalan terjadi pada skenario TC_PY_PAY_02, yaitu penanganan unggah bukti bayar dengan format tidak valid.

8. Modul Pengajuan Komplain Penyewa Kos

   Seluruh 5 kasus uji berhasil (*Pass*). Ini menunjukkan fungsionalitas pengajuan komplain telah solid dan berjalan sesuai spesifikasi.

4.3.2 Analisis Kinerja Pengujian Keseluruhan

Berdasarkan hasil pengujian yang komprehensif pada seluruh modul fungsionalitas yang dievaluasi, total kasus uji yang dieksekusi adalah 65 kasus. Dari jumlah tersebut, sebanyak 60 kasus uji dinyatakan *Berhasil (Pass)*, sementara 5 kasus uji dinyatakan *Gagal (Fail)*. Mengacu pada formula persentase keberhasilan yang telah dijelaskan pada sub-bab 3.4.3 Kriteria Keberhasilan dan Penanganan Hasil, perhitungan kinerja keseluruhan adalah sebagai berikut:

$$N = \frac{SP}{SM} \times 100\% \quad (2)$$

$$N = \frac{60}{65} \times 100\%$$

$$N = 92.3\%$$

Keterangan:

- N = Persentase keberhasilan fungsionalitas sistem.
- SP = Jumlah kasus uji yang dinyatakan Berhasil *(Pass)*.
- SM = Total seluruh kasus uji yang dieksekusi.

Dengan persentase keberhasilan sebesar 92.3%, kinerja fungsionalitas inti aplikasi website Nestify berada dalam kategori **"Sangat Baik"** (89% – 100%). Hal ini mengindikasikan bahwa secara umum, aplikasi telah beroperasi dengan sangat stabil dan hampir sepenuhnya memenuhi semua persyaratan fungsional yang ditetapkan. Mayoritas fitur inti berjalan sesuai ekspektasi, menunjukkan fondasi sistem yang solid dan keandalan yang tinggi dalam skenario penggunaan yang diuji.

4.3.3 Analisis Kegagalan Fungsional

Meskipun kinerja keseluruhan tergolong sangat baik, temuan kegagalan pada 6 kasus uji memberikan wawasan penting mengenai area yang memerlukan perbaikan. Analisis mendalam terhadap kegagalan tersebut adalah sebagai berikut:

1. Validasi Batas Maksimal Email (Modul Login - TC_LOGIN_PK_R09)

   Ditemukan bahwa sistem gagal memberikan pesan peringatan atau validasi yang semestinya ketika *input* pada *field* email melebihi batas maksimal karakter yang

ditentukan. Sistem tidak memberikan respons validasi yang dapat membingungkan pengguna dan berpotensi menimbulkan kerentanan.

2. Validasi Format *File* Foto (Modul Kelola Properti - TC_KP_PK_R04)

   Pada fitur Tambah Properti, sistem gagal melakukan validasi format *file* foto secara akurat. Sistem menerima *file* dengan format yang tidak seharusnya (misalnya .pdf) tanpa menampilkan pesan kesalahan yang sesuai. Hal ini berpotensi menyebabkan gambar tidak tampil dengan benar atau bahkan merusak tampilan properti.

3. Validasi *Field* Wajib (Modul Kelola Properti - TC_KP_PK_R07)

   Pada fitur Edit Properti, sistem memperbolehkan penyimpanan data meskipun *field* wajib seperti "Nama Properti" dikosongkan. Ini menunjukkan kelemahan pada aturan validasi untuk *field* yang krusial, yang dapat mengakibatkan data properti tidak lengkap atau tidak konsisten.

4. Validasi Alur Perubahan Status Tagihan (Modul Manajemen Tagihan - TC_MT_PK_06)

   Ditemukan adanya celah dalam alur logika bisnis terkait perubahan status tagihan. Sistem memperbolehkan perubahan status tagihan dari "Belum Bayar" langsung menjadi "Lunas" tanpa melalui status "Butuh Konfirmasi" atau verifikasi bukti bayar. Hal ini berisiko pada akurasi pencatatan keuangan dan dapat disalahgunakan.

5. Validasi Format *File* Bukti Pembayaran (Modul Pembayaran Tagihan Penyewa Kos - TC_PY_PAY_02)

   Serupa dengan temuan pada Modul Kelola Properti, sistem gagal melakukan validasi format *file* bukti pembayaran yang diunggah oleh penyewa. Sistem menerima *file* dengan format tidak valid, yang mengakibatkan *file* yang terunggah rusak atau tidak dapat dibuka oleh pemilik, sehingga proses verifikasi pembayaran terhambat.

Kegagalan-kegagalan ini, meskipun jumlahnya relatif kecil dibandingkan total kasus uji yang berhasil, menunjukkan area-area kritis yang memerlukan perbaikan segera untuk meningkatkan keandalan dan integritas data aplikasi *Nestify*.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan Berdasarkan seluruh tahapan penelitian yang telah dilaksanakan, mulai dari analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian fungsional aplikasi website Nestify, dapat ditarik beberapa kesimpulan utama sebagai berikut:

1. Aplikasi website Nestify berhasil dirancang, dikembangkan, dan diimplementasikan secara terstruktur menggunakan metode *Research and Development* (R&D) berbasis model *Waterfall*. Setiap tahap metodologi ini, dari analisis kebutuhan hingga pemeliharaan, berperan krusial dalam memastikan sistem yang kohesif, fungsional, dan sesuai kebutuhan pengguna, didukung oleh implementasi *framework* Laravel 12, Laravel Filament, dan Laravel Fortify.
2. Fungsionalitas utama aplikasi telah diuji secara komprehensif melalui *Black-box Testing* dengan teknik *Boundary Value Analysis* (BVA). Dari total 65 kasus uji yang dieksekusi, 60 kasus berhasil (*Pass*), menghasilkan *Success Rate* 92.3%. Kinerja ini menempatkan aplikasi dalam kategori "Sangat Baik", secara kuat mengonfirmasi kesesuaian fungsionalitas dengan kebutuhan pengguna serta mendukung pengelolaan properti kos yang efisien dan andal.
3. Aplikasi *Nestify* menyediakan platform terpadu yang secara signifikan mempermudah pengelolaan indekos bagi pemilik properti, sekaligus memberikan kemudahan bagi penyewa dalam mengakses informasi sewa, memantau status pembayaran, dan mengajukan keluhan secara digital dalam satu platform website. Pemanfaatan teknologi modern dan fleksibilitas konfigurasi harga sewa, biaya tambahan, serta diskon, menegaskan kekuatan sistem ini dalam adaptasi penawaran yang beragam.
4. Proyek pengembangan aplikasi website Nestify ini telah berhasil mencapai tujuannya dalam menciptakan solusi manajemen indekos yang fungsional dan teruji. Hasil pengujian yang sangat positif memberikan wawasan berharga untuk perbaikan dan pengembangan di masa mendatang, menegaskan potensi besar aplikasi ini sebagai alat digital yang efisien dan andal dalam mendukung ekosistem indekos.

5.2 Saran

Menindaklanjuti hasil pengembangan aplikasi Nestify dan untuk memastikan relevansi jangka panjang, beberapa saran berikut dapat dipertimbangkan pada penelitian selanjutnya guna meningkatkan fungsionalitas dan pengalaman pengguna pada website Nestify.

1. Untuk penelitian selanjutnya, diharapkan untuk mengeksplorasi pengembangan fitur multi pemilik yang mampu mengakomodasi pengelolaan properti oleh banyak pemilik dengan isolasi data yang terjamin. Selain itu, penelitian lanjutan dapat berfokus pada integrasi sistem notifikasi otomatis dan fitur pesan (*messaging*) untuk komunikasi *real time* antara pemilik dan penyewa.
2. Penelitian mendatang juga dapat mengkaji peningkatan modul finansial dengan menambahkan laporan keuangan yang lebih detail dan analitis, serta mengintegrasikan *gateway* pembayaran *online*. Fokus dapat diberikan pada bagaimana implementasi ini dapat menyederhanakan proses pembayaran dan pembaruan status transaksi secara *real time*.
3. Saran untuk penelitian berikutnya juga mencakup penyempurnaan pengalaman pengguna (UX) melalui pengembangan dashboard penyewa yang lebih personal dan fitur pencarian serta filter yang lebih canggih. Selain itu, kajian dapat dilakukan terhadap penyusunan dokumentasi penggunaan yang lengkap dan mudah diakses untuk mendukung kemandirian pengguna.

DAFTAR PUSTAKA

Abdul Jalil, R., Najib Mohamed Razali, M., Zaireen Zainudin, A., Ahmad, F., Alam Bina, F., Teknologi Malaysia, U., & Kejuruteraan Awam, F. (2023). DIGITALIZATION OF SUSTAINABLE INTEGRATED PROPERTY MANAGEMENT FOR AFFORDABLE STRATA HOUSING. Dalam *Journal of the Malaysian Institute of Planners VOLUME* (Vol. 21).

Afiifah, K., Fira Azzahra, Z., & Anggoro, A. D. (2022). Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database: Sebuah Literature Review. *JURNAL INTECH*, *3*(1), 8–11.

Akbar, S., Indra, M., & Anugrah, G. (2022). *Sistem Pendukung Kepustusan Pemilihan Tempat Kos Untuk Mahasiswa Di Gresik Dengan Metode Saw*. *7*(2).

Al-Saqqa, S., Sawalha, S., & Abdelnabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, *14*(11), 246–270. https://doi.org/10.3991/ijim.v14i11.13269

Amalia, A., Putri Hamidah, S. W., & Kristanto, T. (2021). Pengujian Black Box Menggunakan Teknik Equivalence Partitions Pada Aplikasi E-Learning Berbasis Web. *Building of Informatics, Technology and Science (BITS)*, *3*(3), 269–274. https://doi.org/10.47065/bits.v3i3.1062

APJII 2024. (2024). *SURVEI PENETRASI INTERNET INDONESIA 2024*.

Asih, R. W. (2021). *Mengenal Startup Mamikos, Didirikan oleh Maria Regina Anggit Tut Pinilih*. https://teknologi.bisnis.com/read/20210924/266/1446362/mengenal-startup-mamikos-didirikan-oleh-maria-regina-anggit-tut-pinilih. https://teknologi.bisnis.com/read/20210924/266/1446362/mengenal-startup-mamikos-didirikan-oleh-maria-regina-anggit-tut-pinilih

Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *ArXiv*, *abs/1205.6904*. https://api.semanticscholar.org/CorpusID:5752605

Berenbrok, L. A., Tang, S., Gabriel, N., Guo, J., Sharareh, N., Patel, N., Dickson, S., & Hernandez, I. (2022). Access to community pharmacies: A nationwide geographic information systems cross-sectional analysis. *Journal of the American Pharmacists Association*, *62*(6), 1816-1822.e2. https://doi.org/10.1016/j.japh.2022.07.003

Brawijaya, U., Gibran, M. I., Purnomo, W., & Akbar, M. A. (2017). *Fakultas Ilmu Komputer PEMBANGUNAN APLIKASI PENGELOLAAN INDEKOS BERBASIS WEB (STUDI KASUS: INDEKOS MUSLIM BUNDA RILA)* (Vol. 1, Nomor 1). http://j-ptiik.ub.ac.id

Camburn, B., Viswanathan, V., Linsey, J., Anderson, D., Jensen, D., Crawford, R., Otto, K., & Wood, K. (2017). Design prototyping methods: State of the art in strategies, techniques, and guidelines. *Design Science*, *3*. https://doi.org/10.1017/dsj.2017.10

Cholik, C. A. (2021). *DALAM BERBAGAI BIDANG* (Vol. 2, Nomor 2).

Christoforakos, L., Tretter, S., Diefenbach, S., Bibi, S. A., Fröhner, M., Kohler, K., Madden, D., Marx, T., Pfeiffer, T., Pfeiffer-Leßmann, N., & Valkanova, N. (2019). Potential and challenges of prototyping in product development and innovation: Insights from an expert discussion among researchers and practitioners. *i-com*, *18*(2), 179–187. https://doi.org/10.1515/icom-2019-0010

D. Oktavallyan, S., Seftyarizki, D., & Wardhani, F. (2021). Student Boarding House Quality in term of Health Design (Case study: Boarding House near University of Bengkulu). *IOP Conference Series: Earth and Environmental Science*, *738*(1). https://doi.org/10.1088/1755-1315/738/1/012027

Damar, A. M. (2021). *Masuki Usia ke-6, Mamikos Hubungkan 6 Juta Penyewa Kos Tiap Bulan*. https://www.liputan6.com/tekno/read/4712308/masuki-usia-ke-6-mamikos-hubungkan-6-juta-penyewa-kos-tiap-bulan. https://www.liputan6.com/tekno/read/4712308/masuki-usia-ke-6-mamikos-hubungkan-6-juta-penyewa-kos-tiap-bulan

Danuri, M., Informatika, M., Teknologi, J., & Semarang, C. (2020). *PERKEMBANGAN DAN TRANSFORMASI TEKNOLOGI DIGITAL*.

Daraojimba, E. C., Chinedu, N. N., Abimbola, O. A., Ezeigweneme, A. C., & Joachim, O. G. (2024). COMPREHENSIVE REVIEW OF AGILE METHODOLOGIES IN PROJECT MANAGEMENT. *Computer Science & IT Research Journal*, *5*(1), 190–218. https://doi.org/10.51594/csitrj.v5i.717

Edityorini, E. (2022). *PENGARUH INTENSITAS MENGAKSES AKUN INSTAGRAM @MAMIKOSAPP TERHADAP PEMENUHAN KEBUTUHAN INFORMASI FOLLOWERS MENGENAI REKOMENDASI KOS*. https://e-journal.uajy.ac.id/27900/

Effendy, F., Kartono, & Herawatie, D. (2020). Mobile apps for boarding house recommendation. *International Journal of Interactive Mobile Technologies*, *14*(11), 32–47. https://doi.org/10.3991/ijim.v14i11.11574

Engineering, T. Q. R., Krisna, Y. M., Kusnadi, A., & Tobing, F. A. T. (2023). Recommendation System for Boarding House Selection using Simple Additive Weighting Method. *The Asian Institute of Research Engineering and Technology Quarterly Reviews*, *6*(1), 50–55. https://doi.org/10.31219/osf.io/98ngq

Fauzi, N. F. (2024). *IMPLEMENTASI METODE USER CENTERED DESIGN PADA PERANCANGAN ANTARMUKA APLIKASI INDEKOS*.

Febriana, D., & Iswari, L. (2023). PENGEMBANGAN APLIKASI PEMBELAJARAN BAHASA ARAB UNTUK PEMULA BERBASIS WEB. *Insect (Informatics and Security): Jurnal Teknik Informatika*, *8*(2), 100–109. https://doi.org/10.33506/insect.v8i2.2246

Frantiska, J. (2018). Entity-Relationship Diagrams. Dalam Jr. Frantiska Joseph (Ed.), *Visualization Tools for Learning Environment Development* (hlm. 21–30). Springer International Publishing. https://doi.org/10.1007/978-3-319-67440-7_4

Garousi, V., Mesbah, A., Betin-Can, A., & Mirshokraie, S. (2013). A systematic mapping study of web application testing. Dalam *Information and Software Technology* (Vol. 55, Nomor 8, hlm. 1374–1396). Elsevier B.V. https://doi.org/10.1016/j.infsof.2013.02.006

Joe, T. (2023). CLOUD COMPUTING AND INFORMATION SYSTEMS: ENABLING SCALABILITY AND FLEXIBILITY. Dalam *Business Studies Journal* (Vol. 15, Nomor 3).

Julianto, V. (2017). Rancang Bangun Sistem Informasi Geografis Tower Saluran Udara Tegangan Tinggi (SUTT) (Studi Kasus: PT PLN (Persero) AP2B Sistem Kalsel-Teng). Dalam *Jurnal Sains dan Informatika* (Vol. 3, Nomor 2).

Kadir, A. (2014). *Pengenalan Sistem Informasi Edisi Revisi*. https://doi.org/10.13140/2.1.2637.6328

Kramer, M. (2018). BEST PRACTICES IN SYSTEMS DEVELOPMENT LIFECYCLE: AN ANALYSES BASED ON THE WATERFALL MODEL. *Review of Business & Finance Studies*, *9*(1), 77–84. https://ssrn.com/abstract=3131958www.theIBFR.com

Lauff, C., Kotys-Schwartz, D., & Rentschler, M. (2018). What is a Prototype? What are the Roles of Prototypes in Companies? *Journal of Mechanical Design*, *140*. https://doi.org/10.1115/1.4039340

Li, Q., & Chen, Y.-L. (2009). Entity-Relationship Diagram. Dalam Q. Li & Y.-L. Chen (Ed.), *Modeling and Analysis of Enterprise and Information Systems: From Requirements to*

Realization (hlm. 125–139). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-89556-5_6

M.Ak, J., Ahir, R., Indrianti, E., & Hidayatullah, R. (2021). Pendampingan Penyusunan Laporan Keuangan Pada Usaha Rumah Kos Di Kelurahan Tangkerang Barat Kecamatan Marpoyan Damai Pekanbaru. *JURPIKAT (Jurnal Pengabdian Kepada Masyarakat)*, *2*, 53–59. https://doi.org/10.37339/jurpikat.v2i1.464

Monteverde, A. L., Maderazo, J. J. S., Cruz, K. C. M., & Magnaye, Ni. A. (2023). Web-Based Rental House Smart Finder using Rapid Application Development basis for Evaluation of ISO 205010. *International Journal of Metaverse*, *1*(1), 1–4. https://doi.org/10.54536/ijm.v1i1.1464

Mulya, R., S, A., Ilham, M., & Siregar, A. (2024). JURNAL MANAJEMEN TEKNOLOGI INFORMATIKA WEB BASED APPLICATION FOR BOARDING HOUSE SERVICE IN EKASAKTI UNIVERSITY AREA. *Jl. Veteran No.26B*, 25115.

Mutmainnah Ismail, N. I., Andayani, D. D., & Rahman, E. S. (2022). Pengembangan Aplikasi Pembimbingan Akademik Berbasis Web pada Jurusan Teknik Informatika dan Komputer Prodi Pendidikan Teknik Informatika dan Komputer UNM. *Journal of Embedded Systems, Security and Intelligent Systems*, *3*(1), 50. https://doi.org/10.26858/jessi.v3i1.28014

Ompusunggu, V. M., Ginting Munthe, R., Purba, M., & Sembiring, W. A. (2022). Analisis Faktor Yang Memengaruhi Keputusan Dalam Memilih Tempat Indekos Pada Mahasiswa Universitas Quality. *Mahkota Bisnis (Makbis)*, *1*(2), 2022. http://www.academia.edu

Pulungan, S. M., Febrianti, R., Lestari, T., Gurning, N., & Fitriana, N. (2023). *Analisis Teknik Entity-Relationship Diagram Dalam Perancangan Database*. *01*(2), 143–147. https://doi.org/10.47233/jemb.v2i1.533

Qayyum, A. (2012, Oktober 23). *The Challenges of Designing and Development*. https://smashinghub.com/the-challenges-of-designing-and-development.htm.

Rahayu, N., Agus Supriyono, I., & Mulyawan, E. (2022). Pembangunan Ekonomi Indonesia Dengan Tantangan Transformasi Digital. *ADI Bisnis Digital Interdisiplin Jurnal*, *4*(1), 92–95. https://doi.org/10.34306/abdi.v4i1.823

Ramanda, G., Mulyawan, B., Novario,), & Perdana, J. (2025). *Jurnal Ilmu Komputer dan Sistem Informasi PERANCANGAN APLIKASI PENGELOLAAN KOS BERBASIS WEB MENGGUNAKAN METODE USER CENTERED DESIGN (UCD)*.

Reddy, R. V., Yde, J. V., Barros, F. B. A., Antunes, C., & Wagner, A. (2024). Large scale dynamic web deployment for CERN, experience and outlook. *EPJ Web of Conferences*, *295*. https://doi.org/10.1051/epjconf/202429505011

Rosa, A., & Shalahuddin. (2013). *Rekayasa perangkat lunak terstruktur dan berorientasi objek*.

Saidu, A., Kwadan, S. M., & Adamu, S. H. (2015). *Issues and Challenge towards Enhancement of Web Application Development*. *5*(8). www.iiste.org

Setiyawati, E., & Sardiarinto. (2016). PERANCANGAN SISTEM INFORMASI BERBASIS WEB Studi Kasus: KSU BMT Al-Ikhwan Yogyakarta. Dalam *Indonesian Journal on Computer and Information Technology* (Vol. 1).

Suheri, A., Sulaeman, H., & Widaningsih, S. (2024). Web E-Commerce Kamar Indekost dengan PERN Stack. *Academic Journal of Computer Science Research*, *6*(2), 74. https://doi.org/10.38101/ajcsr.v6i2.10811

Sumiko, S. R. (2022). Sistem Informasi Pembayaran dan Pendataan Kost D'house Petukangan Jakarta Selatan. *Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI)*, *3*(03). https://doi.org/10.30998/jrami.v3i03.4054

Taufik, A., & Ermawati. (2017). Perancangan Sistem Informasi Pemesanan Pentas Seni Berbasis Web Pada Sanggar Seni Getar Pakuan Bogor. *ijse.bsi.ac.id IJSE-Indonesian Journal on Software Engineering*, *3*(2).

Triansah, A., Cahyadi, D., & Astuti, I. F. (2016). Membangun Aplikasi Web Dan Mobile Android Untuk Media Pencarian Kost Menggunakan Phonegap Dan Google Maps API. *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, *10*(1), 58. https://doi.org/10.30872/jim.v10i1.21

Wahyudi, I. R. (2023). Website-Based Boarding Rental Information System Design. *International Journal of Research and Applied Technology*, *3*(1), 31–40. https://doi.org/10.34010/injuratech.v3i1.9871

Widodo, W., Septita Dahlia, A., Ardiansyah, D., Teknik Informatika, P., El Rahma Yogyakarta, S., & Sistem Informasi, P. (2019). PENGARUH KEMUDAHAN PENGGUNAAN, KUALITAS INFORMASI, KUALITAS INTERAKSI, TERHADAP KEPUASAN PENGGUNA APLIKASI OPEN SOURCE MENGGUNAKAN WEBQUAL. *Jurnal Informatika dan Komputer) Akreditasi KEMENRISTEKDIKTI*, *3*(36). https://doi.org/10.33387/jiko

Widya, D., Pamungkas, L., & Rochimah, S. (2019). JURNAL IPTEK MEDIA KOMUNIKASI TEKNOLOGI Pengujian Aplikasi Web-Tinjauan Pustaka Sistematis. *Jurnal IPTEK*, *23*(1). https://doi.org/10.31284/j.iptek.2019.v23i1

Yudha, F., Muhammad, A., & Muryadi, P. (2018). *CyberSecurity dan Forensik Digital PERANCANGAN APLIKASI PENGUJIAN CELAH KEAMANAN PADA APLIKASI BERBASIS WEB*. *1*(1), 1–6.

LAMPIRAN