



SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**Pengembangan Sistem Penagihan Iuran Rumah Kost
Menggunakan *Whatsapp API* Studi Kasus Kost Barokah**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer

**MUHAMAD RIZKI
0110220098**

**PROGRAM STUDI TEKNIK INFORMATIKA
DEPOK
AGUSTUS 2024**



**STT TERPADU
NURUL FIKRI**

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

**Pengembangan Sistem Penagihan Iuran Rumah Kost
Menggunakan *Whatsapp API* Studi Kasus Kost Barokah**

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer

STT - NF
MUHAMAD RIZKI
0110220098

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024

HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.



STT - NF

Muhamad Rizki

HALAMAN PENGESAHAN

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Muhamad Rizki

NIM : 0110220098

Program Studi : Teknik Informatika

Judul Skripsi : Pengembangan Sistem Penagihan Iuran Rumah Kost
Menggunakan *Whatsapp API* Studi Kasus Kost Barokah

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing



(Henry Saptono, S.Si., M.Kom.)

Penguji



(Nasrul, S.Pd.I., S.Kom., M.Kom.)

Ditetapkan di : Depok

Tanggal : 12 Agustus 2024

STT - NF

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat *Allah Subhanahu wa ta ala*, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana komputer Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi/tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. *Allah Subhanahu wa ta ala*.
2. Kedua orang tua penulis, Ayahanda Muhamad Saim dan Ibunda Titin Barokah yang telah menjadi orang tua terhebat. Terima kasih yang tidak terhingga atas limpahan kasih sayang dan do'a yang tidak pernah putus, baik motivasi, nasehat, materi, dan pengorbanan yang diberikan. Serta kakak-kakak saya Fahrudin Baydawi dan Magna Syafrina yang selalu memberikan nasehat dan motivasi. Dan semua anggota keluarga yang telah memberikan dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
3. Bapak Dr. Lukman Rosyidi selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Ibu Tiffany Nabarian, S.Kom, M.TI. selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
5. Bapak Dr. Lukman Rosyidi, ST., MM., MT. selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama perkuliahan di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak Henry Saptono, S.Si., M.Kom. selaku Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
7. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.

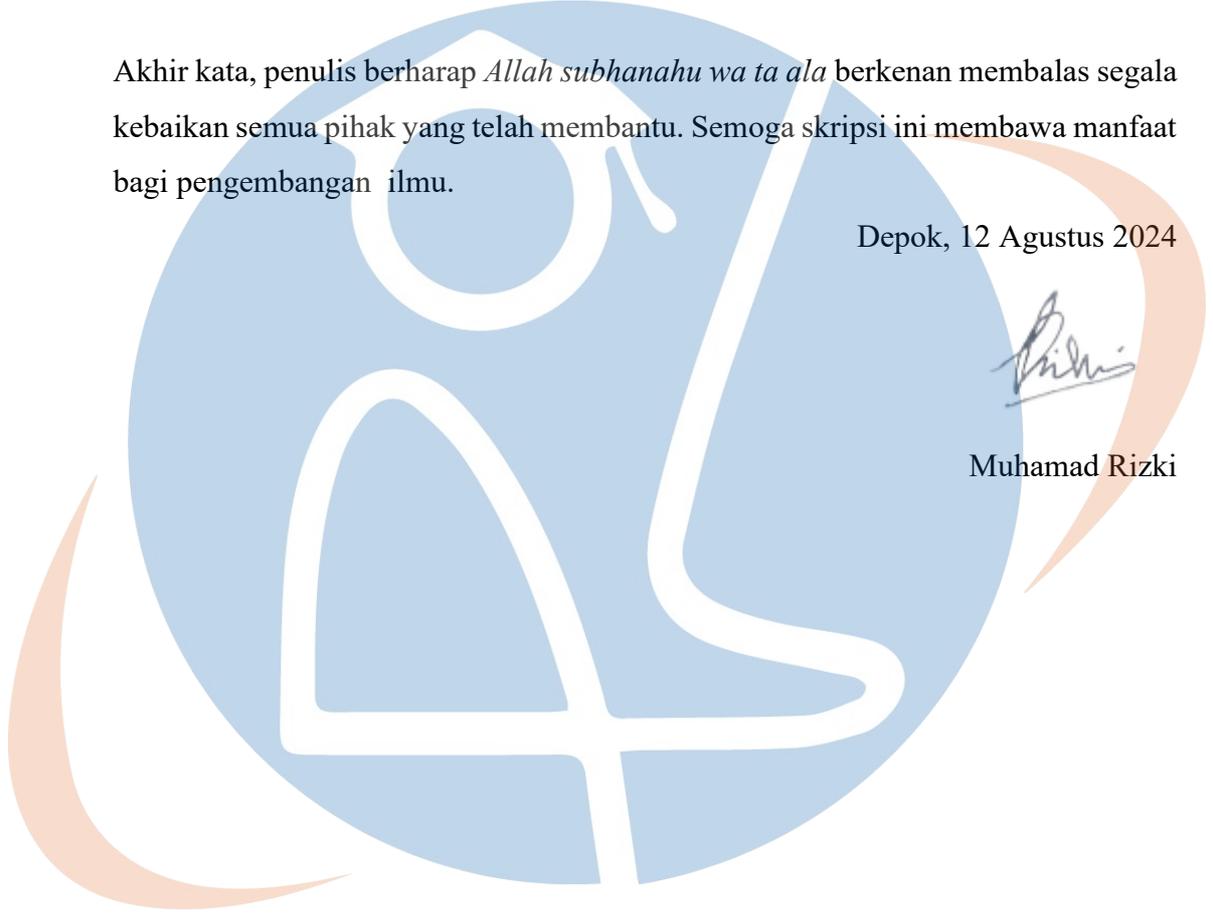
Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila terdapat kekurangan di dalam penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap *Allah subhanahu wa ta ala* berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 12 Agustus 2024



Muhamad Rizki



STT - NF

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Muhamad Rizki

NIM : 0110220098

Program Studi : Teknik Informatika

Jenis karya : Skripsi / Tugas Akhir

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty - Free Right)** atas karya ilmiah saya yang berjudul : Pengembangan Sistem Penagihan Iuran Rumah Kost Menggunakan *Whatsapp API* Studi Kasus Kost Barokah.

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 12 Agustus 2024

Yang Menyatakan



(Muhamad Rizki)

STT - NF

ABSTRAK

Nama : Muhamad Rizki
NIM : 0110220098
Program Studi : Teknik Informatika
Judul : Pengembangan Sistem Penagihan Iuran Rumah Kost Menggunakan *Whatsapp API* Studi Kasus Kost Barokah

Saat ini dengan perkembangan teknologi yang pesat dalam bidang komunikasi dan sistem informasi, kebutuhan akan informasi yang mudah diakses dan cepat semakin meningkat. Salah satu dampaknya adalah perubahan cara manusia berkomunikasi dan berinteraksi. Penelitian bertujuan untuk merancang dan mengevaluasi dalam pembuatan sistem penagihan iuran rumah kost yang efektif. Metode penelitian yang digunakan adalah analisis kualitatif, yang berfokus pada pemahaman yang mendalam seperti kebutuhan pengguna, perilaku, dan motivasi. Penelitian ini menghasilkan pengembangan dari sistem penagihan iuran rumah kost dalam memanfaatkan *API whatsapp* dan *framework laravel*. Sistem ini dirancang menggunakan pemodelan *Unified Modeling Language (UML)* seperti *usecase diagram*, *activity diagram*, dan *class diagram*. Serta menggunakan layanan pihak ketiga untuk mengubungkan *API whatsapp* dan *website* rumah kost. Dalam penelitian ini menggunakan metode pengujian *blackbox testing* dan *usability testing* agar sistem yang dirancang dapat digunakan dan memenuhi harapan pengguna.

STT - NF

Kata kunci : *Blackbox Testing*, *Framework Laravel*, Rumah Kost, *Unified Modeling Language*, *Usability Testing*, *Whatsapp API*.

ABSTRACT

Name : Muhamad Rizki
NIM : 0110220098
Study Program : Informatics
Title : Development of a Boarding House Payment Collection System Using WhatsApp API: A Case Study of Kost Barokah

With the rapid advancement of technology in the fields of communication and information systems, the demand for easily accessible and fast information is increasing. One of the impacts of this is a change in the way humans communicate and interact. This research aims to design and evaluate the creation of an effective boarding house billing system. The research method used is qualitative analysis, which focuses on a deep understanding of user needs, behavior, and motivation. This research resulted in the development of a boarding house fee billing system utilizing the whatsapp API and the Laravel framework. The system is designed using Unified Modeling Language (UML) modeling such as use case diagrams, activity diagrams, and class diagrams. It also uses third-party services to integrate the whatsapp API and the boarding house website. In this research, blackbox testing and usability testing methods are used to ensure that the designed system can be used and meets user expectations.

Key words : Blackbox Testing, Boarding House, Framework Laravel, Unified Modeling Language, Usability Testing, WhatsApp API.

DAFTAR ISI

| | |
|---|------|
| HALAMAN PERNYATAAN ORISINALITAS..... | iii |
| HALAMAN PENGESAHAN..... | iv |
| KATA PENGANTAR | v |
| HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI..... | vii |
| TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS | vii |
| ABSTRAK..... | viii |
| <i>ABSTRACT</i> | ix |
| DAFTAR ISI..... | x |
| DAFTAR GAMBAR | xiii |
| DAFTAR TABEL..... | xvi |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Tujuan dan Manfaat Penelitian..... | 2 |
| 1.4 Batasan Masalah..... | 3 |
| 1.5 Sistematika Penulisan..... | 3 |
| BAB II KAJIAN LITERATUR | 5 |
| 2.1. Kajian Teori..... | 5 |
| 2.1.1. Sistem..... | 5 |
| 2.1.2. Pengembangan Sistem | 5 |
| 2.1.3. Rumah Kost..... | 5 |
| 2.1.4. <i>PHP</i> | 5 |
| 2.1.5. <i>Laravel</i> | 6 |
| 2.1.6. <i>MySQL</i> | 6 |

| | | |
|---|--|-----------|
| 2.1.7. | <i>Instant Messaging</i> | 6 |
| 2.1.8. | <i>API (Application Programming Interface)</i> | 7 |
| 2.1.9. | <i>Whatsapp API</i> | 7 |
| 2.1.10. | <i>System Development Life Cycle</i> | 8 |
| 2.1.11. | <i>Waterfall</i> | 8 |
| 2.1.12. | <i>Watsap.id</i> | 9 |
| 2.2. | <i>Profile Tempat Studi Kasus</i> | 9 |
| 2.3. | <i>Penelitian Terkait</i> | 10 |
| BAB III METODOLOGI PENELITIAN | | 13 |
| 3.1. | <i>Tahapan Penelitian</i> | 13 |
| 3.2. | <i>Rancangan Penelitian</i> | 14 |
| 3.2.1. | <i>Jenis Penelitian</i> | 14 |
| 3.2.2. | <i>Metode Analisis</i> | 14 |
| 3.2.3. | <i>Metode Pengumpulan Data</i> | 15 |
| 3.2.4. | <i>Lingkungan Pengembangan</i> | 16 |
| 3.2.5. | <i>Metode Pengujian</i> | 16 |
| 3.2.6. | <i>Metode Implementasi dan Evaluasi</i> | 17 |
| BAB IV IMPLEMENTASI DAN EVALUASI | | 18 |
| 4.1. | <i>Analisis dan Perancangan</i> | 18 |
| 4.1.1. | <i>Analisis sistem</i> | 18 |
| 4.1.2. | <i>Kebutuhan Perangkat Keras</i> | 18 |
| 4.1.3. | <i>Kebutuhan Perangkat Lunak</i> | 19 |
| 4.2. | <i>Perancangan Sistem</i> | 19 |
| 4.1.1. | <i>Perancangan Arsitektur Sistem Logik</i> | 19 |
| 4.1.2. | <i>Perancangan Arsitektur Sistem Fisik</i> | 20 |
| 4.1.3. | <i>Perancangan UML (Unified Modeling Language)</i> | 21 |

| | | |
|---------------------------------|---|-----|
| 4.1.4. | Perancangan User Interface..... | 28 |
| 4.1.5. | Perancangan Menu | 32 |
| 4.1.6. | Perancangan Pengujian | 33 |
| 4.3. | Implementasi Sistem | 52 |
| 4.3.1. | Memasang dan Menjalankan <i>Framework Laravel</i> | 52 |
| 4.3.2. | Membuat <i>Authentication Laravel</i> | 54 |
| 4.3.3. | Melakukan Pembuatan <i>Menu Jadwal</i> | 55 |
| 4.3.4. | Membuat <i>Menu Laporan Pengiriman Pesan dan Pembayaran</i> | 58 |
| 4.3.5. | Membuat <i>Routing</i> | 62 |
| 4.3.6. | Membuat <i>AdminMiddleware</i> | 63 |
| 4.3.7. | Mendapatkan <i>Key API Whatsapp</i> | 64 |
| 4.3.8. | Memasang <i>API Key Pada Laravel</i> | 67 |
| 4.3.9. | Membuat <i>Traits Laravel</i> | 67 |
| 4.3.10. | Membuat <i>Task Pengiriman Pesan</i> | 69 |
| 4.3.11. | Menjalankan <i>Task Secara Otomatis</i> | 71 |
| 4.4. | Evaluasi Sistem | 71 |
| 4.4.1. | Tampilan Antarmuka Aplikasi..... | 72 |
| 4.4.2. | <i>Blackbox Testing</i> | 80 |
| 4.4.3. | Menghitung Hasil Pengujian <i>Blackbox Testing</i> | 101 |
| 4.4.4. | <i>Usability Testing</i> | 103 |
| 4.4.5. | Menghitung Hasil Pengujian <i>Usability Testing</i> | 115 |
| BAB V KESIMPULAN DAN SARAN..... | | 118 |
| 5.1. | Kesimpulan..... | 118 |
| 5.2. | Saran..... | 118 |
| DAFTAR REFERENSI | | 119 |
| LAMPIRAN..... | | 122 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2. 1 Arsitektur <i>MVC</i> | 6 |
| Gambar 2. 2 <i>Whatsapp API architecture</i> | 7 |
| Gambar 2. 3 Metode <i>Waterfall</i> | 9 |
| Gambar 2. 4 Pencatatan Iuran Bulanan | 10 |
| Gambar 3. 1 Tahapan Penelitian | 13 |
| Gambar 4. 1 Rancangan Arsitektur Sistem Logik | 20 |
| Gambar 4. 2 Rancangan Arsitektur Sistem Fisik | 21 |
| Gambar 4. 3 <i>Class Diagram</i> - Rumah Kost | 22 |
| Gambar 4. 4 <i>Usecase Diagram</i> Sistem Penagihan Iuran Rumah Kost | 23 |
| Gambar 4. 5 <i>Activity Diagram</i> - <i>User Login</i> | 24 |
| Gambar 4. 6 <i>Activity Diagram</i> - <i>Edit Profile</i> | 25 |
| Gambar 4. 7 <i>Activity Diagram</i> - <i>User Kelola Jadwal Tagihan</i> | 26 |
| Gambar 4. 8 <i>Activity Diagram</i> - <i>User Kelola Perangkat</i> | 27 |
| Gambar 4. 9 <i>User Interface</i> – <i>Login</i> | 28 |
| Gambar 4. 10 <i>User Interface</i> – <i>Dashboard</i> | 29 |
| Gambar 4. 11 <i>User Interface</i> - <i>Daftar Data</i> | 29 |
| Gambar 4. 12 <i>User Interface</i> - <i>Formulir Data</i> | 30 |
| Gambar 4. 13 <i>User Interface</i> - <i>Edit Profile</i> | 30 |
| Gambar 4. 14 <i>User Interface</i> – <i>Perangkat</i> | 31 |
| Gambar 4. 15 <i>User Interface</i> - <i>Jadwal Pengiriman Pesan</i> | 31 |
| Gambar 4. 16 Tampilan Halaman - <i>Login Watsap.id</i> | 65 |
| Gambar 4. 17 Tampilan Halaman - <i>Register Watsap.id</i> | 65 |
| Gambar 4. 18 Tampilan Halaman - <i>Dashboard Watsap.id</i> | 66 |
| Gambar 4. 19 Tampilan Halaman - <i>Device Watsap.id</i> | 66 |
| Gambar 4. 20 Tampilan Halaman - <i>Tambah Device Watsap.id</i> | 67 |
| Gambar 4. 21 Implementasi <i>User Interface</i> - <i>Halaman Login</i> | 72 |
| Gambar 4. 22 Implementasi <i>User Interface</i> - <i>Halaman Dashboard</i> | 73 |
| Gambar 4. 23 Implementasi <i>User Interface</i> - <i>Halaman Edit Profile</i> | 73 |
| Gambar 4. 24 Implementasi <i>User Interface</i> - <i>Halaman Daftar User</i> | 74 |

| | |
|---|-----|
| Gambar 4. 25 Implementasi <i>User Interface</i> - Halaman Tambah <i>User</i> | 74 |
| Gambar 4. 26 Implementasi <i>User Interface</i> - Halaman Jadwal | 75 |
| Gambar 4. 27 Implementasi <i>User Interface</i> - Halaman Perangkat | 75 |
| Gambar 4. 28 Implementasi <i>User Interface</i> - Halaman Daftar Laporan..... | 76 |
| Gambar 4. 29 Implementasi <i>User Interface</i> - Halaman Tambah Laporan..... | 76 |
| Gambar 4. 30 Implementasi <i>User Interface</i> - Halaman Edit Laporan | 77 |
| Gambar 4. 31 Implementasi <i>User Interface</i> - Halaman Daftar Kost | 77 |
| Gambar 4. 32 Implementasi <i>User Interface</i> - Halaman Tambah Data Kost..... | 78 |
| Gambar 4. 33 Implementasi <i>User Interface</i> - Halaman Mengubah Data Kost..... | 78 |
| Gambar 4. 34 Implementasi <i>User Interface</i> - Halaman Daftar Kontak | 79 |
| Gambar 4. 35 Implementasi <i>User Interface</i> - Halaman Tambah Kontak | 79 |
| Gambar 4. 36 Implementasi <i>User Interface</i> - Halaman Merubah Kontak..... | 80 |
| Gambar 4. 37 Hasil Pengujian <i>Blackbox</i> - Halaman <i>Login</i> | 82 |
| Gambar 4. 38 Hasil Pengujian <i>Blackbox</i> – Melihat <i>Dashboard</i> | 84 |
| Gambar 4. 39 Hasil Pengujian <i>Blackbox</i> - Mengelola Data Kost..... | 85 |
| Gambar 4. 40 Hasil Pengujian <i>Blackbox</i> - Mengelola Data Kontak..... | 87 |
| Gambar 4. 41 Hasil Pengujian <i>Blackbox</i> - Mengelola Laporan Pembayaran | 88 |
| Gambar 4. 42 Hasil Pengujian <i>Blackbox</i> - Mengelola Perangkat | 90 |
| Gambar 4. 43 Hasil Pengujian <i>Blackbox</i> - Mengelola <i>User</i> | 91 |
| Gambar 4. 44 Hasil Pengujian <i>Blackbox</i> - Mengedit <i>Profile</i> | 92 |
| Gambar 4. 45 Hasil Pengujian <i>Blackbox</i> – Pengiriman Pesan - Waktu Tepat..... | 94 |
| Gambar 4. 46 Hasil Pengujian <i>Blackbox</i> - Pengiriman Pesan – Waktu Terlewat | 96 |
| Gambar 4. 47 Hasil Pengujian <i>Blackbox</i> – Pengiriman Pesan – Tanggal Terlewat | 98 |
| Gambar 4. 48 Hasil Pengujian <i>Blackbox</i> – Pengiriman Pesan – Jadwal Tidak Aktif | 99 |
| Gambar 4. 49 Hasil Pengujian <i>Blackbox</i> – Pengiriman Pesan – Jadwal Tidak Aktif | 101 |
| Gambar 4. 50 Hasil Pengujian <i>Usability Testing</i> - <i>Login</i> | 104 |
| Gambar 4. 51 Hasil Pengujian <i>Usability Testing</i> - Mengakses <i>Dashboard</i> | 105 |
| Gambar 4. 52 Hasil Pengujian <i>Usability Testing</i> - Mengelola Kost..... | 107 |

| | |
|--|-----|
| Gambar 4. 53 Hasil Pengujian <i>Usability Testing</i> - Mengelola Kontak..... | 108 |
| Gambar 4. 54 Hasil Pengujian <i>Usability Testing</i> - Mengelola Laporan | 110 |
| Gambar 4. 55 Hasil Pengujian <i>Usability Testing</i> - Mengelola Perangkat..... | 111 |
| Gambar 4. 56 Hasil Pengujian <i>Usability Testing</i> - Mengubah <i>Profile</i> | 113 |
| Gambar 4. 57 Hasil Pengujian <i>Usability Testing</i> - Mengelola <i>User</i> | 114 |
| Gambar 4. 58 Hasil Pengujian <i>Usability Testing</i> – Melihat Status Pengiriman Pesan | 115 |



STT - NF

DAFTAR TABEL

| | |
|--|----|
| Tabel 4. 1 Kebutuhan Perangkat Keras..... | 18 |
| Tabel 4. 2 Kebutuhan Perangkat Lunak..... | 19 |
| Tabel 4. 3 Kebutuhan Pengguna | 32 |
| Tabel 4. 4 <i>Menu Aplikasi</i> | 32 |
| Tabel 4. 5 Perancangan Pengujian <i>Blackbox - Login</i> | 34 |
| Tabel 4. 6 Perancangan Pengujian <i>Blackbox - Mengelola Data Kost</i> | 35 |
| Tabel 4. 7 Perancangan Pengujian <i>Blackbox - Mengelola Data Kontak</i> | 36 |
| Tabel 4. 8 Perancangan Pengujian <i>Blackbox - Mengelola Laporan Pembayaran</i> .. | 37 |
| Tabel 4. 9 Perancangan Pengujian <i>Blackbox - Pengaturan Perangkat</i> | 38 |
| Tabel 4. 10 Perancangan Pengujian <i>Blackbox - Mengelola User</i> | 39 |
| Tabel 4. 11 Perancangan Pengujian <i>Blackbox - Mengedit Profile</i> | 39 |
| Tabel 4. 12 Rancangan Pengujian <i>Blackbox – Melihat Dashboard</i> | 40 |
| Tabel 4. 13 Perancangan Pengujian <i>Blackbox – Pengiriman Pesan - Waktu Tepat</i> | 40 |
| Tabel 4. 14 Perancangan Pengujian <i>Blackbox - Pengiriman Pesan – Waktu</i> <i>Terlewati</i> | 41 |
| Tabel 4. 15 Perancangan Pengujian <i>Blackbox – Pengiriman Pesan – Tanggal</i> <i>Terlewati</i> | 42 |
| Tabel 4. 16 Perancangan Pengujian <i>Blackbox – Pengiriman Pesan – Jadwal Tidak</i> <i>Aktif</i> | 44 |
| Tabel 4. 17 Perancangan Pengujian <i>Blackbox - Pengiriman Pesan – Tanggal dan</i> <i>Waktu Salah</i> | 45 |
| Tabel 4. 18 Perancangan Pengujian <i>Usability Testing - Login</i> | 46 |
| Tabel 4. 19 Perancangan Pengujian <i>Usability Testing - Mengakses Dashboard</i> .. | 46 |
| Tabel 4. 20 Perancangan Pengujian <i>Usability Testing - Mengelola Kost</i> | 47 |
| Tabel 4. 21 Perancangan Pengujian <i>Usability Testing - Mengelola Kontak</i> | 48 |
| Tabel 4. 22 Perancangan Pengujian <i>Usability Testing - Mengelola Laporan</i> | 49 |
| Tabel 4. 23 Perancangan Pengujian <i>Usability Testing - Mengelola Perangkat</i> | 50 |
| Tabel 4. 24 Perancangan Pengujian <i>Usability Testing - Mengubah Profile</i> | 51 |
| Tabel 4. 25 Perancangan Pengujian <i>Usability Testing - Mengelola User</i> | 51 |

| | |
|--|-----|
| Tabel 4. 26 Perancangan Pengujian <i>Usability Testing</i> – Melihat Status Pengiriman Pesan | 52 |
| Tabel 4. 27 Hasil Pengujian <i>Blackbox</i> - <i>Login</i> | 81 |
| Tabel 4. 28 Hasil Pengujian <i>Blackbox</i> – Melihat Dashboard..... | 83 |
| Tabel 4. 29 Hasil Pengujian <i>Blackbox</i> - Mengelola Data Kost..... | 84 |
| Tabel 4. 30 Hasil Pengujian <i>Blackbox</i> - Mengelola Data Kontak..... | 86 |
| Tabel 4. 31 Hasil Pengujian <i>Blackbox</i> - Mengelola Laporan Pembayaran..... | 87 |
| Tabel 4. 32 Hasil Pengujian <i>Blackbox</i> - Mengelola Perangkat..... | 89 |
| Tabel 4. 33 Hasil Pengujian <i>Blackbox</i> - Mengelola <i>User</i> | 90 |
| Tabel 4. 34 Hasil Pengujian <i>Blackbox</i> - Mengedit <i>Profile</i> | 91 |
| Tabel 4. 35 Hasil Pengujian <i>Blackbox</i> – Pengiriman Pesan - Waktu Tepat | 93 |
| Tabel 4. 36 Hasil Pengujian <i>Blackbox</i> - Pengiriman Pesan – Waktu Terlewati ... | 94 |
| Tabel 4. 37 Hasil Pengujian <i>Blackbox</i> – Pengiriman Pesan – Tanggal Terlewati | 96 |
| Tabel 4. 38 Hasil Pengujian <i>Blackbox</i> – Pengiriman Pesan – Jadwal Tidak Aktif | 98 |
| Tabel 4. 39 Hasil Pengujian <i>Blackbox</i> - Pengiriman Pesan – Tanggal dan Waktu Salah..... | 100 |
| Tabel 4. 40 Hasil Pengujian <i>Usability Testing</i> - <i>Login</i> | 103 |
| Tabel 4. 41 Hasil Pengujian <i>Usability Testing</i> - Mengakses <i>Dashboard</i> | 105 |
| Tabel 4. 42 Hasil Pengujian <i>Usability Testing</i> - Mengelola Kost..... | 106 |
| Tabel 4. 43 Hasil Pengujian <i>Usability Testing</i> - Mengelola Kontak..... | 107 |
| Tabel 4. 44 Hasil Pengujian <i>Usability Testing</i> - Mengelola Laporan | 109 |
| Tabel 4. 45 Hasil Pengujian <i>Usability Testing</i> - Mengelola Perangkat | 110 |
| Tabel 4. 46 Hasil Pengujian <i>Usability Testing</i> - Mengubah <i>Profile</i> | 112 |
| Tabel 4. 47 Hasil Pengujian <i>Usability Testing</i> - Mengelola User..... | 113 |
| Tabel 4. 48 Hasil Pengujian <i>Usability Testing</i> – Melihat Status Pengiriman Pesan | 114 |

BAB I

PENDAHULUAN

1.1 Latar belakang

Rumah kost adalah tempat tinggal sementara bagi individu yang bekerja atau mahasiswa yang sedang menempuh pendidikan diluar kota asal mereka.[1] Dalam hal ini, harga dan fasilitas yang ditawarkan oleh pemilik kost sangat bervariasi, mulai dari yang sederhana, menengah, hingga yang mewah, tergantung pada kebutuh dan biaya setiap penghuni kost. Di Indonesia, khususnya terletak di daerah Kabupaten Bogor, Kecamatan Gunung Putri, terdapat rumah kost barokah, yang dimana itu adalah kost campuran yang menawarkan beragam fasilitas kepada penghuninya.

Dengan perkembangan teknologi yang pesat, terutama dalam bidang komunikasi dan sistem informasi, hal ini mendorong peningkatan kebutuhan akan informasi yang mudah diakses, cepat, dan akurat. Salah satu contoh dampak dari perkembangan teknologi ini yaitu mengubah bagaimana cara manusia berkomunikasi dan berinteraksi. Salah satu platform komunikasi populer yaitu *whatsapp* menjadi platform komunikasi yang sangat digemari, dengan penggunaan yang meluas di hampir 180 negara.[2] Fitur-fitur yang ada di *whatsapp* menjadi faktor utama dalam peningkatan popularitasnya karena meningkatkan efisiensi, mengurangi penggunaan data, dan meningkatkan kinerja pekerjaan.[2] Penggunaan aplikasi ini untuk berkomunikasi sangat efektif dalam hal biaya dan waktu, baik itu dalam dunia bisnis, organisasi maupun institusi. [2]

Didalam bisnis proses dari rumah kost barokah, masih ada proses penagihan iuran pembayaran kost yang dilakukan secara manual. Pemilik kost harus mencatat pembayaran secara manual, dan memberikan bukti secara manual. Karena proses inilah sering terjadinya kesalahan manusia dan memakan waktu.

Dengan adanya *whatsapp api* yang terintegrasi dengan *website*, pemilik rumah kost barokah dapat mengurangi kesalahan dan menghemat waktu dalam proses penagihan, pembayaran, dan mengelola informasi

pembayaran. Pemanfaat api whatsapp memungkinkan pemilik tidak perlu melakukan hal tersebut, cukup melalui website yang sudah terintegrasi whatsapp api maka pemilik kost tidak perlu melakukan pengihan *door to door* atau satu-satu kepada penyewa. Melalui *website* yang sudah dihubungkan dengan *api* milik *whatsapp*, pemilik kost dapat dengan mudah mengirimkan pesan tagihan kepada semua penghuni kost secara bersamaan.

Memanfaatkan teknologi seperti *whatsapp* dapat membantu rumah kost barokah dalam meningkatkan efisiensi operasional dan mengurangi kesalahan manusia. Dengan demikian, *whastapp* menjadi alat yang memberikan dampak positif bagi keberlangsungan operasional dan kepuasan pengelola bisnis.

1.2 Rumusan Masalah

1. Bagaimana rancangan dari sistem penagihan iuran rumah kost menggunakan *whatsapp API*?
2. Bagaimana efektivitas dari sistem penagihan iuran rumah kost menggunakan *whatsapp API*?

1.3 Tujuan dan Manfaat Penelitian

Berikut ini adalah beberapa poin dari tujuan penelitian ini:

1. Membuat rancangan sistem penagihan iuran rumah kost
2. Mengetahui efektivitas sistem penagihan iuran rumah kost

Berikut ini adalah beberapa poin dari manfaat penelitian ini:

1. Memberikan kemudahan bagi pemilik kost dalam menagih iuran bulanan kos-kosan
2. Mengurangi kesalahan pencatatan dalam penagihan iuran rumah kost bagi pemilik rumah kost
3. Memberikan kontribusi terhadap pengembangan teori dalam menggunakan *framework laravel*

1.4 Batasan Masalah

1. Dalam penelitian ini, aplikasi yang dihasilkan dapat dioperasikan melalui *website*.
2. Dalam penelitian ini, proses pengujian aplikasi peneliti hanya melakukan penagihan iuran kost melalui *website*
3. Dalam aplikasi yang dirancang membahas tentang pengelolaan jadwal pengiriman penagihan iuran kost.
4. Aplikasi yang dirancang menggunakan *framework laravel* yang hanya dapat diakses oleh admin dan pemilik kost.
5. Hasil dari aplikasi ini adalah berhasil melakukan penagihan iuran kost melalui *website*.

1.5 Sistematika Penulisan

untuk memberikan gambaran yang lebih terperinci dan terstruktur, tugas akhir ini dibagi dalam beberapa bab dengan susunan sebagai berikut.

BAB I PENDAHULUAN

Pada bagian bab I merupakan bagian awal yang memberikan pengantar terhadap penelitian yang dilakukan, di dalam bab ini mencakup pembahasan tentang latar belakang, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB II KAJIAN LITERATUR

Pada bagian II akan dibahas definisi dan teori terkait pembuatan website. Dengan mengulas elemen-elemen penting dan mengkaji penelitian terkait, bab ini memberikan landasan yang kuat untuk memahami konsep-konsep yang relevan dengan pengembangan aplikasi manajemen properti kost-kostan berbasis website.

BAB III METODOLOGI PENELITIAN

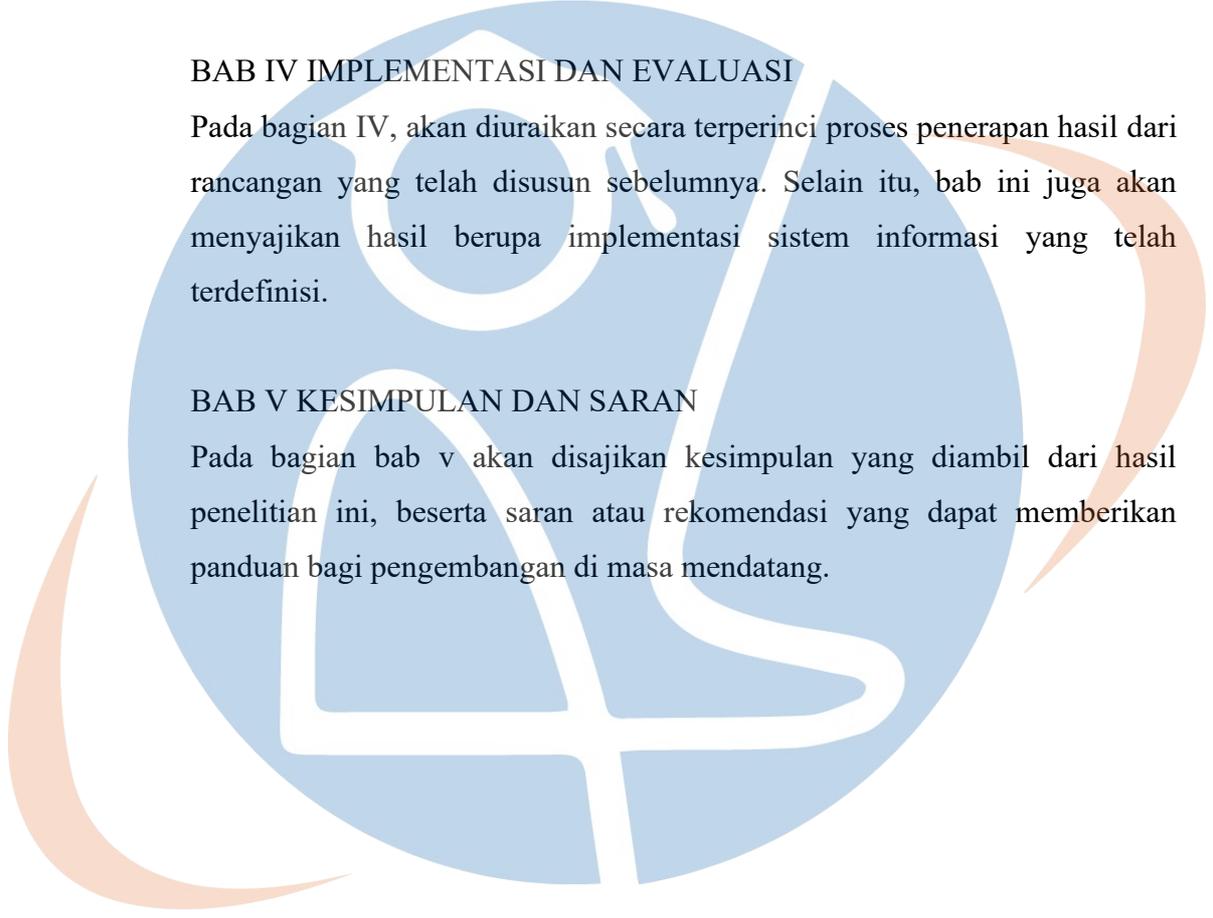
Pada bagian III akan menggambarkan secara detail yang akan dilakukan dalam penelitian ini. Langkah-langkah tersebut mencakup studi literatur, analisis, perancangan sistem, implementasi, testing, dan evaluasi. Selanjutnya, bab ini membahas rancangan penelitian, rencana waktu penelitian dan metode pengujian.

BAB IV IMPLEMENTASI DAN EVALUASI

Pada bagian IV, akan diuraikan secara terperinci proses penerapan hasil dari rancangan yang telah disusun sebelumnya. Selain itu, bab ini juga akan menyajikan hasil berupa implementasi sistem informasi yang telah terdefinisi.

BAB V KESIMPULAN DAN SARAN

Pada bagian bab v akan disajikan kesimpulan yang diambil dari hasil penelitian ini, beserta saran atau rekomendasi yang dapat memberikan panduan bagi pengembangan di masa mendatang.



STT - NF

BAB II

KAJIAN LITERATUR

Dalam bab ini akan diuraikan mengenai proses penelitian, hubungannya dengan riset terdahulu, serta mengumpulkan teori atau definisi terkait dengan penelitian yang akan dilakukan.

2.1. Kajian Teori

2.1.1. Sistem

Sistem adalah serangkaian komponen yang terhubung dan berinteraksi satu sama lain untuk mencapai tujuan tertentu. Sistem ini biasanya terbagi menjadi subsistem yang lebih kecil yang berperan dalam mendukung fungsi sistem yang lebih besar.[3]

2.1.2. Pengembangan Sistem

Pengembangan sistem adalah perubahan dari suatu sistem yang sudah ada diolah atau dimodifikasi menjadi versi yang lebih baik dan lebih bermanfaat. Perubahan ini dilakukan ketika sistem lama mengalami masalah, ketika ada peluang baru yang dapat dimanfaatkan, atau ketika ada intruksi dari pimpinan untuk melakukan perubahan.[4]

2.1.3. Rumah Kost

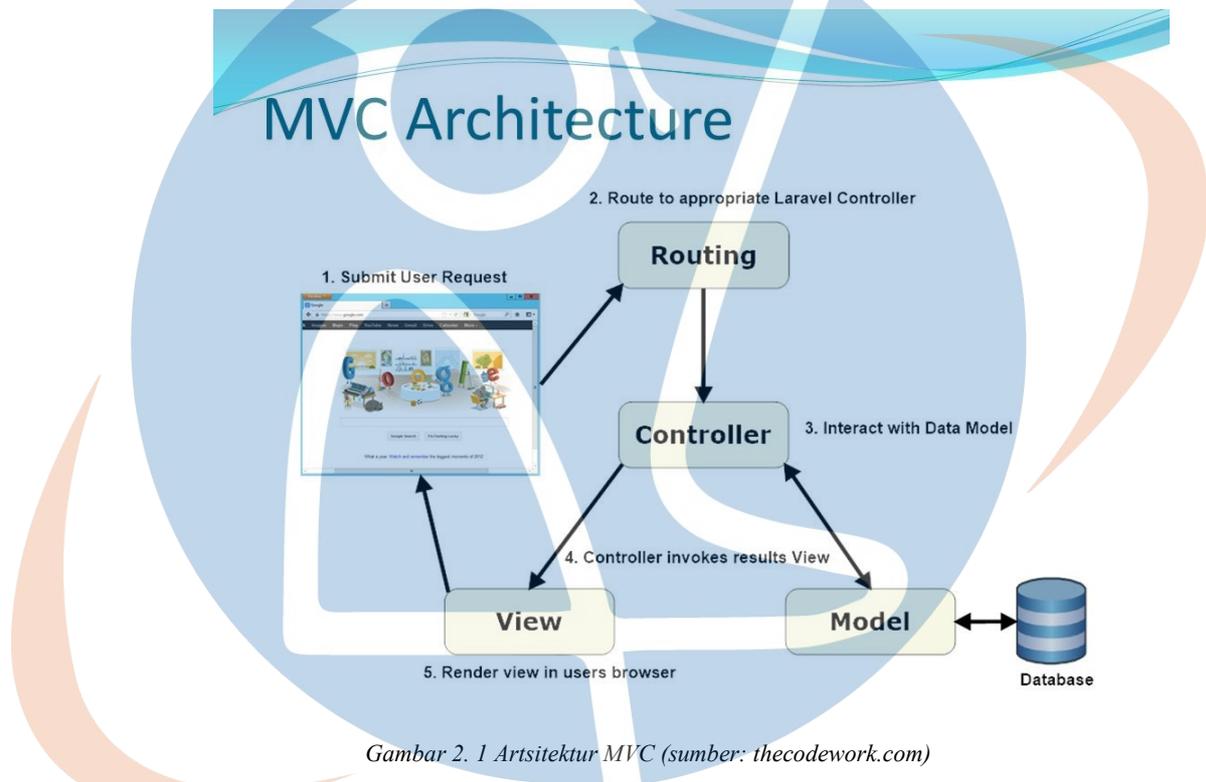
Rumah kost adalah tempat tinggal sementara bagi individu yang bekerja atau mahasiswa yang sedang menempuh pendidikan diluar kota asal.[1] Dalam hal ini, harga dan fasilitas yang ditawarkan oleh pemilik kost sangat bervariasi, mulai dari yang sederhana, menengah, hingga yang mewah, tergantung pada kebutuh dan biaya setiap penghuni kost.

2.1.4. PHP

PHP (Perl Hypertext Protocol) adalah sebuah bahasa pemrograman scripting yang berjalan di server dan diproses oleh server. Selain itu, PHP juga merupakan salah satu dari banyak bahasa pemrograman *HTML (Hypertext Markup Language)*. Bahasa awal ini awalnya diciptakan oleh Rasmus Lerdorf diawali sebagai proyek pribadi dan kemudian diperbaiki dan disempurnakan oleh *Group Six of Developer* sebelum kemudian dikenal kembali dengan nama *PHP*.[5]

2.1.5. *Laravel*

Laravel adalah kerangka kerja *web* berbasis bahasa pemrograman *PHP* yang memiliki sumber terbuka secara gratis, kerangka kerja ini diciptakan oleh Taylor Otwell dan dirancang khusus untuk memudahkan pengembangan aplikasi *web* dengan menggunakan pola *MVC (Model-View-Controller)*. Salah satu fitur utama *laravel* adalah sistem *routing* yang menghubungkan permintaan (*request*) dari pengguna (*user*) dengan kontroler (*controller*). Menjadikan *controller* tidak langsung menerima *request* tersebut secara langsung. [6]



Gambar 2. 1 Arsitektur MVC (sumber: thecodework.com)

2.1.6. *MySQL*

MySQL adalah sebuah sistem manajemen basis data relasional atau *Relational Database Management System (RDBMS)* yang tersedia secara gratis dengan lisensi *GPL (General Public License)*. Meskipun dapat digunakan secara bebas, namun ada penggunaan perangkat lunak ini tidak boleh digunakan untuk tujuan komersial yang menghasilkan produk turunan.[7]

2.1.7. *Instant Messaging*

Instant Messaging adalah sebuah layanan komunikasi online yang memfasilitasi pengguna internet untuk berinteraksi dan berkomunikasi dengan

sesama pengguna. Selain itu, layanan ini juga memungkinkan pengguna untuk berbagi file secara langsung antara satu sama lain.[8] Contoh dari layanan *instant messaging* yang terkenal saat ini adalah *whatsapp*.

2.1.8. API (Application Programming Interface)

Application Programming Interface adalah sebuah antarmuka yang dibuat oleh pengembang supaya aplikasi lain dapat mengakses dan menggunakan fungsi-fungsi sistem secara terprogram.[9]

2.1.9. Whatsapp API

WhatsApp Messenger adalah aplikasi pesan yang dapat diinstall di *smartphone*, dan diakses menggunakan *desktop*. Selain itu, *WhatsApp Messenger* juga dikenal sebagai aplikasi pesan lintas platform yang memungkinkan pengguna untuk bertukar pesan tanpa menggunakan pulsa karena mengandalkan paket data internet. Saat ini, *WhatsApp* dilengkapi dengan fitur melakukan panggilan, membuat grup, pengiriman pesan, dan pesan suara.[10]



Gambar 2. 2 Whatsapp API architecture (sumber: get.chat)

Salah satu fitur unggulan dari *WhatsApp* adalah adanya *API WhatsApp* yang memungkinkan berintegrasi dengan sistem perangkat lunak lainnya. Dengan memanfaatkan *API WhatsApp*, sistem yang telah terintegrasi dapat langsung terhubung dengan aplikasi pesan dari platform aplikasi *smartphone* maupun *web*. [10]

2.1.10. System Development Life Cycle

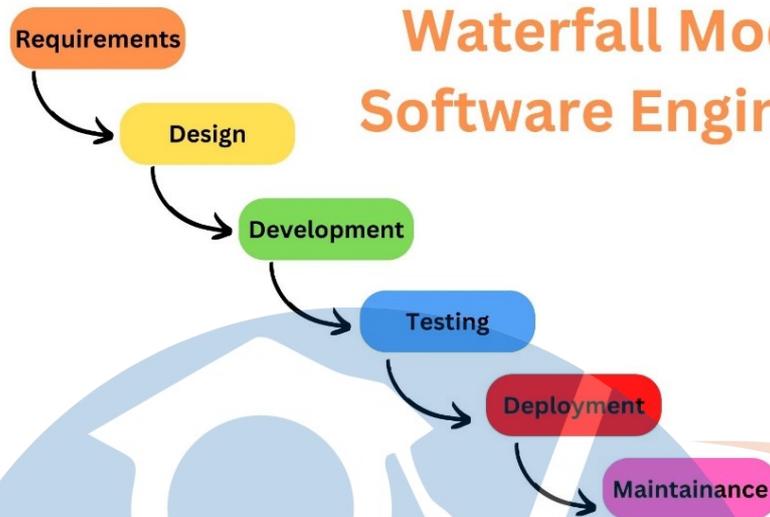
Sirkulus hidup pengembangan perangkat lunak (SDLC) dalam rekayasa perangkat lunak dan rekayasa sistem merupakan proses yang digunakan oleh tim pengembangan untuk membuat dan mengubah perangkat lunak, sekaligus menghemat biaya dan waktu. Tujuan dari SDLC adalah untuk mengurangi risiko dalam proyek dengan melakukan perencanaan yang matang, sehingga dapat memenuhi harapan pelanggan selama proses produksi dan dimasa-masa berikutnya.

SDLC juga merupakan metodologi yang umum digunakan dalam pengembangan sistem, yang menandai langkah maju dalam proses analisis dan desain. Tahapan-tahapan SDLC mencakup perencanaan sistem (*system planning*), analisis sistem (*system analysis*), perancangan sistem (*system design*), implementasi sistem (*system implementation*), pemeliharaan sistem (*system maintenance*). Konsep dari SDLC menjadi landasan untuk berbagai model pengembangan perangkat lunak, seperti model *waterfall*, prototipe, iteratif, *spiral*, *rapid application development* (RAD), dan sebagainya.[11] Dalam lingkup penelitian ini, hanya satu model yaitu *waterfall* yang akan dibahas.

2.1.11. Waterfall

Waterfall adalah metode pengembangan perangkat lunak yang umum digunakan dengan melibatkan serangkaian tahap linier mulai dari perencanaan, analisis, desain, implementasi, dan pengujian. Dimana setiap tahapan harus diselesaikan sebelum melanjutkan ke tahap berikutnya, dan tidak memungkinkan untuk mundur atau mengulang ke tahapan sebelumnya.[11]

Waterfall Model in Software Engineering



Gambar 2. 3 Metode Waterfall (sumber: www.boardinfinity.com)

2.1.12. Watsap.id

Whatsapp adalah alat aplikasi yang digunakan untuk mengirim pesan ke banyak nomor handphone yang sudah terdaftar di *whatsapp*. Didalam aplikasi watsap.id ini terdapat kode unik yang dapat digunakan untuk berintegrasi dengan layanan *whatsapp* yang nantinya akan digunakan untuk melakukan pengiriman pesan otomatis, notifikasi, dan interaksi dengan pengguna melalui pihak ketiga. Contoh penerapan dari pihak ketiga dapat ditemukan dalam gambar 2.3.

2.2. Profile Tempat Studi Kasus

Rumah kost barokah dibangun pada tahun 2009 Dan beralamat di Kampung Kedep RT 02 RW 22, Kecamatan Gunung Putri, Desa Tlajung Udik, Kabupaten Bogor, Provinsi Jawa Barat. Saat ini pencatatan pembayaran iuran kost bulanan masih dilakukan secara manual dan pengingat penagihan iuran kost masih menggunakan cara manual.

Nama : Mamah Amel .

| No: | Bulan | Tanggal Pembayaran | TTD | Keterangan |
|-----|-----------|--------------------|---------------------|------------|
| 1 | Jandari | 7-2-2024. | 350.000 + 100.000 | |
| 2 | Februari | 3-2-2024. | 350.000 + | |
| 3 | Maret | | | |
| 4 | April | | | |
| 5 | Mei | | | |
| 6 | Juni | | | |
| 7 | Juli | | | |
| 8 | Agustus | | | |
| 9 | September | 5-9-23. | 400.000 + 50.000. | |
| 10 | Oktober | | -400.000 + 500.000 | |
| 11 | Nopember | 5-12-23. | 350.000 + 100.000 | |
| 12 | Desember | - 12 - | 350.000 + 100.000 . | |

Gambar 2. 4 Pencatatan Iuran Bulanan

2.3. Penelitian Terkait

Setelah melakukan tinjauan literatur, terdapat beberapa penelitian terkait yang telah teridentifikasi dan akan digunakan sebagai acuan dalam pengembangan penelitian ini. Berikut adalah beberapa di antaranya:

Tabel 2. 1 Penelitian Terkait

| No | Nama dan Tahun | Judul | Topik | Subjek | Hasil |
|----|--|--|--|-------------------|--------------------|
| 1 | Yohanes Mada Masa, Emerensiana Ngaga, Yovinia Carmeneja Hoar Siki, Donatus Joseph Manehat, 2023 [12] | “Rancang bangun sistem informasi rumah kost di kota kupang berbasis web” | Rancangan sistem informasi rumah kost | Remaja dan Dewasa | Aplikasi Website |
| 2 | Aditia, Sultan, Najamuddin Dwi Miharja, Muhammad, Aguswin, Ahmad, 2023 [13] | “Implementasi sistem kehadiran praktikum berbasis qrcode dengan whatsapp gateway menggunakan metode rapid application development (RAD)” | Pembuatan sistem kehadiran menggunakan qrcode dan whatsapp gateway | Pelajar | Aplikasi kehadiran |

| No | Nama dan Tahun | Judul | Topik | Subjek | Hasil |
|----|---|---|--|---------------------------|-------------------------|
| 3 | Arimbi, Yuti Dewita, Kartinah, Dwi, Nila, Anggit, Della, Waska, 2022 [14] | “Rancangan sistem informasi kost putri malika berbasis <i>website</i> menggunakan <i>framework laravel</i> dan <i>mysql</i> ” | Perancangan webiste informasi kost | Pemilik dan Penghuni kost | Aplikasi <i>Website</i> |
| 4 | Mike, Amran, Hutauruk, Gilbert, Parga Zen, Bitu, Utami, Annisaa [15] | “Penerapan Metode <i>Agile</i> Pada <i>Website</i> Indekost Sruntul Menggunakan <i>Framework Laravel</i> ” | Perancangan sistem informasi aplikasi kost | Pemilik dan penghuni kost | Aplikasi <i>Website</i> |

1. Pada penelitian “Rancang bangun sistem informasi rumah kost di kota kupang berbasis *web*” bahwa Sistem Informasi rumah kost yang dirancang untuk rumah kost di kota kupang membuktikan dirinya dapat mengatasi masalah bagi pemilik dan calon penghuni kost. bagi pemilik rumah kost yaitu mengurangi waktu yang dibutuhkan dalam mendapatkan calon penghuni kost, dan bagi calon penghuni kost yaitu dapat menyesuaikan pencarian kepada calon pemilik kost dari fasilitas, harga, serta kenyamanan tempat kost.
2. Pada penelitian “Implementasi sistem kehadiran praktikum berbasis *qrcode* dengan *whatsapp gateway* menggunakan metode *rapid application development* (RAD)” menjelaskan tentang sistem kemudahan dalam pencatatan dan pemantauan secara *real-time*. Dengan menggunakan sistem ini, mahasiswa peserta pratikum tidak perlu lagi menyediakan kartu kendali yang harus disetujui dan ditandatangani oleh pengawas. Sebaliknya, pengawas dapat langsung menginput data kehadiran mahasiswa yang mengikuti pratikum ke dalam file arsip *microsoft excel* secara langsung melalui sistem yang telah disediakan.[13]
3. Pada penelitian “Rancangan sistem informasi kost putri malika berbasis *website* menggunakan *framework laravel* dan *mysql*” studi ini membahas tentang perancangan sistem informasi berbasis website untuk kost Putri Malika menggunakan *framework laravel* dan *MySQL*. *Website* ini dirancang untuk

mempermudah proses pendaftaran dan reservasi kost Putri Malika. Tujuan dari *website* ini adalah menggantikan proses pencatatan manual yang sebelumnya diterapkan untuk menjadikan pencatatannya dengan lebih akurat dan memudahkan dalam pendaftaran dan reservasi bagi calon penghuni kost Putri Malika. [14]

4. Pada penelitian “Penerapan Metode Agile Pada Website Indekost Sruntul Menggunakan Framework Laravel” pendataan penghuni kost masih menggunakan metode manual, seperti pencatatan dalam buku catatan, metode ini dapat mengakibatkan kesulitan dalam pengelolaan data pribadi penghuni kost, data pemesanan, dan data pembayaran. Oleh karena itu, penerapan sistem informasi kost sruntul berbasis *website* bertujuan untuk mempermudah proses pendataan penghuni kost, pencatatan, pembayaran, manajemen data pemesanan, serta pengarsipan data pribadi penghuni kost yang lebih terstruktur. [15]

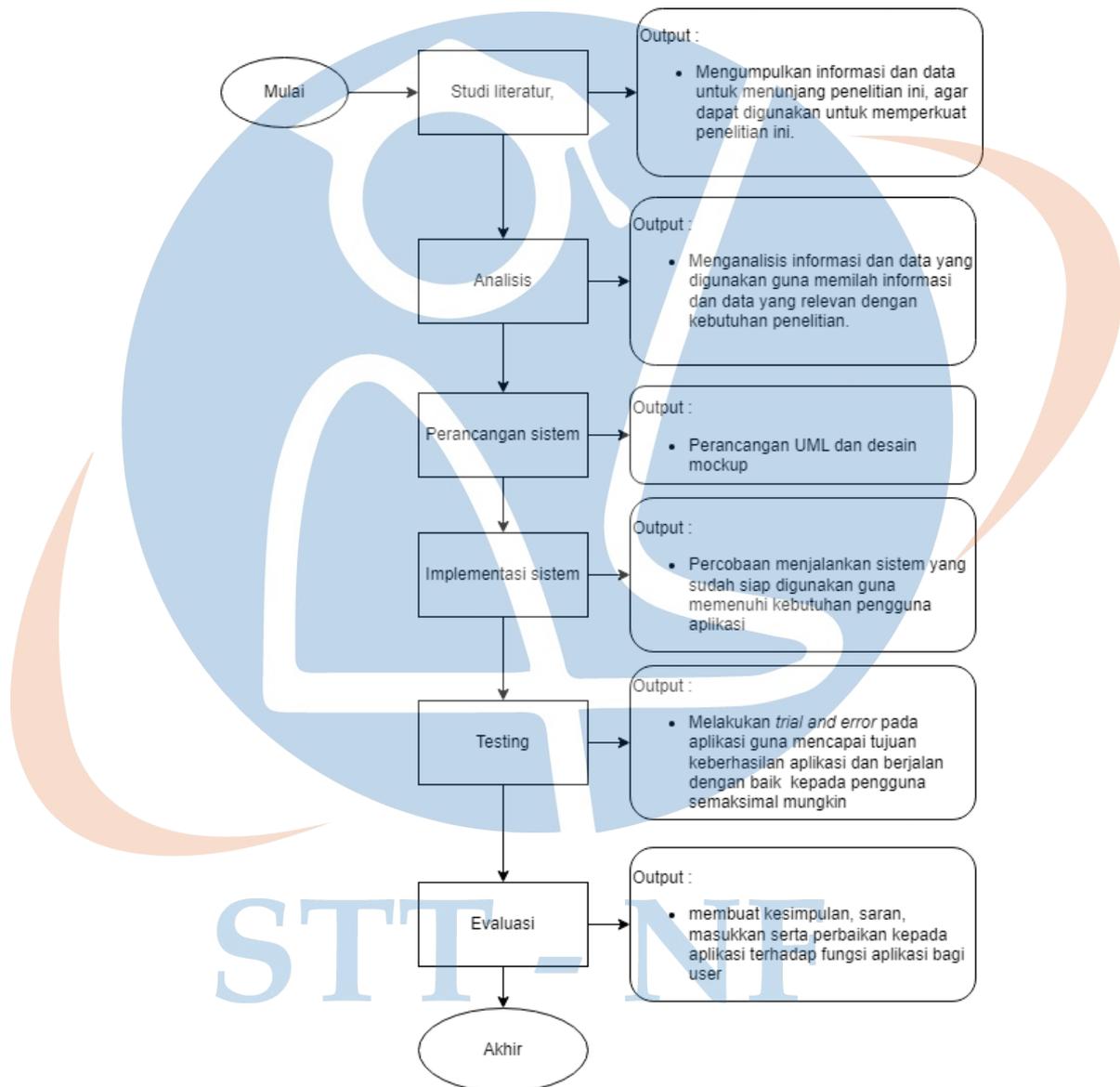
Pada posisi penelitian ini, peneliti berfokus dalam pengembangan sistem penagihan iuran rumah kost menggunakan *whatsapp api* yang bertujuan untuk melakukan penagihan iuran otomatis setiap bulannya dengan memanfaatkan *framework laravel* dan *whatsapp API*. Peneliti melakukan analisis mendalam pada penelitian yang sudah ada, yang dimana penelitian tersebut dapat memberikan inspirasi dalam aspek efisiensi dalam pembuatan penelitian ini.

STT - NF

BAB III METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Dibawah ini adalah langkah-langkah penelitian yang akan digunakan pada penulisan ini:



Gambar 3. 1 Tahapan Penelitian

Alur dalam tahapan penulisan ini dapat dilihat pada Gambar 3. 1 Tahapan-tahapan tersebut dapat diuraikan sebagai berikut:

1. **Studi literatur :** Mengumpulkan informasi dan data terkait penelitian ini, termasuk perkembangan dan konsep terbaru dalam penelitian ini.

2. **Analisis** : Menganalisa informasi dan data seperti mencari penyebab kendala pengelolaan, masalah terkait manajemen rumah kost, serta hambatan dalam proses manajemen bagi pemilik rumah kost.
3. **Perancangan sistem** : Pembuatan aplikasi penagihan rumah kost dengan memperhatikan kebutuhan pemilik rumah kost. Perancangan aplikasi dibuat sesuai pendekatan *UML (Unified Modeling Language)* dan *Mock-up*.
4. **Testing** : Dilakukan pengujian sistem guna memastikan sistem berjalan dengan baik dan memenuhi kebutuhan pengguna.
5. **Evaluasi** : Melakukan penilaian terhadap aplikasi guna mencapai tujuan kebutuhan user. Menarik kesimpulan apakah aplikasi yang sudah dapat memenuhi kebutuhan user atau belum dapat memenuhi kebutuhan user.

3.2. Rancangan Penelitian

Rancangan penelitian memiliki peran penting untuk memberikan panduan yang jelas dan tujuan yang terukur dalam proses penelitian. Rancangan penelitian memberikan gambaran alur proses yang dapat memecahkan sebuah permasalahan yang diangkat pada objek penelitian. Dengan tujuan penelitian yang jelas dan terukur, proses penelitian dan pemecahan masalah dapat menemukan solusi yang tepat untuk masalah yang dihadapi.

3.2.1. Jenis Penelitian

Penelitian ini menggunakan jenis penelitian pengembangan atau *research and development (R&D)*, tujuan dalam jenis penelitian ini adalah mengembangkan atau meningkatkan kualitas produk yang sudah ada .[16] dalam penelitian ini, peneliti mengembangkan sistem aplikasi penagihan iuran kost menggunakan *website*.

3.2.2. Metode Analisis

Dalam penelitian ini, penulis memilih untuk menggunakan metode analisis kualitatif. Sesuai dengan konsep yang dijelaskan oleh Walidin, Saifullah & Tabrani, metode ini berfokus pada pengamatan dan pemahaman fenomena yang terjadi dalam kehidupan nyata. Tujuan utamanya adalah untuk menyelidiki serta memahami secara mendalam alasan mengapa dan bagaimana suatu fenomena

terjadi yang dapat dalam kehidupan sehari-hari.[17] Dalam penelitian ini metode kualitatif digunakan untuk memenuhi berbagai aspek, seperti motivasi dan perilaku pemilik kost, keinginan dan kebutuhan oleh pemilik kost, permasalahan yang dihadapi oleh pemilik kost. Penelitian ini bertujuan untuk menghasilkan sebuah prototipe aplikasi untuk membantu pemilik rumah kost dalam pengihan iuran bulanan secara efisien.

3.2.3. Metode Pengumpulan Data

Dalam proses pengumpulan data penelitian ini, peneliti menerapkan berbagai metode dan teknik yang berbeda sumbernya. Metode yang digunakan antara lain studi literatur, wawancara dan observasi. Pendekatan ini dipilih untuk memastikan keakuratan dan kelengkapan informasi yang diperoleh dalam penelitian ini.

a. Studi literatur

Dalam metode ini menggunakan analisi mendalam terhadap literatur terkait penelitian ini. Informasi penelitian ini diperoleh dari sumber-sumber ilmiah seperti jurnal, artikel, buku referensi yang relevan. Metode ini digunakan untuk memahami konsep-konsep landasan teoritis, dan praktik-praktik terkini yang berkaitan dengan penelitian ini yaitu manajemen rumah kost.

b. Wawancara

Penelitian ini menggunakan metode wawancara untuk memperoleh pemahaman mendalam mengenai pengelolaan dan kebutuhan properti kost. Penulis melakukan wawancara dengan pemilik kost untuk menggali isu-isu yang dihadapi pemilik kost. Terdapat pertanyaan yang akan diajukan kepada pemilik kost, yang mencakup berbagai aspek terkait pengelolaan properti, harapan, dan kebutuhan properti kost. Sehingga data yang diperoleh dapat memberikan gambaran tentang situasi yang dihadapi pemilik kost.

c. Observasi

Metode observasi digunakan untuk mengamati proses terhadap pengelolaan manajemen rumah kost di lingkungan penelitian. Observasi didapatkan dari aktivitas-aktivitas seperti pengelolaan pembayaran,

pemeliharaan fasilitas, dan interaksi antara penghuni kost. Hasil observasi dicatat untuk mendapatkan pemahaman yang lebih mendalam mengenai kejadian riil yang ada di lingkungan penelitian.

3.2.4. Lingkungan Pengembangan

Dalam penelitian ini peneliti menggunakan *framework* dan *tools* yang telah dipilih akan digunakan untuk menunjang pengembangan aplikasi penagihan iuran kost. *Framework* yang digunakan adalah *laravel*, yang didukung oleh bahasa pemrograman *php*. Dan *tools* yang digunakan oleh peneliti adalah *visual studio code* sebagai *IDE* untuk pengembangan kode, *MySQL* sebagai basis data untuk menyimpan informasi, serta laptop *acer 5 aspire* sebagai perangkat utama dalam pengembangan aplikasi. Lokasi penelitian berada di Kabupaten Bogor, Kecamatan Gunung Putri, Desa Tlajung Udik sebagai lingkungan penelitian yang relevan dengan permasalahan yang dibahas. Narasumber utama penelitian ini adalah pemilik kost yang berasal langsung dari lingkungan tersebut.

3.2.5. Metode Pengujian

Pada penelitian ini, metode pengujian yang digunakan menggunakan metode *blackbox testing* dan *usability testing*. Tujuannya adalah untuk menilai kinerja sistem secara menyeluruh dan mengidentifikasi area yang memerlukan perbaikan guna meningkatkan keberhasilan berjalannya sistem secara menyeluruh.

3.2.5.1. Blackbox Testing

Blackbox testing adalah metode pengujian perangkat lunak yang berfokus pada evaluasi dan fungsionalitas sistem.[18] dalam *blackbox testing*, fokus utamanya adalah pengujian pada *input* dan *output* sistem tanpa memperhatikan detail implementasi internal. Metode ini bertujuan untuk mengevaluasi apakah sistem dapat berfungsi sesuai dengan yang diharapkan dari sudut pandang pengguna.

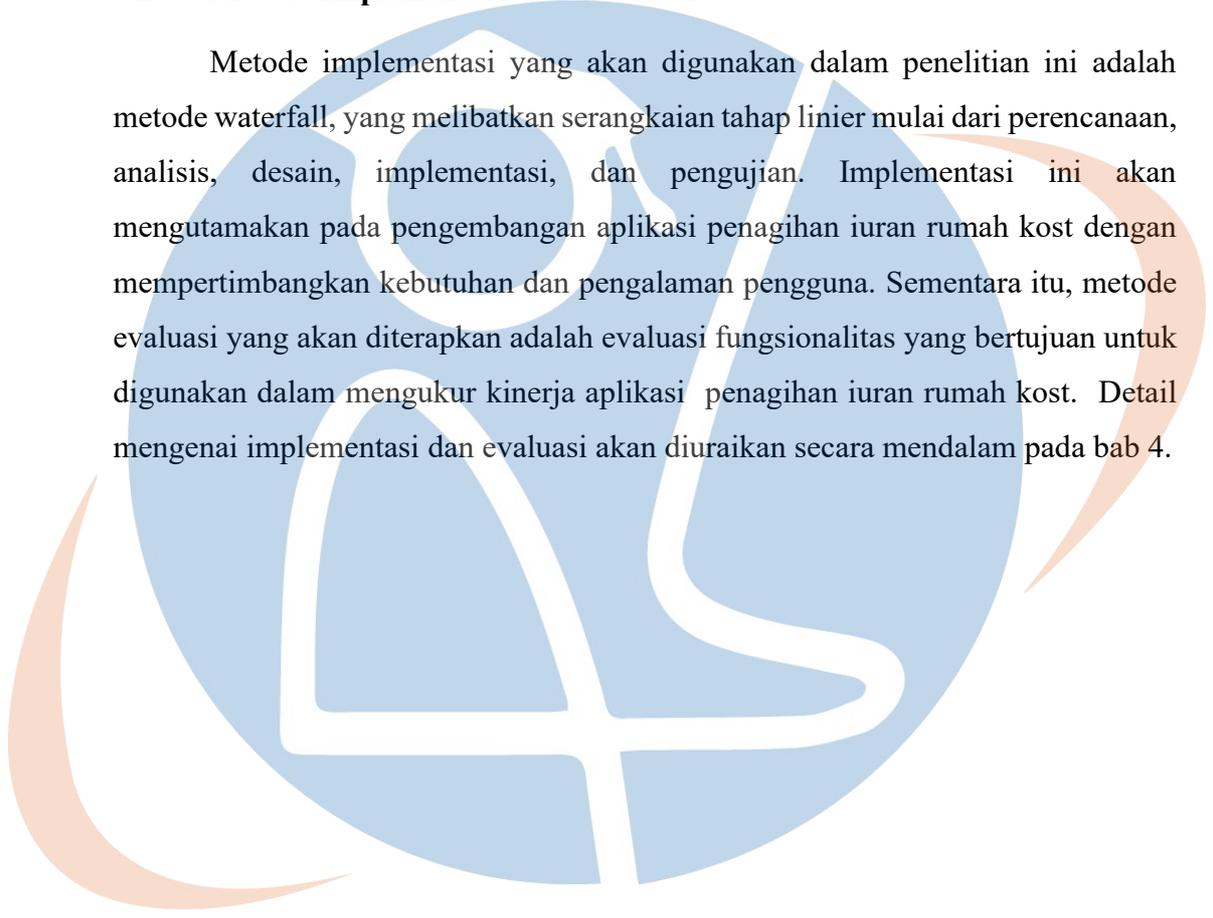
3.2.5.2. Usability Testing

Usability testing merupakan metode pengujian yang mengevaluasi pengalaman pengguna dalam memahami dan menggunakan teknologi, aplikasi atau situs *website* tertentu. Indikator-indikator dalam *usability testing* dipakai untuk

mengukur tingkat kepuasan pengguna dalam menggunakan teknologi, aplikasi, atau produk tersebut untuk mencapai yang diinginkan.[19] metode ini berfokus pada pengalaman pengguna dalam menggunakan sistem dengan tujuan memastikan bahwa sistem dapat memberikan tingkat kenyamanan, efisien, dan kemudahan bagi pengguna.

3.2.6. Metode Implementasi dan Evaluasi

Metode implementasi yang akan digunakan dalam penelitian ini adalah metode waterfall, yang melibatkan serangkaian tahap linier mulai dari perencanaan, analisis, desain, implementasi, dan pengujian. Implementasi ini akan mengutamakan pada pengembangan aplikasi penagihan iuran rumah kost dengan mempertimbangkan kebutuhan dan pengalaman pengguna. Sementara itu, metode evaluasi yang akan diterapkan adalah evaluasi fungsionalitas yang bertujuan untuk digunakan dalam mengukur kinerja aplikasi penagihan iuran rumah kost. Detail mengenai implementasi dan evaluasi akan diuraikan secara mendalam pada bab 4.



STT - NF

BAB IV IMPLEMENTASI DAN EVALUASI

4.1. Analisis dan Perancangan

4.1.1. Analisis sistem

Sebelum masuk ketahap perancangan sistem, peneliti terlebih dahulu melakukan analisis sistem yang telah berjalan, pada sistem yang telah berjalan ditemukan bahwa aplikasi untuk penagihan iuran rumah kost masih menggunakan cara manual dan pencatatan tagihan masih dilakukan secara tertulis. diharapkan dengan adanya sistem penagihan iuran rumah kost yang baru, proses penagihan iuran rumah kost dapat dilakukan secara otomatis dan lebih efisien, serta pencatatan tagihan iuran rumah kost dapat mengurangi dampak *human error* dan menjadi lebih terstruktur.

4.1.2. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan untuk merancang antarmuka pada aplikasi sistem penagihan iuran rumah kost sebagai berikut:

Tabel 4. 1 Kebutuhan Perangkat Keras

| Nama | Spesifikasi Minimum | Jumlah | Fungsi |
|---------------------------|---|--------|---|
| <i>Server</i> | <i>Processor: Intel Xeon, RAM: 16GB, Storage: 1TB SSD</i> | 1 | Untuk menjalankan aplikasi dan <i>database</i> dari sistem penagihan iuran rumah kost |
| Laptop Klien | <i>Processor: Intel i3, RAM: 8Gb, Storage: 500GB SSD</i> | 2 | Untuk Penggunaan Admin dan Pengelola |
| <i>Backup Storage</i> | Jenis: <i>External HDD</i> , Kapasitas: <i>2 TB</i> | 1 | Untuk penyimpanan backup data |
| <i>Mouse dan Keyboard</i> | Standar Pabrik | 1 | Untuk dapat menggunakan <i>server</i> |
| <i>Monitor</i> | Ukuran Layar: <i>21 inc</i> | 1 | Untuk menampilkan layar <i>server</i> |

4.1.3. Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk merancang antarmuka pada aplikasi sistem penagihan iuran rumah kost sebagai berikut:

Tabel 4. 2 Kebutuhan Perangkat Lunak

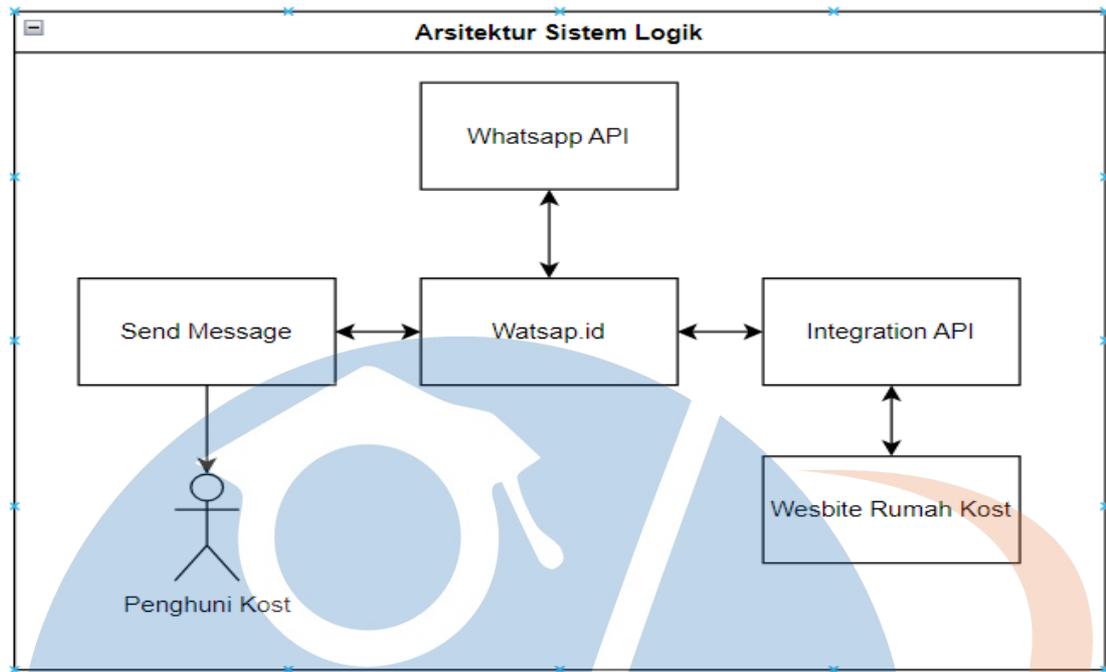
| No | Perangkat lunak | Fungsi |
|----|------------------------|---|
| 1 | <i>Laravel 11</i> | <i>Framework</i> untuk pengembangan aplikasi web. |
| 2 | <i>Balsamiq</i> | Alat untuk pembuatan <i>wireframe</i> . |
| 3 | <i>App.diagram.net</i> | Alat untuk pembuatan <i>activity diagram</i> . |
| 4 | <i>Visual-paradigm</i> | Alat untuk pembuatan <i>usecase diagram</i> . |
| 5 | <i>MySQL workbench</i> | Alat untuk desain dan manajemen <i>database</i> . |
| 6 | <i>Brave</i> | <i>Browser</i> untuk pengujian aplikasi web. |

4.2. Perancangan Sistem

Perancangan sistem adalah proses merencanakan, merancang, dan mengembangkan sistem aplikasi yang akan dibuat secara terstruktur dan terorganisir. Tujuan dari perancangan sistem adalah untuk menciptakan alur yang efisien dalam membuat fungsionalitas aplikasi yang diinginkan dan dapat memenuhi kebutuhan pengguna. Dalam tahapan perancangan sistem, peneliti akan merancang dua aspek utama, yaitu perancangan arsitektur sistem logik dan perancangan arsitektur sistem fisik. Perancangan tersebut dijelaskan sebagai berikut:

4.1.1. Perancangan Arsitektur Sistem Logik

Pada penelitian ini peneliti melakukan perancangan arsitektur sistem logik dengan memanfaatkan *third party* atau pihak ketiga yang menghubungkan layanan *whatsapp API* dengan *website* rumah kost. Dalam hal ini, *third party* bertindak sebagai perantara antara layanan *whatsapp API* dan *website* rumah kost. Pihak ketiga ini bertanggung jawab dalam meneruskan pesan-pesan yang akan dikirimkan kepada penghuni rumah kost dari pengguna *website* rumah kost. Dengan adanya integrasi ini, penagihan iuran rumah kost kepada penghuni dapat menjadi lebih efisien dan terorganisir. Berikut ini adalah ilustrasi dari arsitektur sistem logik yang digunakan:

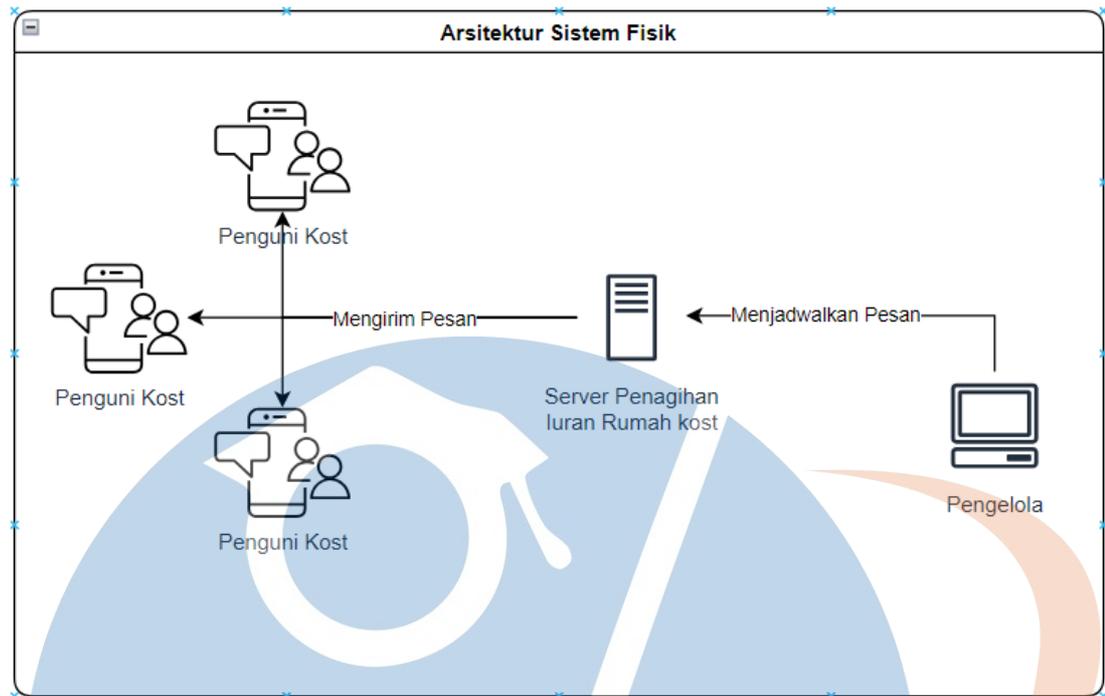


Gambar 4. 1 Rancangan Arsitektur Sistem Logik

Gambar di atas menunjukkan perancangan dari sistem arsitektur logik yang akan berjalan pada *website* rumah kost, pada penelitian ini *website* rumah kost akan mengirimkan pesan kepada penghuni kost melalui *API* yang telah disediakan oleh layanan pihak ketiga yakni *watsapp.id*. *watsapp.id* menerima pesan dari *website* rumah kost dan meneruskannya ke *Whatsapp API* untuk dikirimkan ke penerima akhir yaitu Penghuni Kost.

4.1.2. Perancangan Arsitektur Sistem Fisik

Pada tahapan ini adalah proses dari perencanaan dan penataan komponen perangkat keras yang akan digunakan dalam penelitian ini. Perancangan ini mencakup pemilihan perangkat keras yang tepat, penentuan lokasi komponen fisik server, pengelola, dan penghuni kost. Serta pengaturan koneksi antara komponen-komponen tersebut. Dengan merancang sistem fisik yang baik, diharapkan aplikasi sistem penagihan iuran rumah kost dapat berjalan dengan lancar dan optimal sesuai dengan harapan pengguna. Berikut ini ilustrasi dari arsitektur sistem fisik yang digunakan:



Gambar 4. 2 Rancangan Arsitektur Sistem Fisik

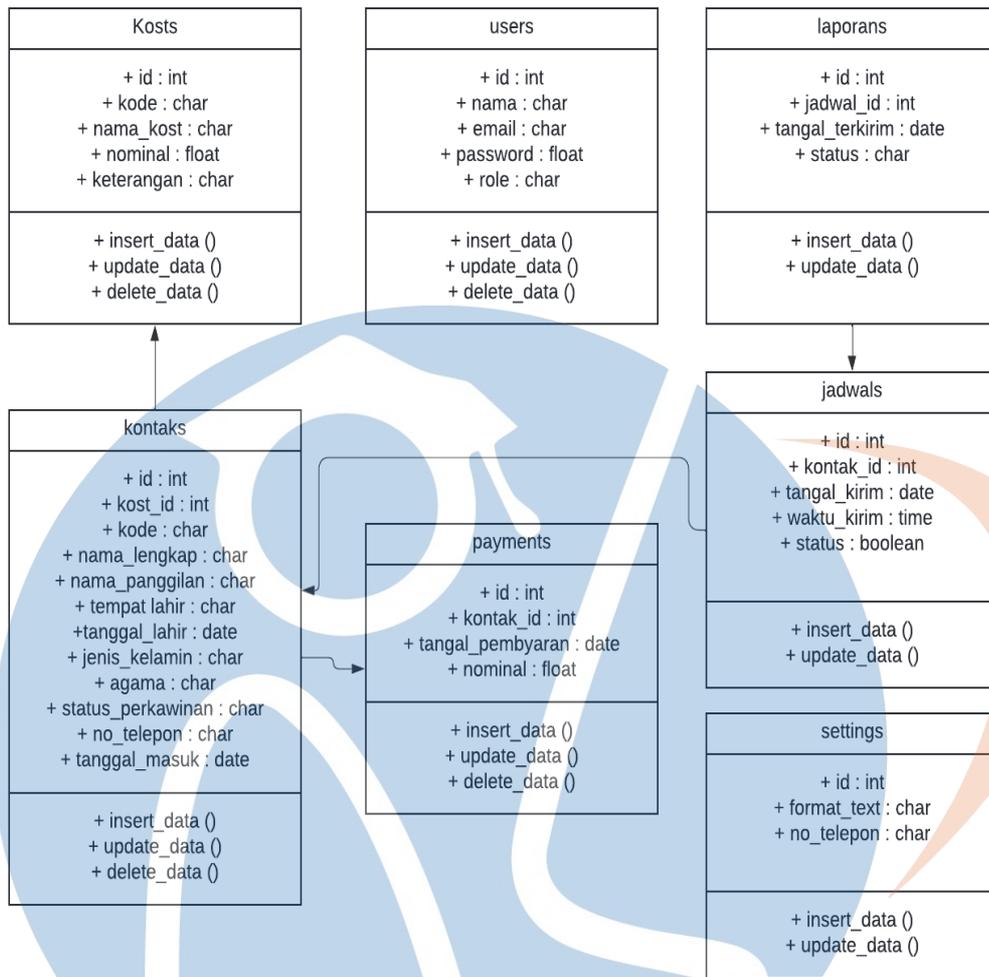
Gambar di atas merupakan rancangan arsitektur sistem fisik dimana pengelola dapat menjadwalkan pesan yang akan dikirimkan kepada penghuni kost, setelah jadwal pesan dibuat maka website rumah kost akan memproses permintaan tersebut lalu mengirimkan pesan sesuai jadwal yang telah ditentukan, dan selanjutnya pesan yang dijadwalkan akan diterima oleh penghuni kost.

4.1.3. Perancangan UML (Unified Modeling Language)

Perancangan UML adalah langkah-langkah dalam perancangan sistem seperti membuat *usecase diagram*, *activity diagram*, dan *class diagram*. Hal ini agar dapat memastikan bahwa semua fungsi dan alur kerja yang telah direncanakan dapat berjalan dengan baik sesuai dengan harapan.[14]

4.1.3.1. Class Diagram

Class diagram sangat membantu dalam pembuatan UML karena memberikan gambaran yang jelas tentang struktur sistem, termasuk deskripsi kelas, atribut, dan objek yang digunakan oleh sistem tersebut.[14] berikut ini adalah ilustrasi dari *class diagram*:

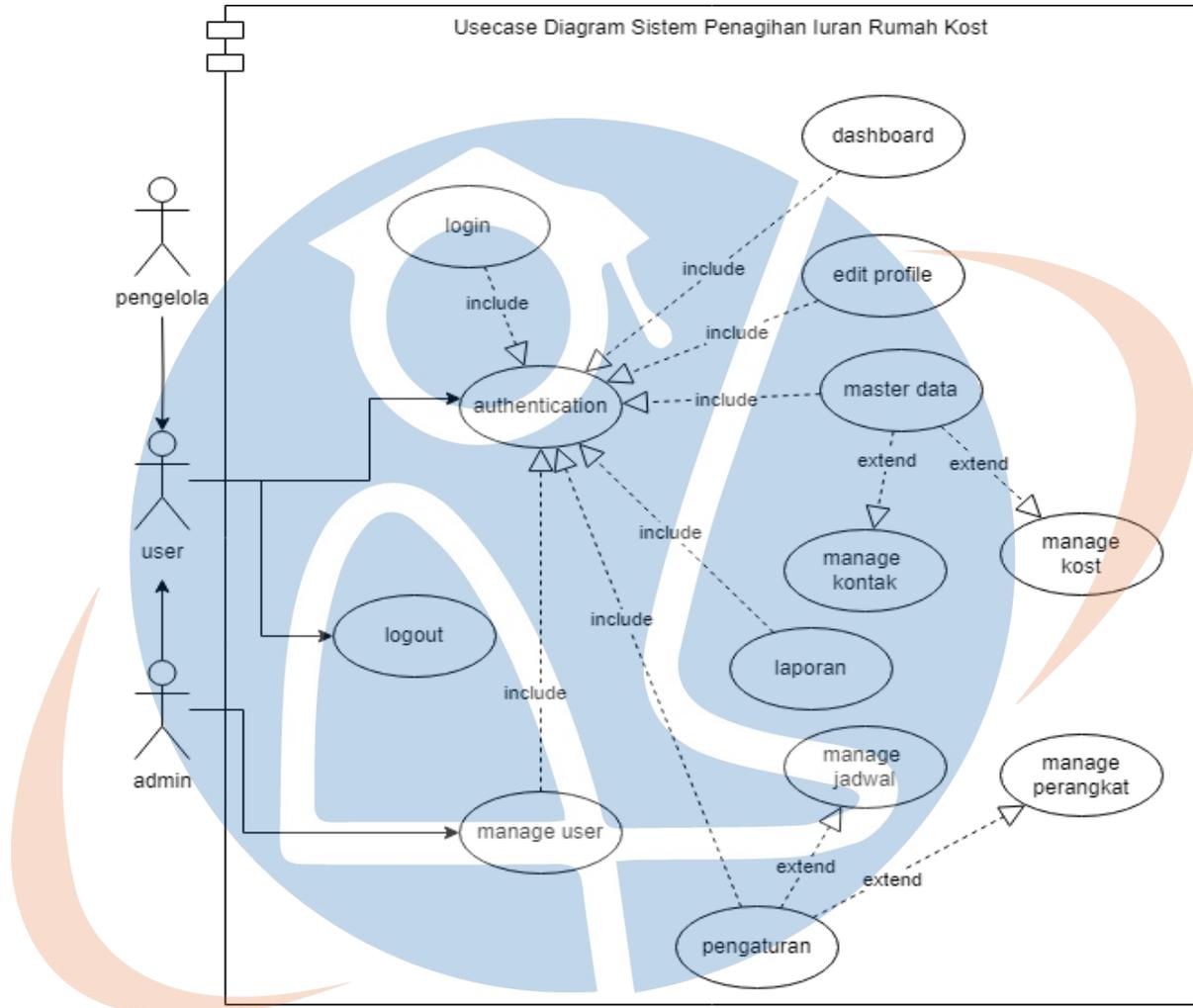


Gambar 4. 3 Class Diagram - Rumah Kost

Gambar 4. 3 merupakan *class diagram* dari aplikasi rumah kost yang akan digunakan untuk mengidentifikasi entitas, seperti data kunci, dan hubungan antara data yang akan mewakili suatu *model*. Pada *class diagram* tersebut terdiri dari *class* kost, *users*, laporan, kontak, *payment*, jadwal, dan *setting*. Dalam *class* masing-masing terdapat atribut dan metode yang memungkinkan pengelolaan data terkait manajemen kost dan manajemen penagihan iuran kost. Pada penelitian ini, peneliti lebih mengutamakan kelas utama seperti *user*, laporan, *setting*, jadwal, dan kontak. Tujuannya adalah untuk melakukan penagihan iuran rumah kost menggunakan data yang sudah ada pada kelas tersebut agar dapat dilakukan penagihan iuran kepada penghuni kost.

4.1.3.2. Usecase Diagram Sistem Penagihan Iuran Rumah Kost

Usecase diagram bertujuan untuk membuat ilustrasi dari aktivitas yang dapat dilakukan oleh *user* pada aplikasi penagihan iuran rumah kost. Berikut ini adalah ilustrasi *usecase diagram* dari sistem penagihan iuran rumah kost.

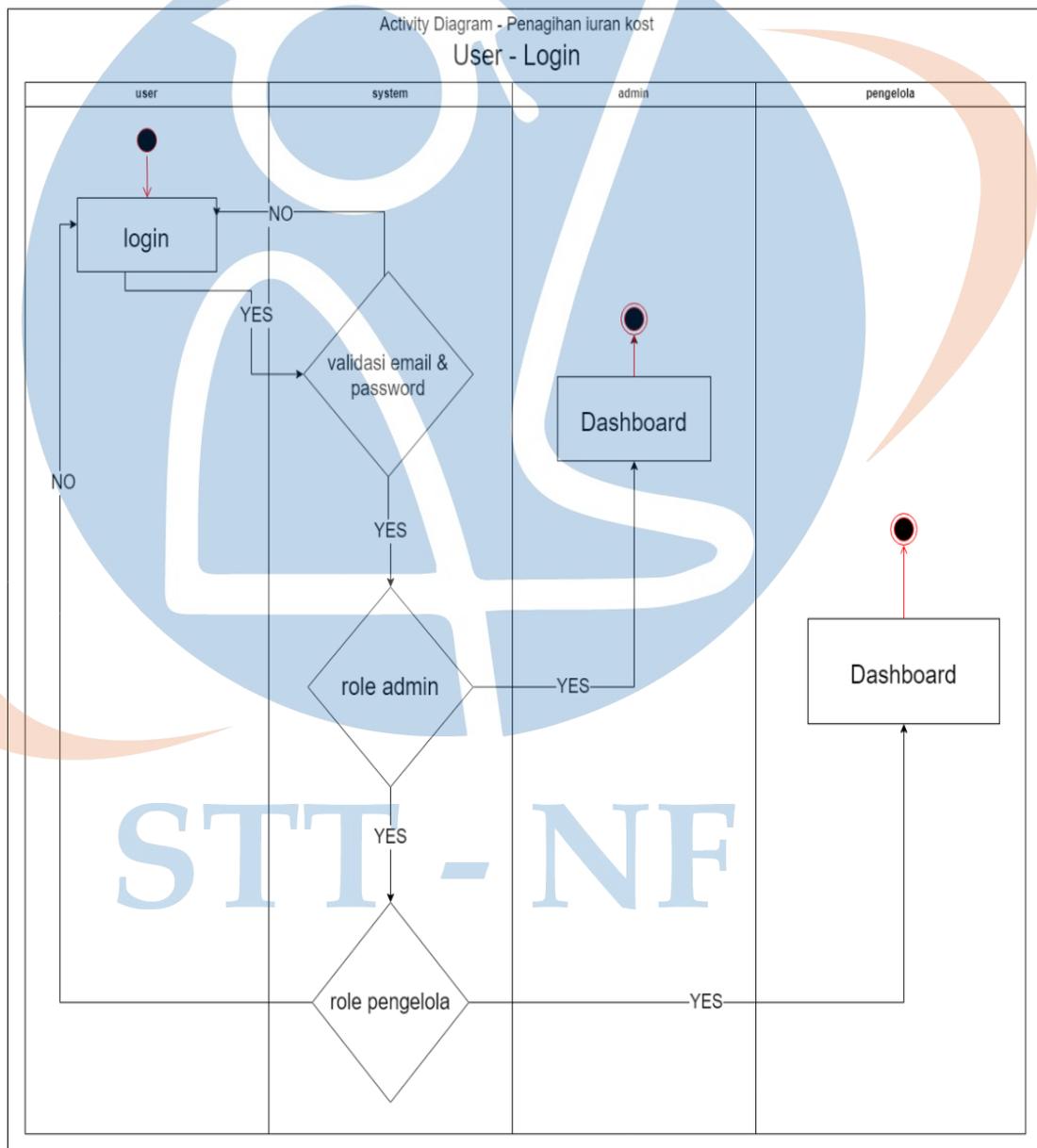


Gambar 4. 4 Usecase Diagram Sistem Penagihan Iuran Rumah Kost

Gambar di atas merupakan ilustrasi dari fitur yang dapat diakses oleh *user* pengelola dan admin di dalam *website* penagihan iuran rumah kost. Terdapat *menu* tambahan yaitu *manage user* yang hanya dapat diakses oleh peran admin.

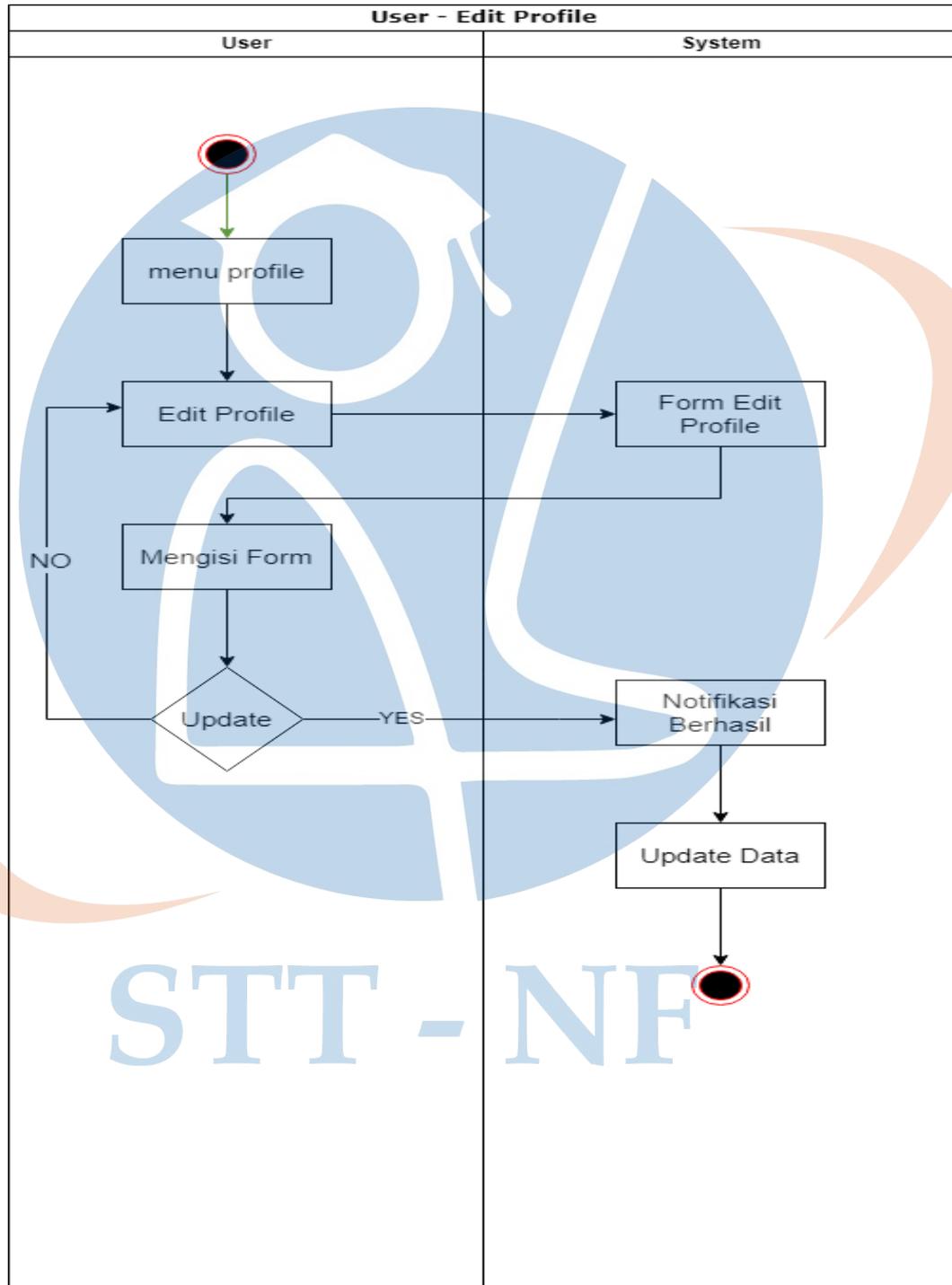
4.1.2.2. Activity Diagram

Activity diagram adalah ilustrasi visual yang dapat memodelkan urutan langkah-langkah secara terstruktur atau proses yang terjadi di dalam sistem. *Activity* diagram dapat digunakan untuk memahami alur kerja sistem secara lebih detail. Dengan menggunakan diagram ini, pengembang sistem dapat mengidentifikasi proses-proses yang terlibat, dan memudahkan dalam pemahaman dan pengembangan sistem. [3] Berikut ini adalah ilustrasi *activity diagram* dari sistem penagihan iuran rumah kost.



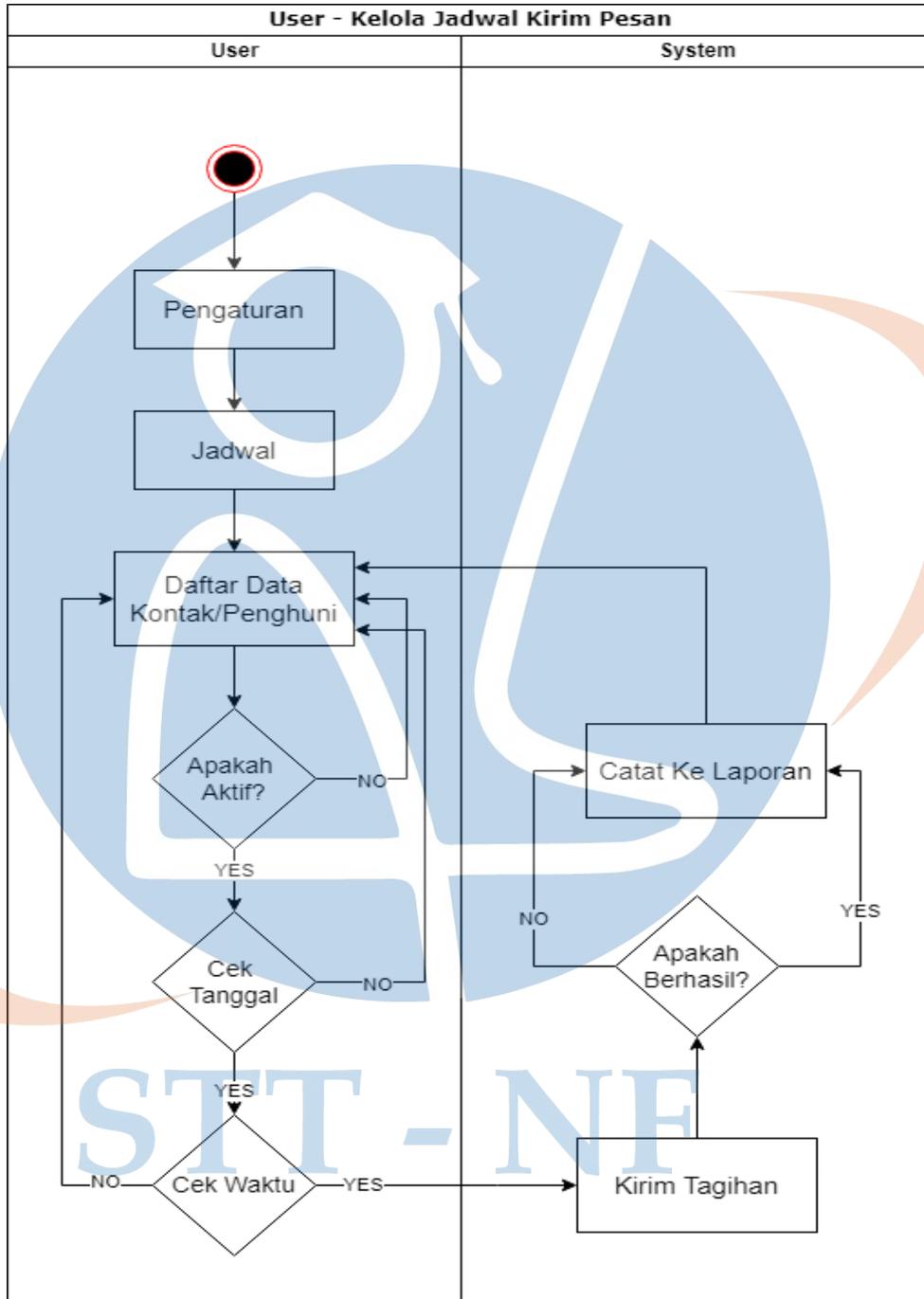
Gambar 4. 5 Activity Diagram - User Login

Gambar di atas merupakan *activity diagram user – login* yang merupakan proses *login* pada aplikasi rumah kost, yang dimana *user* akan melakukan *login* lalu sistem akan mengidentifikasi *role* dari *user* ketika *login* lalu mengarahkannya ke menu *dashboard* sesuai dengan *role* dari *user*.



Gambar 4. 6 Activity Diagram - Edit Profile

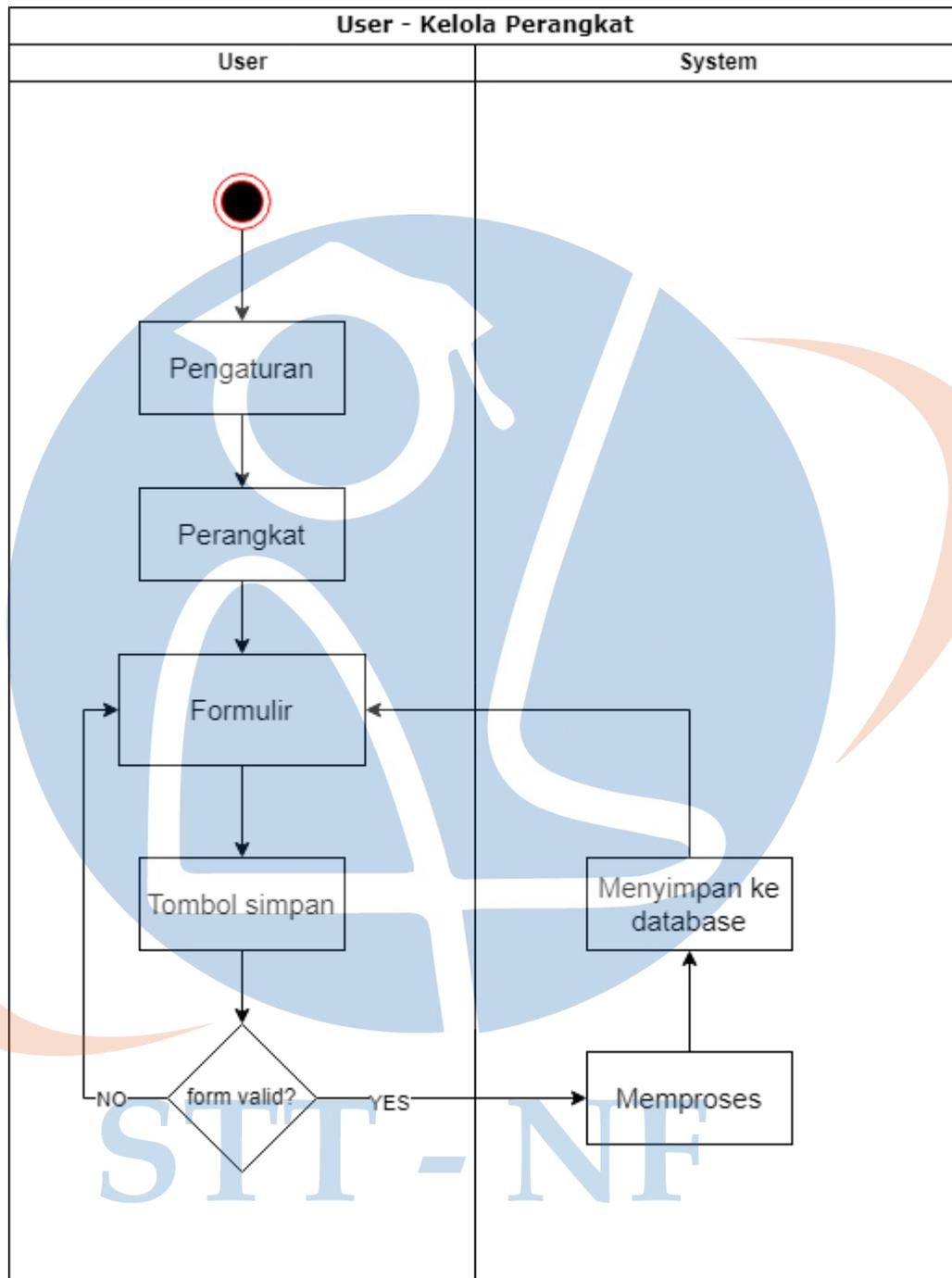
Gambar di atas merupakan proses dari *edit profile user* diaplikasi rumah kost. *Activity diagram* ini merupakan proses user agar dapat mengubah informasi pribadi dari *user*.



Gambar 4. 7 Activity Diagram - User Kelola Jadwal Tagihan

Gambar 4. 7 merupakan proses dari user yang dapat mengelola jadwal tugas penagihan iuran kost secara otomatis diaplikasi. User dengan role admin dan

pengelola dapat melakukan penjadwalan kepada penghuni kost bagi yang telah didaftarkannya nomor handphone penghuni kost di aplikasi ini.

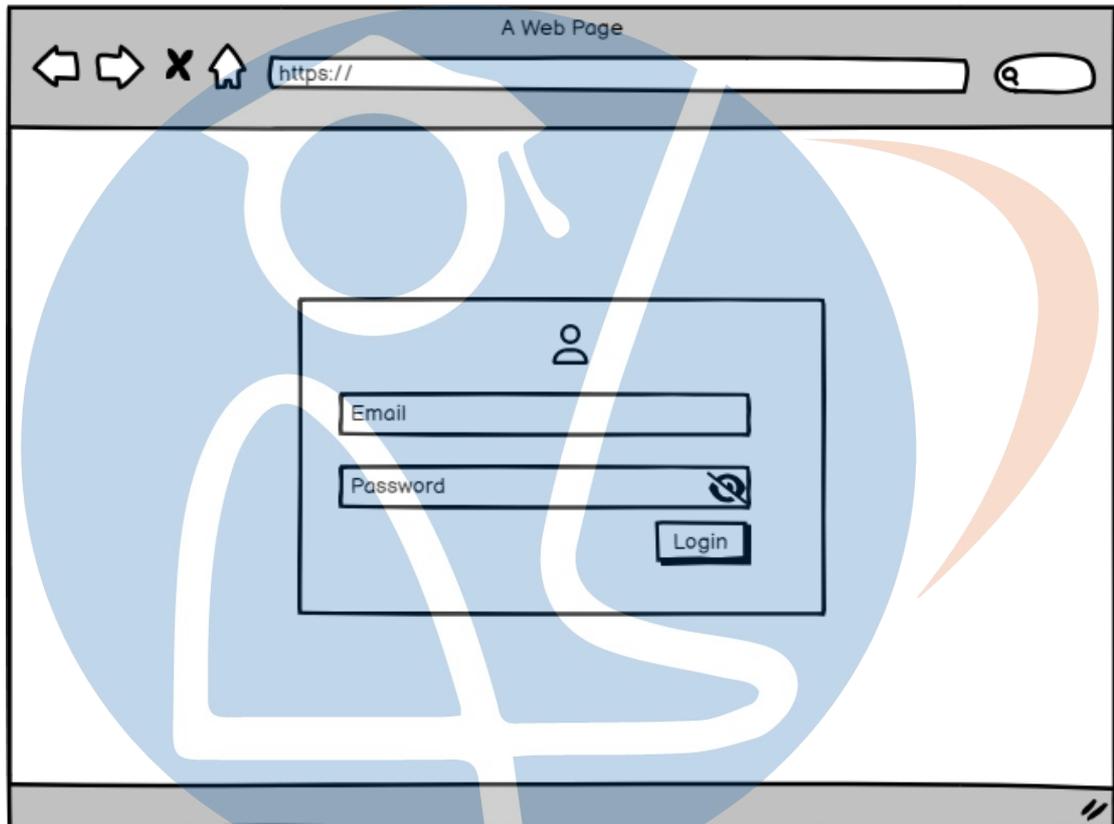


Gambar 4. 8 Activity Diagram - User Kelola Perangkat

Gambar 4. 8 adalah proses dari user yang dapat mengelola perangkat. didalam formulir menu ini terdapat nomor telpon untuk mengirimkan pesan dan format pesan yang nantinya pesan tersebut akan dikirimkan kepada penghuni kost melalui menu jadwal.

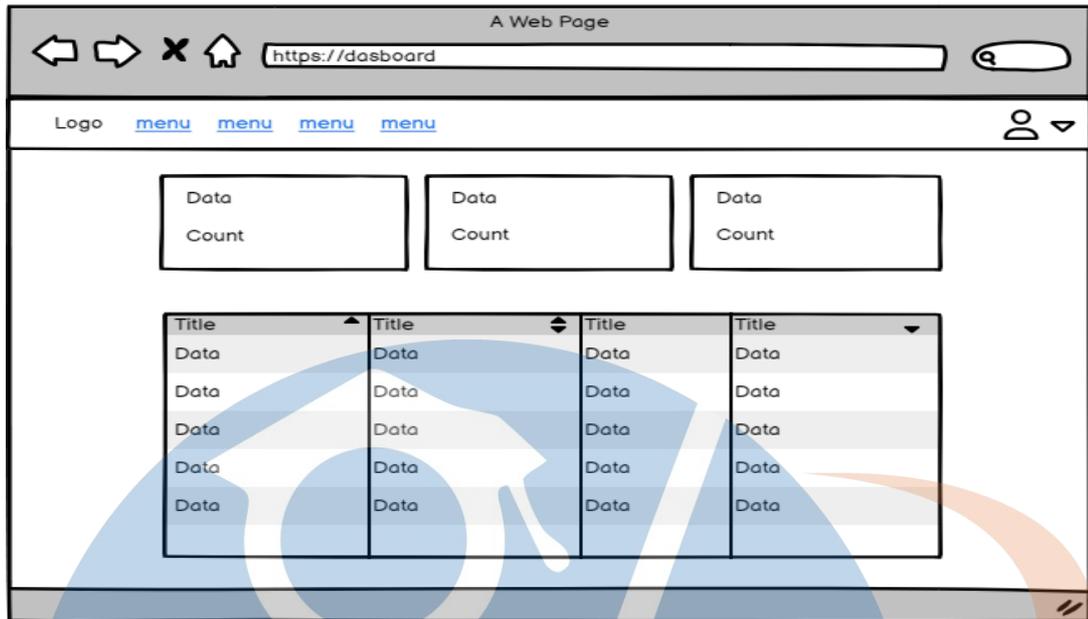
4.1.4. Perancangan User Interface

Tahap terakhir ini adalah perancangan antarmuka pengguna yang bertujuan untuk menggambarkan fitur-fitur yang akan dibuat untuk aplikasi penagihan iuran kost juga untuk mempermudah pembuatan bagi pengembang terhadap fungsionalitas yang disediakan. Berikut ini adalah rancangan *user interface* dari sistem penagihan iuran rumah kost.



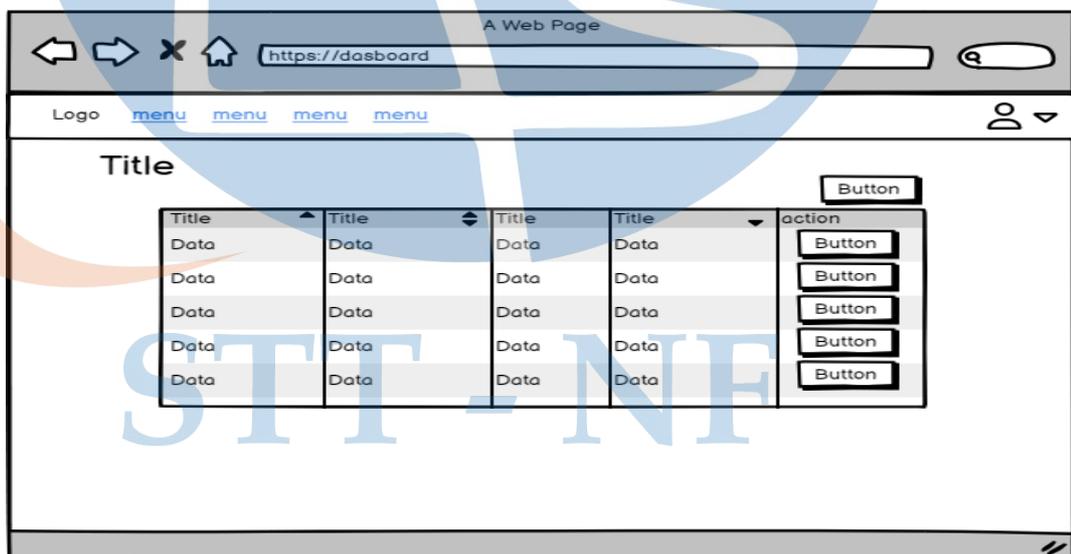
Gambar 4. 9 User Interface – Login

Gambar 4. 9 adalah *user interface* dari fitur *login*. Pada halaman *login user* akan diminta memasukkan *email* dan *password*. Jika *user* sudah mengisi formulir maka *user* lalu *click* tombol “LOG IN” untuk masuk kedalam *dashboard*.



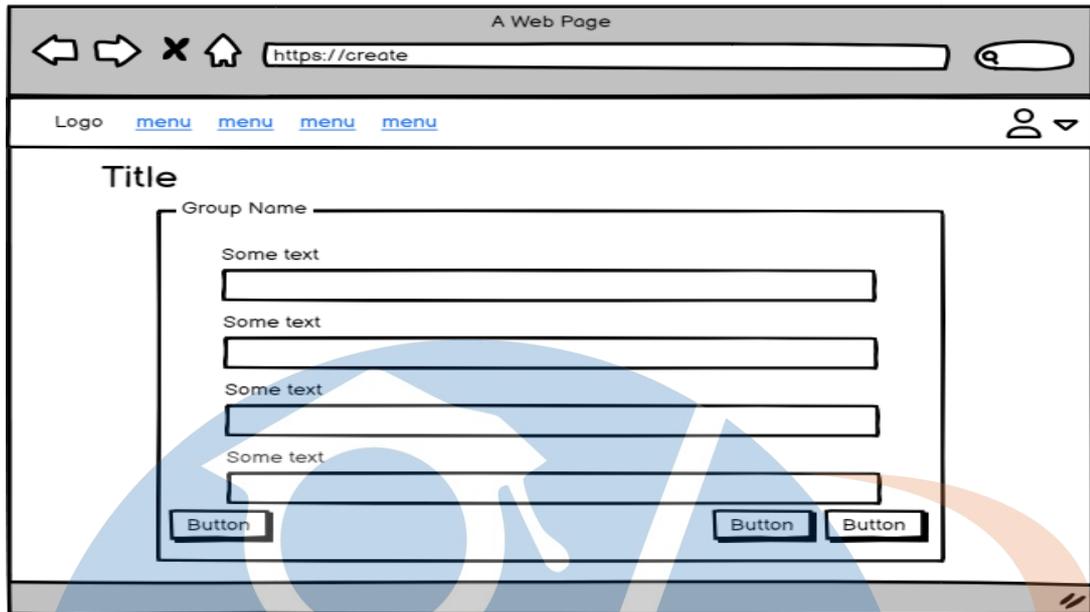
Gambar 4. 10 User Interface – Dashboard

Gambar 4. 10 adalah tampilan awal *user interface* setelah *user* berhasil masuk ketika login. Pada halaman ini akan terdapat informasi dari data yang sudah ada seperti informasi kontak, Jadwal aktif, laporan pesan terkirim, dan Laporan Pembayaran.



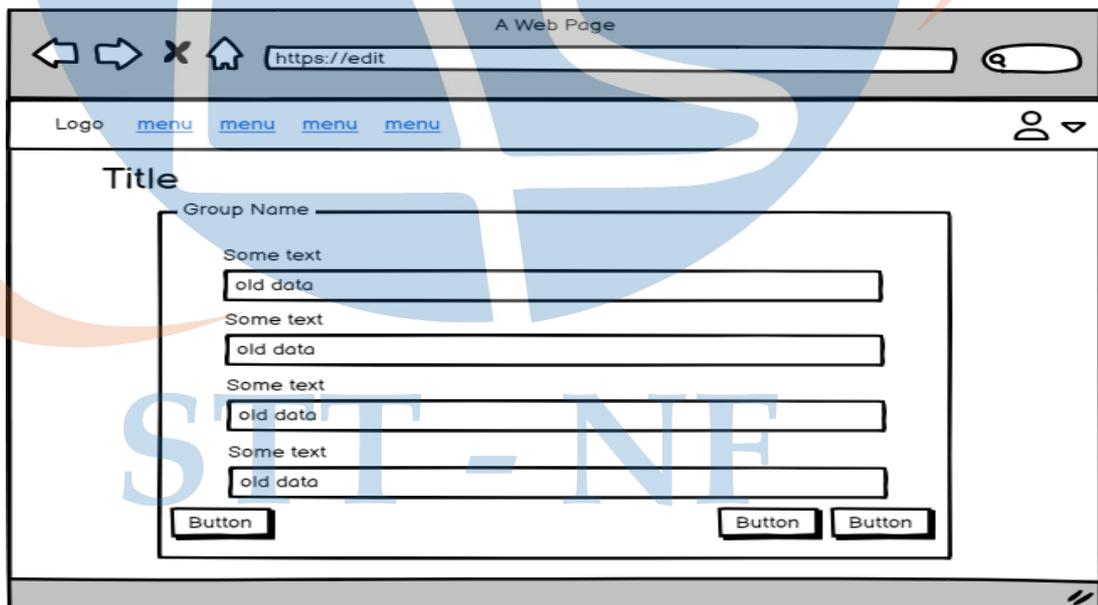
Gambar 4. 11 User Interface - Daftar Data

Gambar 4. 11 adalah tampilan antarmuka pengguna yang akan digunakan untuk menampilkan daftar data, tampilan ini akan digunakan untuk mencakup *menu* kontak, kost, laporan pembayaran, laporan pesan terkirim, dan *user*.



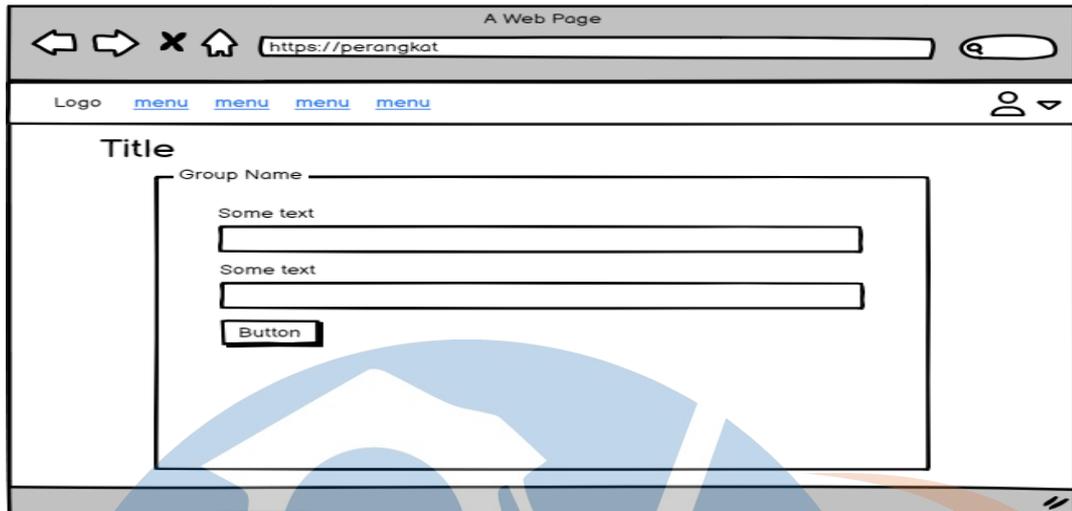
Gambar 4. 12 User Interface - Formulir Data

Gambar 4. 12 merupakan *user interface* untuk tampilan formulir data yang akan digunakan, formulir ini berfungsi untuk menambah data baru atau merubah data yang sudah ada, tampilan ini digunakan pada *menu* kontak, kost, laporan pembayaran, dan *user*.



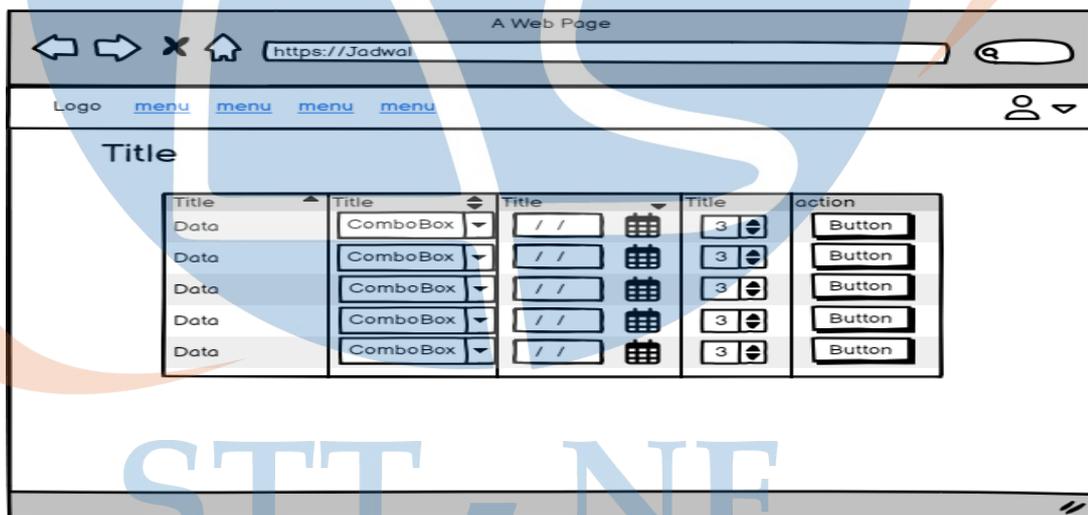
Gambar 4. 13 User Interface - Edit Profile

Gambar 4. 13 merupakan *user interface* dari *edit profile*. Pada fungsi ini setiap *user* dapat mengubah *profile* pribadi seperti nama, *email* dan *password*.



Gambar 4. 14 User Interface – Perangkat

Gambar 4. 14 merupakan *user interface* untuk pengaturan perangkat. dalam *menu* ini *role* admin memiliki kemampuan untuk nomor telepon dan format teks yang akan dikirimkan sebagai pesan kepada penghuni kost. Sementara itu, *role* pengelola hanya memiliki wewenang untuk mengubah format teks pesan yang akan dikirimkan kepada penghuni kost.



Gambar 4. 15 User Interface - Jadwal Pengiriman Pesan

Gambar 4. 15 merupakan *user interface* untuk pengaturan jadwal pengiriman pesan kepada penghuni kost. Yang nantinya semua *role* dapat mengelola jadwal mengirim pesan, seperti mengubah status pengiriman pesan, mengubah tanggal dan waktu mengirim pesan kepada penghuni kost.

4.1.5. Perancangan Menu

Pada aplikasi website penagihan iuran kost didapatkan kebutuhan pengguna yang dibutuhkan sebagai berikut:

1. Kebutuhan Pengguna

Tabel 4. 3 Kebutuhan Pengguna

| No | Keterangan | Pengguna |
|----|--|------------------|
| 1 | <i>Login dan Logout</i> | Admin, Pengelola |
| 2 | Informasi <i>Dashboard</i> | Admin, Pengelola |
| 3 | Melihat informasi dan kelola Laporan | Admin, Pengelola |
| 4 | Melihat informasi dan kelola Kost | Admin, Pengelola |
| 5 | Melihat informasi dan kelola Jadwal | Admin, Pengelola |
| 6 | Melihat informasi dan kelola Pengaturan | Admin, Pengelola |
| 7 | Melihat informasi dan kelola <i>User</i> | Admin |
| 8 | Kelola Profile | Admin, Pengelola |

Pada Tabel 4. 3 menjelaskan rincian tentang kebutuhan pengguna yang perlu dipenuhi oleh aplikasi website penagihan iuran kost. Tabel tersebut menguraikan fungsi-fungsi utama yang diperlukan oleh *user*, serta mengidentifikasi siapa yang berhak mengakses setiap fungsi. Tabel diatas mengindikasikan bahwa pengguna admin dan pengelola kost memiliki akses sebagian besar fungsi sistem. Kecuali fitur melihat informasi dan user yang hanya bisa diakses oleh admin. Informasi dalam tabel ini memberikan gambaran sistematis tentang fungsionalitas sistem dan dapat memastikan dapat memenuhi kebutuhan pengguna dengan tepat dan efisien.

2. Menu aplikasi

Tabel 4. 4 Menu Aplikasi

| Pengguna | Menu |
|----------|--|
| Admin | <ol style="list-style-type: none">1. <i>Dashboard</i>2. Data Master<ol style="list-style-type: none">a. Kostb. Kontak3. Laporan4. Pengaturan<ol style="list-style-type: none">a. Jadwalb. Perangkatc. <i>User</i>5. <i>Edit Profile</i> |

| Pengguna | Menu |
|----------------|---|
| Pengelola Kost | <ol style="list-style-type: none"> 1. <i>Dashboard</i> 2. Data Master <ol style="list-style-type: none"> a. Kost b. Kontak 3. Laporan 4. Pengaturan <ol style="list-style-type: none"> a. Jadwal b. Perangkat 5. <i>Edit Profile</i> |

Tabel 4. 4 menjelaskan berbagai menu yang dapat diakses oleh pengguna didalam aplikasi. *User* dengan peran admin memiliki akses penuh terhadap menu aplikasi seperti menu Informasi *dashboard*, kost, kontak, laporan, jadwal, perangkat, kelola *profile*, dan *user*. Dan dalam diperan pengelola kost terdapat sebagian besar *menu* yang sama dengan peran admin, yang berbeda hanya tidak dapat melihat menu *user*.

4.1.6. Perancangan Pengujian

Rancangan pengujian adalah sebuah perencanaan yang meliputi serangkaian kegiatan untuk menilai kinerja suatu sistem atau program. Tujuan dari pengujian ini adalah untuk memperoleh hasil yang menentukan sejauh mana sistem atau program tersebut memenuhi harapan yang telah ditetapkan. Pengujian ini akan dilakukan dengan menggunakan dua metode yaitu, pengujian *blackbox testing*, dan *usability testing*.

1. *Blackbox Testing*

Dalam rancangan pengujian yang pertama yaitu menggunakan *blackbox testing* yang akan dilakukan dengan membuat skenario pengujian untuk memperoleh hasil yang sesuai dengan ekspetasi pengguna. Berikut ini adalah tabel pengujian *blackbox* untuk menilai hasil dari ekspetasi penggunaan:

Tabel 4. 5 Perancangan Pengujian Blackbox - Login

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|-------------|---|-----------------------------|--------------|
| 1 | <i>Login</i> | Admin 1 | Pengujian proses <i>login</i> dengan data benar | Halaman utama ditampilkan | Ya / Tidak |
| 2 | <i>Login</i> | Admin 1 | Pengujian proses <i>login</i> dengan data salah | Pesan kesalahan ditampilkan | Ya / Tidak |
| 3 | <i>Login</i> | Admin 2 | Pengujian proses <i>login</i> dengan data benar | Halaman utama ditampilkan | Ya / Tidak |
| 4 | <i>Login</i> | Admin 2 | Pengujian proses <i>login</i> dengan data salah | Pesan kesalahan ditampilkan | Ya / Tidak |
| 5 | <i>Login</i> | Pengelola 1 | Pengujian proses <i>login</i> dengan data benar | Halaman utama ditampilkan | Ya / Tidak |
| 6 | <i>Login</i> | Pengelola 1 | Pengujian proses <i>login</i> dengan data salah | Pesan kesalahan ditampilkan | Ya / Tidak |
| 7 | <i>Login</i> | Pengelola 2 | Pengujian proses <i>login</i> dengan data benar | Halaman utama ditampilkan | Ya / Tidak |
| 8 | <i>Login</i> | Pengelola 2 | Pengujian proses <i>login</i> dengan data salah | Pesan kesalahan ditampilkan | Ya / Tidak |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|-------------|---|-----------------------------|--------------|
| 9 | <i>Login</i> | Pengelola 3 | Pengujian proses <i>login</i> dengan data benar | Halaman utama ditampilkan | Ya / Tidak |
| 10 | <i>Login</i> | Pengelola 3 | Pengujian proses <i>login</i> dengan data salah | Pesan kesalahan ditampilkan | Ya / Tidak |

Tabel 4. 6 Perancangan Pengujian Blackbox - Mengelola Data Kost

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---------------------|-------------|---|---|--------------|
| 1 | Mengelola data kost | Admin 1 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Ya / Tidak |
| 2 | Mengelola data kost | Admin 2 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Ya / Tidak |
| 3 | Mengelola data kost | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Ya / Tidak |
| 4 | Mengelola data kost | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Ya / Tidak |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---------------------|-------------|---|---|--------------|
| 5 | Mengelola data kost | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Ya / Tidak |

Tabel 4. 7 Perancangan Pengujian Blackbox - Mengelola Data Kontak

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|-----------------------|-------------|---|---|--------------|
| 1 | Mengelola data Kontak | Admin 1 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Ya / Tidak |
| 2 | Mengelola data Kontak | Admin 2 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Ya / Tidak |
| 3 | Mengelola data Kontak | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Ya / Tidak |
| 4 | Mengelola data Kontak | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Ya / Tidak |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|-----------------------|-------------|---|---|--------------|
| 5 | Mengelola data Kontak | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Ya / Tidak |

Tabel 4. 8 Perancangan Pengujian Blackbox - Mengelola Laporan Pembayaran

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|-------------------|-------------|---|--|--------------|
| 1 | Mengelola Laporan | Admin 1 | Menguji proses menambah, mengubah, dan menghapus Laporan Pembayaran | Data berhasil ditambahkan, diubah, dihapus | Ya / Tidak |
| 2 | Mengelola Laporan | Admin 2 | Menguji proses menambah, mengubah, dan menghapus Laporan Pembayaran | Data berhasil ditambahkan, diubah, dihapus | Ya / Tidak |
| 3 | Mengelola Laporan | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus Laporan Pembayaran | Data berhasil ditambahkan, diubah, dihapus | Ya / Tidak |
| 4 | Mengelola Laporan | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus Laporan Pembayaran | Data berhasil ditambahkan, diubah, dihapus | Ya / Tidak |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|-------------------|-------------|---|--|--------------|
| 5 | Mengelola Laporan | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus Laporan Pembayaran | Data berhasil ditambahkan, diubah, dihapus | Ya / Tidak |

Tabel 4. 9 Perancangan Pengujian Blackbox - Pengaturan Perangkat

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|--|-------------|---|--|--------------|
| 1 | Menu pengaturan > perangkat, peran admin | Admin 1 | Pengujian proses merubah formulir telepon dan format teks | Formulir telepon dan format teks berhasil diperbarui | Ya / Tidak |
| 2 | Menu pengaturan > perangkat, peran admin | Admin 2 | Pengujian proses merubah formulir telepon dan format teks | Formulir telepon dan format teks berhasil diperbarui | Ya / Tidak |
| 3 | Menu pengaturan > perangkat, peran pengelola | Pengelola 1 | Pengujian proses merubah formulir format teks | Formulir format teks berhasil diperbarui | Ya / Tidak |
| 4 | Menu pengaturan > perangkat, peran pengelola | Pengelola 2 | Pengujian proses merubah formulir format teks | Formulir format teks berhasil diperbarui | Ya / Tidak |
| 5 | Menu pengaturan > perangkat, peran pengelola | Pengelola 3 | Pengujian proses merubah formulir format teks | Formulir format teks berhasil diperbarui | Ya / Tidak |

Tabel 4. 10 Perancangan Pengujian Blackbox - Mengelola User

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------------|---------|--|--|--------------|
| 1 | Mengelola User | Admin 1 | Menguji proses menambah, dan menghapus data user | Data berhasil ditambahkan, dan dihapus | Ya / Tidak |
| 2 | Menu pengaturan > User | Admin 2 | Menguji proses menambah, dan menghapus data user | Data berhasil ditambahkan, dan dihapus | Ya / Tidak |

Tabel 4. 11 Perancangan Pengujian Blackbox - Mengedit Profile

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|-------------|--|---------------------------------------|--------------|
| 1 | Edit Profile | Admin 1 | Pengujian proses merubah informasi pribadi | Informasi pribadi berhasil diperbarui | Ya / Tidak |
| 2 | Edit Profile | Admin 2 | Pengujian proses merubah informasi pribadi | Informasi pribadi berhasil diperbarui | Ya / Tidak |
| 3 | Edit Profile | Pengelola 1 | Pengujian proses merubah informasi pribadi | Informasi pribadi berhasil diperbarui | Ya / Tidak |
| 4 | Edit Profile | Pengelola 2 | Pengujian proses merubah informasi pribadi | Informasi pribadi berhasil diperbarui | Ya / Tidak |
| 5 | Edit Profile | Pengelola 3 | Pengujian proses merubah informasi pribadi | Informasi pribadi berhasil diperbarui | Ya / Tidak |

Tabel 4. 12 Rancangan Pengujian Blackbox – Melihat Dashboard

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|-------------|--|---------------------------------|--------------|
| 1 | Dashboard | Admin 1 | Pengujian proses melihat informasi dashboard | Informasi dashboard ditampilkan | Ya / Tidak |
| 2 | Dashboard | Admin 2 | Pengujian proses melihat informasi dashboard | Informasi dashboard ditampilkan | Ya / Tidak |
| 3 | Dashboard | Pengelola 1 | Pengujian proses melihat informasi dashboard | Informasi dashboard ditampilkan | Ya / Tidak |
| 4 | Dashboard | Pengelola 2 | Pengujian proses melihat informasi dashboard | Informasi dashboard ditampilkan | Ya / Tidak |
| 5 | Dashboard | Pengelola 3 | Pengujian proses melihat informasi dashboard | Informasi dashboard ditampilkan | Ya / Tidak |

Tabel 4. 13 Perancangan Pengujian Blackbox – Pengiriman Pesan - Waktu Tepat

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|---------|--|-----------------------|--------------|
| 1 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu tepat | Admin 1 | Pesan dikirim dengan waktu yang telah ditentukan | Pesan terkirim | Ya / Tidak |
| 2 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu tepat | Admin 2 | Pesan dikirim dengan waktu yang telah ditentukan | Pesan terkirim | Ya / Tidak |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|-------------|--|-----------------------|--------------|
| 3 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu tepat | Pengelola 1 | Pesan dikirim dengan waktu yang telah ditentukan | Pesan terkirim | Ya / Tidak |
| 4 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu tepat | Pengelola 2 | Pesan dikirim dengan waktu yang telah ditentukan | Pesan terkirim | Ya / Tidak |
| 5 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu tepat | Pengelola 3 | Pesan dikirim dengan waktu yang telah ditentukan | Pesan terkirim | Ya / Tidak |

Tabel 4. 14 Perancangan Pengujian Blackbox - Pengiriman Pesan – Waktu Terlewati

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|---------|--|-----------------------|--------------|
| 1 | Mengirim pesan dengan jadwal aktif, tanggal yang tepat, dan waktu yang telah melewati | Admin 1 | Pesan dikirim dengan waktu yang telah melewati | Pesan terkirim | Ya / Tidak |
| 2 | Mengirim pesan dengan jadwal aktif, tanggal yang tepat, dan waktu yang telah melewati | Admin 2 | Pesan dikirim dengan waktu yang telah melewati | Pesan terkirim | Ya / Tidak |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|-------------|--|-----------------------|--------------|
| 3 | Mengirim pesan dengan jadwal aktif, tanggal yang tepat, dan waktu yang telah melewati | Pengelola 1 | Pesan dikirim dengan waktu yang telah melewati | Pesan terkirim | Ya / Tidak |
| 4 | Mengirim pesan dengan jadwal aktif, tanggal yang tepat, dan waktu yang telah melewati | Pengelola 2 | Pesan dikirim dengan waktu yang telah melewati | Pesan terkirim | Ya / Tidak |
| 5 | Mengirim pesan dengan jadwal aktif, tanggal yang tepat, dan waktu yang telah melewati | Pengelola 3 | Pesan dikirim dengan waktu yang telah melewati | Pesan terkirim | Ya / Tidak |

Tabel 4. 15 Perancangan Pengujian Blackbox – Pengiriman Pesan – Tanggal Terlewat

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|---------|--|-----------------------|--------------|
| 1 | Mengirim pesan dengan jadwal aktif, tanggal yang telah melewati, dan waktu yang tepat | Admin 1 | Tanggal telah melewati, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|-------------|--|-----------------------|--------------|
| 2 | Mengirim pesan dengan jadwal aktif, tanggal yang telah melewati, dan waktu yang tepat | Admin 2 | Tanggal telah melewati, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |
| 3 | Mengirim pesan dengan jadwal aktif, tanggal yang telah melewati, dan waktu yang tepat | Pengelola 1 | Tanggal telah melewati, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |
| 4 | Mengirim pesan dengan jadwal aktif, tanggal yang telah melewati, dan waktu yang tepat | Pengelola 2 | Tanggal telah melewati, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |
| 5 | Mengirim pesan dengan jadwal aktif, tanggal yang telah melewati, dan waktu yang tepat | Pengelola 3 | Tanggal telah melewati, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |

Tabel 4. 16 Perancangan Pengujian Blackbox – Pengiriman Pesan – Jadwal Tidak Aktif

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|-------------|--|-----------------------|--------------|
| 1 | Mengirim pesan dengan jadwal tidak aktif, tanggal dan waktu tepat | Admin 1 | Jadwal tidak aktif, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |
| 2 | Mengirim pesan dengan jadwal tidak aktif, tanggal dan waktu tepat | Admin 2 | Jadwal tidak aktif, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |
| 3 | Mengirim pesan dengan jadwal tidak aktif, tanggal dan waktu tepat | Pengelola 1 | Jadwal tidak aktif, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |
| 4 | Mengirim pesan dengan jadwal tidak aktif, tanggal dan waktu tepat | Pengelola 2 | Jadwal tidak aktif, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |
| 5 | Mengirim pesan dengan jadwal tidak aktif, tanggal dan waktu tepat | Pengelola 3 | Jadwal tidak aktif, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |

Tabel 4. 17 Perancangan Pengujian Blackbox - Pengiriman Pesan – Tanggal dan Waktu Salah

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|--|-------------|---|-----------------------|--------------|
| 1 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu yang salah | Admin 1 | Tanggal dan waktu salah, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |
| 2 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu yang salah | Admin 2 | Tanggal dan waktu salah, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |
| 3 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu yang salah | Pengelola 1 | Tanggal dan waktu salah, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |
| 4 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu yang salah | Pengelola 2 | Tanggal dan waktu salah, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |
| 5 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu yang salah | Pengelola 3 | Tanggal dan waktu salah, pesan tidak terkirim | Pesan tidak terkirim | Ya / Tidak |

2. Usability Testing

Usability testing merupakan metode pengujian yang mengevaluasi pengalaman *user* dalam memahami dan menggunakan teknologi, aplikasi, atau website tertentu. Pengguna tersebut melibatkan peran admin dan pengelola. tujuan dari pengujian ini adalah untuk mengevaluasi sejauh mana aplikasi penagihan iuran rumah kost telah memenuhi spesifikasi dan persyaratan yang telah ditentukan.

Tabel 4. 18 Perancangan Pengujian Usability Testing - Login

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|------------------|-------------|--|---|------------------------|
| 1 | Login | Admin 1 | Menguji proses login dengan data benar | User berhasil login dan melihat dashboard | Lulus / Tidak Lulus |
| 2 | Login | Admin 2 | Menguji proses login dengan data benar | User berhasil login dan melihat dashboard | Lulus / Tidak Lulus |
| 3 | Login | Pengelola 1 | Menguji proses login dengan data benar | User berhasil login dan melihat dashboard | Lulus / Tidak Lulus |
| 4 | Login | Pengelola 2 | Menguji proses login dengan data benar | User berhasil login dan melihat dashboard | Lulus / Tidak Lulus |
| 5 | Login | Pengelola 3 | Menguji proses login dengan data benar | User berhasil login dan melihat dashboard | Lulus / Tidak Lulus |

Tabel 4. 19 Perancangan Pengujian Usability Testing - Mengakses Dashboard

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|---------------------|---------|------------------------------------|--------------------------------|------------------------|
| 1 | Mengakses Dashboard | Admin 1 | Menguji akses ke halaman dashboard | Dashboard berhasil ditampilkan | Lulus / Tidak Lulus |
| 2 | Mengakses Dashboard | Admin 2 | Menguji akses ke halaman dashboard | Dashboard berhasil ditampilkan | Lulus / Tidak Lulus |

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|---------------------|-------------|------------------------------------|--------------------------------|------------------------|
| 3 | Mengakses Dashboard | Pengelola 1 | Menguji akses ke halaman dashboard | Dashboard berhasil ditampilkan | Lulus / Tidak Lulus |
| 4 | Mengakses Dashboard | Pengelola 2 | Menguji akses ke halaman dashboard | Dashboard berhasil ditampilkan | Lulus / Tidak Lulus |
| 5 | Mengakses Dashboard | Pengelola 3 | Menguji akses ke halaman dashboard | Dashboard berhasil ditampilkan | Lulus / Tidak Lulus |

Tabel 4. 20 Perancangan Pengujian Usability Testing - Mengelola Kost

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|------------------|-------------|---|---|------------------------|
| 1 | Mengelola Kost | Admin 1 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |
| 2 | Mengelola Kost | Admin 2 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |
| 3 | Mengelola Kost | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|------------------|-------------|---|---|------------------------|
| 4 | Mengelola Kost | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |
| 5 | Mengelola Kost | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |

Tabel 4. 21 Perancangan Pengujian Usability Testing - Mengelola Kontak

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|------------------|-------------|---|---|------------------------|
| 1 | Mengelola Kontak | Admin 1 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |
| 2 | Mengelola Kontak | Admin 2 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |
| 3 | Mengelola Kontak | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|------------------|-------------|---|---|------------------------|
| 4 | Mengelola Kontak | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |
| 5 | Mengelola Kontak | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |

Tabel 4. 22 Perancangan Pengujian Usability Testing - Mengelola Laporan

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|-------------------|-------------|--|---|------------------------|
| 1 | Mengelola Laporan | Admin 1 | Menguji proses menambah, mengubah, dan menghapus laporan | Laporan berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |
| 2 | Mengelola Laporan | Admin 2 | Menguji proses menambah, mengubah, dan menghapus laporan | Laporan berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |
| 3 | Mengelola Laporan | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus laporan | Laporan berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|-------------------|-------------|--|---|------------------------|
| 4 | Mengelola Laporan | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus laporan | Laporan berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |
| 5 | Mengelola Laporan | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus laporan | Laporan berhasil ditambahkan, diubah, dihapus | Lulus / Tidak Lulus |

Tabel 4. 23 Perancangan Pengujian Usability Testing - Mengelola Perangkat

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|---------------------|-------------|---|---|------------------------|
| 1 | Mengelola perangkat | Admin 1 | Menguji proses merubah formulir telepon dan format teks | Formulir telepon dan teks berhasil diubah | Lulus / Tidak Lulus |
| 2 | Mengelola perangkat | Admin 2 | Menguji proses merubah formulir telepon dan format teks | Formulir telepon dan teks berhasil diubah | Lulus / Tidak Lulus |
| 3 | Mengelola perangkat | Pengelola 1 | Menguji proses merubah formulir format teks | Formulir teks berhasil diubah | Lulus / Tidak Lulus |
| 4 | Mengelola perangkat | Pengelola 2 | Menguji proses merubah formulir format teks | Formulir teks berhasil diubah | Lulus / Tidak Lulus |
| 5 | Mengelola perangkat | Pengelola 3 | Menguji proses merubah formulir format teks | Formulir teks berhasil diubah | Lulus / Tidak Lulus |

Tabel 4. 24 Perancangan Pengujian Usability Testing - Mengubah Profile

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|-------------------------------|-------------|--|-----------------------------------|------------------------|
| 1 | Melakukan <i>Edit Profile</i> | Admin 1 | Menguji proses merubah informasi pribadi | Informasi pribadi berhasil diubah | Lulus / Tidak Lulus |
| 2 | Melakukan <i>Edit Profile</i> | Admin 2 | Menguji proses merubah informasi pribadi | Informasi pribadi berhasil diubah | Lulus / Tidak Lulus |
| 3 | Melakukan <i>Edit Profile</i> | Pengelola 1 | Menguji proses merubah informasi pribadi | Informasi pribadi berhasil diubah | Lulus / Tidak Lulus |
| 4 | Melakukan <i>Edit Profile</i> | Pengelola 2 | Menguji proses merubah informasi pribadi | Informasi pribadi berhasil diubah | Lulus / Tidak Lulus |
| 5 | Melakukan <i>Edit Profile</i> | Pengelola 3 | Menguji proses merubah informasi pribadi | Informasi pribadi berhasil diubah | Lulus / Tidak Lulus |

Tabel 4. 25 Perancangan Pengujian Usability Testing - Mengelola User

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|------------------|---------|--|------------------------------------|------------------------|
| 1 | Mengelola user | Admin 1 | Menguji proses menambah dan menghapus user | User berhasil ditambahkan, dihapus | Lulus / Tidak Lulus |
| 2 | Mengelola user | Admin 2 | Menguji proses menambah dan menghapus user | User berhasil ditambahkan, dihapus | Lulus / Tidak Lulus |

Tabel 4. 26 Perancangan Pengujian Usability Testing – Melihat Status Pengiriman Pesan

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|-----------------------------------|-------------|---|--|------------------------|
| 1 | Memeriksa status pengiriman pesan | Admin 1 | Menguji pemeriksaan status pengiriman pesan di menu laporan | Status pengiriman pesan berhasil diperiksa | Lulus / Tidak Lulus |
| 2 | Memeriksa status pengiriman pesan | Admin 2 | Menguji pemeriksaan status pengiriman pesan di menu laporan | Status pengiriman pesan berhasil diperiksa | Lulus / Tidak Lulus |
| 3 | Memeriksa status pengiriman pesan | Pengelola 1 | Menguji pemeriksaan status pengiriman pesan di menu laporan | Status pengiriman pesan berhasil diperiksa | Lulus / Tidak Lulus |
| 4 | Memeriksa status pengiriman pesan | Pengelola 2 | Menguji pemeriksaan status pengiriman pesan di menu laporan | Status pengiriman pesan berhasil diperiksa | Lulus / Tidak Lulus |
| 5 | Memeriksa status pengiriman pesan | Pengelola 3 | Menguji pemeriksaan status pengiriman pesan di menu laporan | Status pengiriman pesan berhasil diperiksa | Lulus / Tidak Lulus |

4.3. Implementasi Sistem

Pada tahap implementasi akan dijelaskan langkah-langkah dalam pembuatan perancangan sistem aplikasi penagihan iuran rumah kost.

4.3.1. Memasang dan Menjalankan *Framework Laravel*

Pada tahap ini, peneliti akan melakukan serangkaian langkah dalam memasang dan menjalankan *framework laravel*. Berikut ini adalah langkah-langkahnya:

4.3.1.1. Memasang *Composer*

Pada penelitian ini diperlukan melakukan tahapan memasang paket *composer* agar *framework laravel* dapat berjalan didalam laptop. Pada langkah ini dapat dilihat pada lampiran A1.

4.3.1.2. Memasang *Laragon*

Langkah selanjutnya dalam penelitian ini, peneliti melakukan *installasi laragon* yang nantinya akan digunakan sebagai *server localhost* dan *database* pada saat pengembangan proyek aplikasi rumah kost. Petunjuk tentang cara menjalankan *laragon* dapat ditemukan di lampiran A2.

Setelah menyelesaikan langkah-langkah diatas, buka *command prompt* pada *windows* dan jalankan perintah berikut:

```
C:\laragon\www\composer create-project --prefer-dist laravel/laravel:^11.0 rumah_kost
```

4.3.1.3. Menjalankan *Project laravel*

Setelah berhasil melakukan instalasi, langkah selanjutnya adalah menjalankan *framework laravel*, silahkan ketikkan perintah berikut di dalam *terminal*:

```
C:\laragon\www\ cd rumah_kost
```

Perintah ini digunakan untuk masuk ke dalam direktori proyek rumah kost. Setelah berhasil masuk ke dalam direktori tersebut, jalankan proyek dengan perintah berikut:

```
php artisan serve
```

4.3.1.4. Konfigurasi *Database*

Framework laravel menyediakan sebuah file bernama *.env*, file tersebut digunakan untuk mengonfigurasi koneksi ke *database*. Berikut ini adalah sintaks untuk mengonfigurasi koneksi database tersebut.

```
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
```

Dalam hal ini, diperlukan penyesuaian terhadap *database* yang peneliti miliki menjadi seperti berikut ini:

```
DB_DATABASE=rumah_kost
DB_USERNAME=root
DB_PASSWORD=
```

4.3.2. Membuat *Authentication Laravel*

Langkah-langkah dalam membuat *authentication* dalam *laravel* pada penelitian ini memanfaatkan paket dari *laravel breeze*, langkah-langkah implementasi dan penjelasan dari paket *laravel breeze* ini dapat dilihat pada lampiran A3. Setelah berhasil melakukan instalasi paket *laravel breeze*, diperlukan mengubah kode pada file *resource/view/welcome.blade.php* dan *resource/view/auth/login.blade.php* dengan mengubahnya menjadi seperti berikut ini:

```
<div class="login-dark" style="height: 695px;">
  <form method="POST" action="{{ route('login') }}">
    @csrf

    <p class="kost text-left font-semibold text-gray-800 dark:text-gray-200 leading-tight">Selamat Datang di</p>
    <h4 class="rumah text-center font-semibold text-gray-800 dark:text-gray-200 leading-tight">Rumah
      <span class="kost">Kost</span>
    </h4>
    <h2 class="sr-only">Login Form</h2>
    <div class="illustration"><i class="icon ion-ios-locked-outline"></i></div>
    <div class="form-group">
      <input class="form-control block mt-1 w-full" type="email" name="email" :value="old('email')"
        required autofocus
        autocomplete="username" placeholder="Email">
      <x-input-error :messages="$errors->get('email')" class="mt-2" />
    </div>
  </div>
```

```

        <div class="form-group">
            <input id="password" class="form-control
block mt-1 w-full" type="password" name="password" required
                autocomplete="current-password"
placeholder="password">
            </div>
            <div class="form-group">
                <button class="btn btn-primary btn-block"
type="submit">Log In</button>
            </div>
            @if (Route::has('password.request'))
                <a class="underline text-sm forgot text-
gray-600 dark:text-gray-400 hover:text-gray-900 dark: hover:text-
gray-100 rounded-md focus:outline-none focus:ring-2 focus:ring-
offset-2 focus:ring-indigo-500 dark:focus:ring-offset-gray-800"
                href="{{ route('password.request') }}">
                    {{ __('Forgot your password?') }}
                </a>
            @endif
        </form>
    </div>

```

Untuk melihat kode lengkap pada kode di atas, dapat dilihat di dalam kode sumber yang terdapat pada lampiran A6.

4.3.3. Melakukan Pembuatan *Menu* Jadwal

Pada tahap ini peneliti melakukan pembuatan untuk *menu* jadwal agar dapat mengelola *menu* tersebut, seperti menambah, dan mengubah data. Berikut ini adalah implementasinya:

4.3.3.1. Membuat *Model* Jadwal

Model jadwal diperlukan untuk dapat membuat data jadwal yang akan disimpan ke dalam *database*. Berikut ini adalah langkah-langkahnya.

Buka *terminal* lalu berikan perintah ini

```
php artisan make:model Jadwal -m
```

Pada langkah selanjutnya buka *file migration* yang berada di *folder database/migrations/2024_05_04_102624_create_jadwals_table.php*, pada *function up*, diperlukan merubah *code*-nya menjadi seperti ini:

```

public function up(): void
{
    Schema::create('jadwals', function (Blueprint $table) {
        $table->id();
    });
}

```

```

        $table->foreignIdFor(Kontak::class)->constrained()-
->cascadeOnDelete();
        $table->date('tanggal_kirim');
        $table->time('waktu_kirim');
        $table->boolean('status')->default(true);
        $table->timestamps();
    });
}

```

Langkah selanjutnya buka file *app/models/Jadwal.php*. Dalam hal ini diperlukan untuk merubah *code*-nya menjadi seperti ini.

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\HasOne;

class Jadwal extends Model
{
    use HasFactory;

    protected $guarded = [];

    protected static function boot()
    {
        parent::boot();

        static::creating(function ($jadwal) {
            $jadwal->status = 0; // Atur nilai status ke 0 saat
            pembuatan jadwal baru
        });
    }

    public function kontak()
    {
        return $this->belongsTo(Kontak::class);
    }

    public function laporan()
    {
        return $this->hasOne(Laporan::class);
    }
}

```

4.3.3.2. Membuat *Controller* Jadwal

Langkah selanjutnya adalah membuat *controller* dari *model* jadwal, buka *terminal* lalu ketikkan perintah ini.

```
php artisan make:controller JadwalController
```

Pada langkah selanjutnya buka *file controller* yang berada di *folder app/http/controller/JadwalController.php*, didalam *file JadwalController.php* diperlukan merubah *code-nya* menjadi berikut

```
<?php

namespace App\Http\Controllers;

use App\Models\Jadwal;
use App\Models\Kontak;
use Illuminate\Http\Request;

class JadwalController extends Controller
{
    public function index()
    {
        $title = "Atur Jadwal Pesan";
        $jadwals = Jadwal::with('kontak')->get();
        $kontak = Kontak::with('jadwal')->get();
        // dd($jadwals);
        return view('dashboard.pengaturan.jadwal.index', [
            'title' => $title,
            'jadwals' => $jadwals,
            'kontak' => $kontak
        ]);
    }
    public function store(Request $request)
    {
        try {
            $validatedData = request()->validate([
                'kontak_id' => 'required',
                'status' => 'required',
                'tanggal_kirim' => 'required',
                'waktu_kirim' => 'required',
            ]);

            $jadwal = Jadwal::updateOrCreate(
                ['kontak_id' => $validatedData['kontak_id']],
                $validatedData
            );
            $jadwal->update();
            toastr()->success('Data berhasil disimpan!');
            return redirect()->route('jadwal.index')-
            >with('Success', 'Data berhasil Diperbarui');
        } catch (\Throwable $th) {
            toastr()->error('Data gagal disimpan!');
            return response()->json([
                'status' => 'error',
                'message' => 'Message failed to send',
                'error' => $th->getMessage()
            ], 500);
        }
    }
}
```

Penulisan pada kode diatas adalah hasil dari ringkasan penelitian peneliti, untuk dapat melihat kode lengkapnya dapat dilihat pada lampiran A6.

4.3.4. Membuat *Menu Laporan Pengiriman Pesan dan Pembayaran*

Pada tahap ini laporan daftar dari laporan pembayaran dan laporan pengiriman pesan digabungkan menjadi satu di dalam *file index.blade.php*, namun dalam membuat, mengubah, dan menghapus laporan pembayaran berbeda penempatan dari *menu* laporan pengiriman pesan terhadap *file model, controller, create.blade.php* dan *edit.blade.php*. Berikut ini adalah implementasinya:

4.3.4.1. Membuat *Model Pengiriman Pesan*

Model pengiriman pesan untuk selanjutnya disebut *model* laporan, diperlukan untuk dapat memanipulasi data pengiriman pesan yang akan disimpan ke dalam database. Berikut ini adalah langkah-langkahnya.

Buka terminal lalu berikan perintah ini :

```
php artisan make:model Laporan -m
```

Pada langkah selanjutnya buka file migration yang berada di *folder database/migrations/2024_05_14_022831_create_laporans_table.php*, pada *function up*, diperlukan merubah code-nya menjadi seperti ini:

```
public function up(): void
{
    Schema::create('laporans', function (Blueprint $table) {
        $table->id();
        $table->foreignIdFor(Jadwal::class)->constrained();
        $table->date('tanggal_terkirim');
        $table->string('status');
        $table->timestamps();
    });
}
```

Langkah selanjutnya buka file *app/models/Laporan.php*. Dalam hal ini diperlukan untuk merubah code-nya menjadi seperti ini.

```
<?php
namespace App\Models;
```

```

use App\Models\Jadwal;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;

class Laporan extends Model
{
    use HasFactory;
    protected $guarded = [];
    public function jadwal()
    {
        return $this->belongsTo(Jadwal::class);
    }
}

```

4.3.4.2. Membuat *Model* Pembayaran

Model pembayaran diperlukan untuk dapat memanipulasi data laporan pembayaran penghuni kost yang dapat disimpan ke dalam *database*. Berikut ini adalah langkah-langkahnya.

Buka *terminal* lalu tuliskan perintah seperti dibawah ini

```
php artisan make:model Payment -m
```

Pada langkah selanjutnya buka *file migration* yang berada di *folder database/migrations/2024_05_15_145951_create_payments_table.php*, pada *function up*, diperlukan merubah *code*-nya menjadi seperti ini:

```

public function up(): void
{
    Schema::create('payments', function (Blueprint $table) {
        $table->id();
        $table->foreignIdFor(Kontak::class)->constrained();
        $table->date('tanggal_pembayaran');
        $table->float('nominal');
        $table->timestamps();
    });
}

```

Langkah selanjutnya buka *file app/models/Payment.php*. Dalam hal ini diperlukan untuk merubah *code*-nya menjadi seperti ini.

```

<?php

namespace App\Models;

use App\Models\Kontak;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Factories\HasFactory;

```

```

class Payment extends Model
{
    use HasFactory;

    protected $guarded = [];

    public function kontak()
    {
        return $this->belongsTo(Kontak::class);
    }
}

```

4.3.4.3. Membuat *Controller* Laporan

Langkah selanjutnya adalah membuat *controller* dari model kost, buka terminal lalu ketikkan perintah ini.

```
php artisan make:controller LaporanController
```

Pada langkah selanjutnya buka file controller yang berada di folder *app/http/controller/LaporanController.php*, didalam file *LaporanController.php* diperlukan merubah *code*-nya menjadi berikut

```

<?php
namespace App\Http\Controllers;

use App\Models\Jadwal;
use App\Models\Laporan;
use App\Models\Payment;
use App\Models\Setting;
use App\Traits\WatsappTrait;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Date;

class LaporanController extends Controller
{
    use WatsappTrait;
    public function index()
    {
        $title = "Laporan";

        $laporans = Laporan::with(['jadwal'])->orderBy('id',
'desc')->paginate(5);
        $payments = Payment::with(['kontak']->orderBy('tanggal_pembayaran', 'desc')->paginate(5);
        return view('dashboard.laporan.index', [
            'title' => $title,
            'laporans' => $laporans,
            'payments' => $payments
        ]);
    }
}

```

```

}
public function update(Request $request, Laporan $laporan)
{
    try {
        $phone = $laporan->jadwal->kontak->no_telpon;
        $db_setting = Setting::first();
        $message = $db_setting->format_text;
        $payload = [
            'phone' => $phone,
            'message' => $message
        ];
        $laporan->update([
            'tanggal_terkirim' => Date::now(),
            'status' => 'terkirim'
        ]);
        try {
            $this->sendMessage($payload);
        } catch (\Throwable $th) {
            $this->sendTextWatsapp($phone, $message);
        }
        toastr()->success('Data berhasil dikirim ulang!');
        return redirect()->route('laporan.index')-
>with('Success', 'Data berhasil dikirim ulang');
    } catch (\Throwable $th) {
        $laporan->update([
            'tanggal_terkirim' => Date::now(),
            'status' => 'pending'
        ]);
        toastr()->error('Data gagal dikirim ulang!');
        return redirect()->route('laporan.index')-
>with('Error', 'Data gagal dikirim ulang'); //throw $th;
    }
}
}

```

4.3.4.4. Membuat *Controller* Laporan Pembayaran

Langkah selanjutnya adalah membuat *controller* dari model Laporan, buka terminal lalu ketikkan perintah ini.

```
php artisan make:controller PaymentController
```

Pada langkah selanjutnya buka file controller yang berada di folder `app/http/controller/PaymentController.php`, didalam file `PaymentController.php` diperlukan merubah *code*-nya menjadi berikut:

```

<?php

namespace App\Http\Controllers;

use App\Models\Kontak;
use App\Models\Payment;

```

```

use Illuminate\Http\Request;

class PaymentController extends Controller
{
    public function create()
    {
        $title = "Tambah Pembayaran";
        $kontak = Kontak::cursor();
        return view('dashboard.payment.create', [
            'title' => $title,
            'kontak' => $kontak
        ]);
    }
    public function edit(Payment $payment)
    {
        $title = "Edit Pembayaran";
        $kontak = Kontak::cursor();
        return view('dashboard.payment.edit', [
            'title' => $title,
            'kontak' => $kontak,
            'payment' => $payment
        ]);
    }
}

```

Penulisan pada kode diatas adalah hasil dari ringkasan penelitian peneliti, untuk dapat melihat kode lengkapnya dapat dilihat pada lampiran A6.

4.3.5. Membuat *Routing*

Tahap ini berfungsi menangani semua permintaan alamat yang nantinya diperlukan untuk pengiriman data atau informasi kepada pengguna, sebelum melakukan hal tersebut diperlukan mendaftarkan routing agar semua permintaan dapat diproses oleh aplikasi. Diperlukan mengubah kode pada *file route/web.php* menjadi berikut ini:

```

<?php

use App\Http\Controllers\API\DashboardController as
APIDashboardController;
use App\Http\Controllers\DashboardController;
use App\Http\Controllers\JadwalController;
use App\Http\Controllers\KontakController;
use App\Http\Controllers\KostController;
use App\Http\Controllers\LaporanController;
use App\Http\Controllers\PaymentController;
use App\Http\Controllers\ProfileController;
use App\Http\Controllers\SettingController;
use App\Http\Controllers\TestController;
use App\Http\Controllers\UserController;
use Illuminate\Support\Facades\Route;

Route::get('/', function () {

```

```

        return view('auth.login');
    });

Route::get('/dashboard', [DashboardController::class, 'index'])
    ->middleware(['auth', 'verified'])
    ->name('dashboard');

Route::middleware('auth')->group(function () {
    Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');
    Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');
    Route::delete('/profile', [ProfileController::class, 'destroy'])->name('profile.destroy');
    Route::resource('/perangkat', SettingController::class);
    Route::resource('/kontak', KontakController::class);
    Route::resource('/jadwal', JadwalController::class);
    Route::resource('/laporan', LaporanController::class);
    Route::resource('/payment', PaymentController::class);
    Route::resource('/kost', KostController::class);
    Route::get('/api/kontak/{id}', 'App\Http\Controllers\KontakController@getJSONKontak')->name('api.kontak');
    Route::get('api/dashboard', [APIDashboardController::class, 'getDashboardData']);
});

Route::middleware(['AdminMiddleware'])->group(function () {
    Route::resource('/user', UserController::class)->middleware('auth', 'admin');
});

route::resource('/test', TestController::class);

require __DIR__ . '/auth.php';

```

4.3.6. Membuat *AdminMiddleware*

Tahap ini berfungsi untuk memisahkan peran admin dan pengelola, untuk cara pembuatannya adalah sebagai berikut:

Buka terminal dan tuliskan perintah berikut

```
php artisan make:middleware AdminMiddleware
```

Perintah tersebut akan otomatis membuat *file AdminMiddleware*, dan *file* tersebut terletak pada *app/Http/Middleware/AdminMiddleware.php*. peneliti merubah kode yang didalam *file AdminMiddleware.php* menjadi sebagai berikut:

```
<?php
```

```

namespace App\Http\Middleware;

use Closure;
use App\Models\User;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;

class AdminMiddleware
{
    public function handle(Request $request, Closure $next):
Response
    {
        if ($request->user()->role == User::ROLE_ADMIN) {
            return $next($request);
        }

        abort(401);
    }
}

```

Selanjutnya didalam *file app/Models/User.php* peneliti menambah kode berikut ini dengan ditempatkan dibawah `use HasFactory, Notifiable;`

```

const ROLE_PENGELOLA = 'pengelola';
const ROLE_ADMIN = 'admin';

```

pada tahap selanjutnya peneliti mendaftarkan *AdminMiddleware* untuk dapat digunakan fungsinya. Buka file *bootstrap/app.php* didalam *script ->withMiddleware(function(Middleware \$middleware){}* diperlukan merubah menjadi seperti kode berikut ini

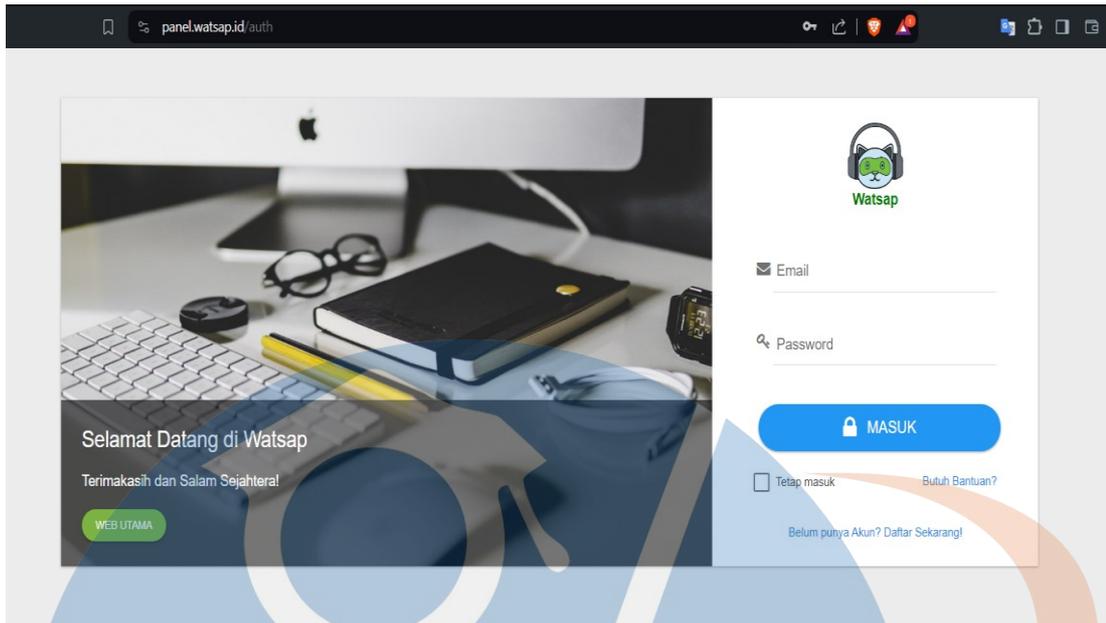
```

->withMiddleware(function (Middleware $middleware) {
$middleware->alias([
AdminMiddleware::class,
CorsService::class,
],);
'AdminMiddleware' =>
'cors' =>
});

```

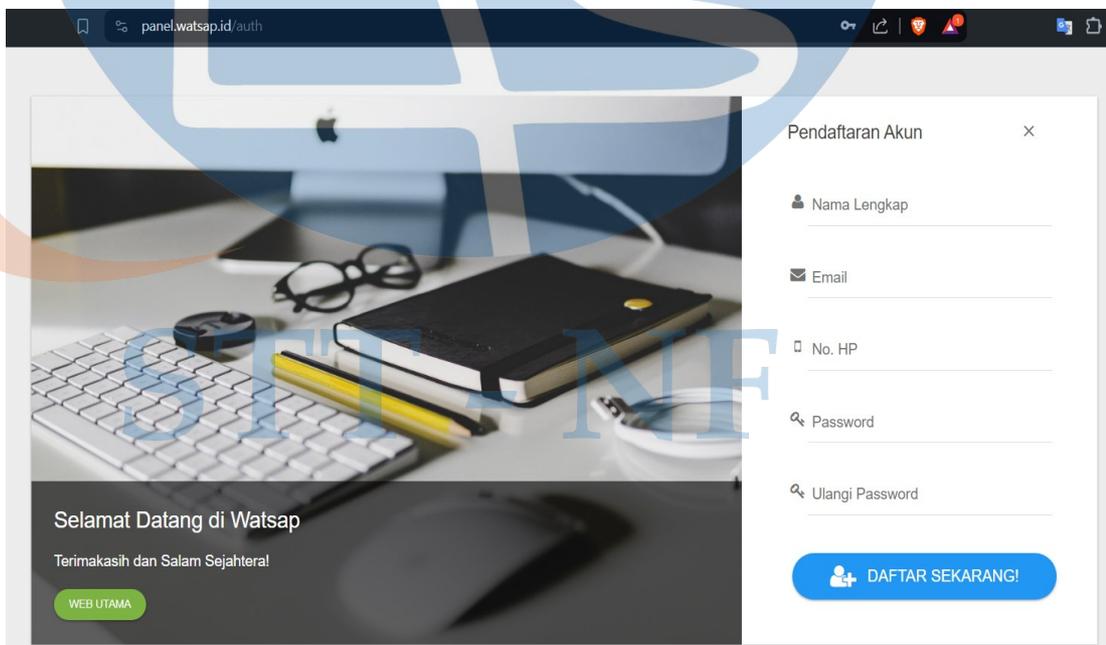
4.3.7. Mendapatkan *Key API Whatsapp*

Pada tahap ini, untuk mendapatkan *key API whatsapp* peneliti memanfaatkan layanan pihak ketiga yaitu *Watsap.id*. untuk dapat mengakses pihak ketiga diperlukan memiliki *account* dari pihak ketiga. Pada langkah pertama yaitu masuk ke *website panel.watsap.id/auth*



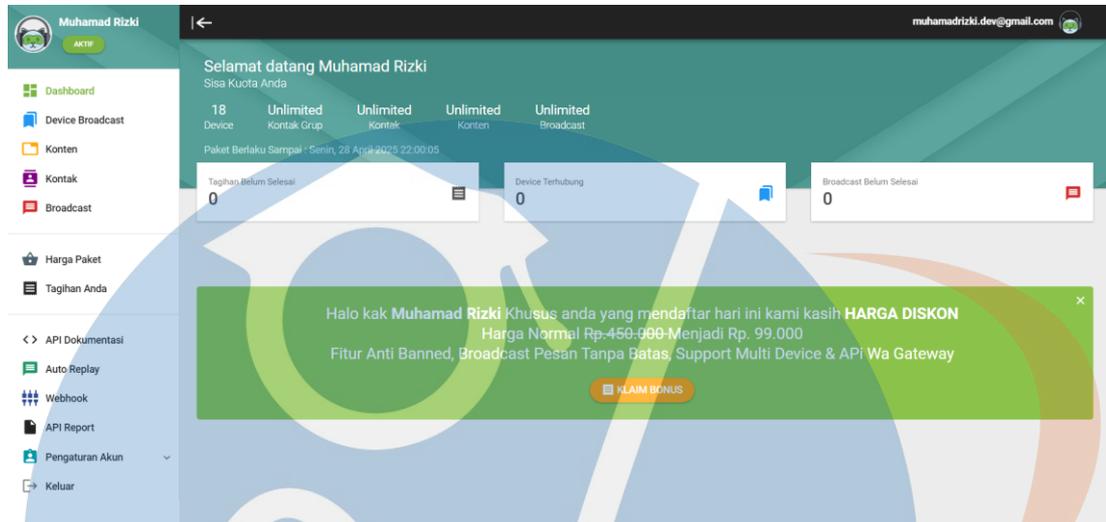
Gambar 4. 16 Tampilan Halaman - Login Watsap.id

Gambar 4. 16 adalah halaman tampilan awal aplikasi untuk mengakses ke dalam halaman *dashboard* aplikasi, dalam halaman ini diperlukan memasukkan email dan password yang telah terdaftar. Jika belum mempunyai akun, pengguna diperlukan melakukan pendaftaran. Halaman pendaftaran dapat diakses jika pengguna mengklik [belum punya akun? Daftar Sekarang!](#).



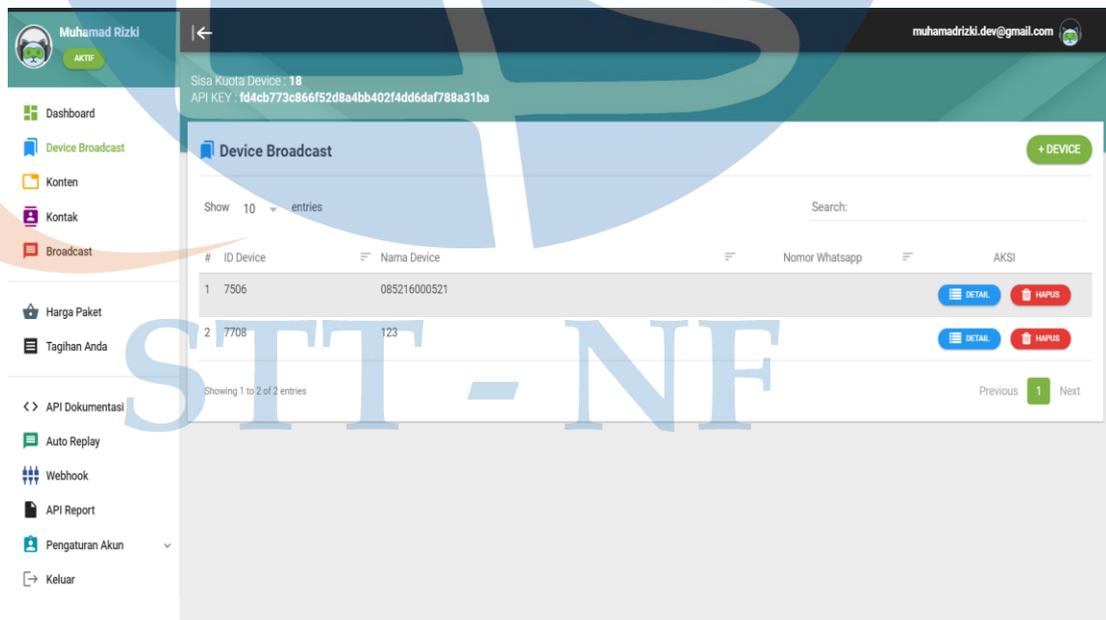
Gambar 4. 17 Tampilan Halaman - Register Watsap.id

Gambar 4. 17 adalah tampilan antarmuka pada halaman daftar pengguna, terdapat formulir yang perlu dilengkapi. Setelah pengguna berhasil mendaftar maka selanjutnya pengguna akan diarahkan ke halaman login pengguna.



Gambar 4. 18 Tampilan Halaman - Dashboard Watsap.id

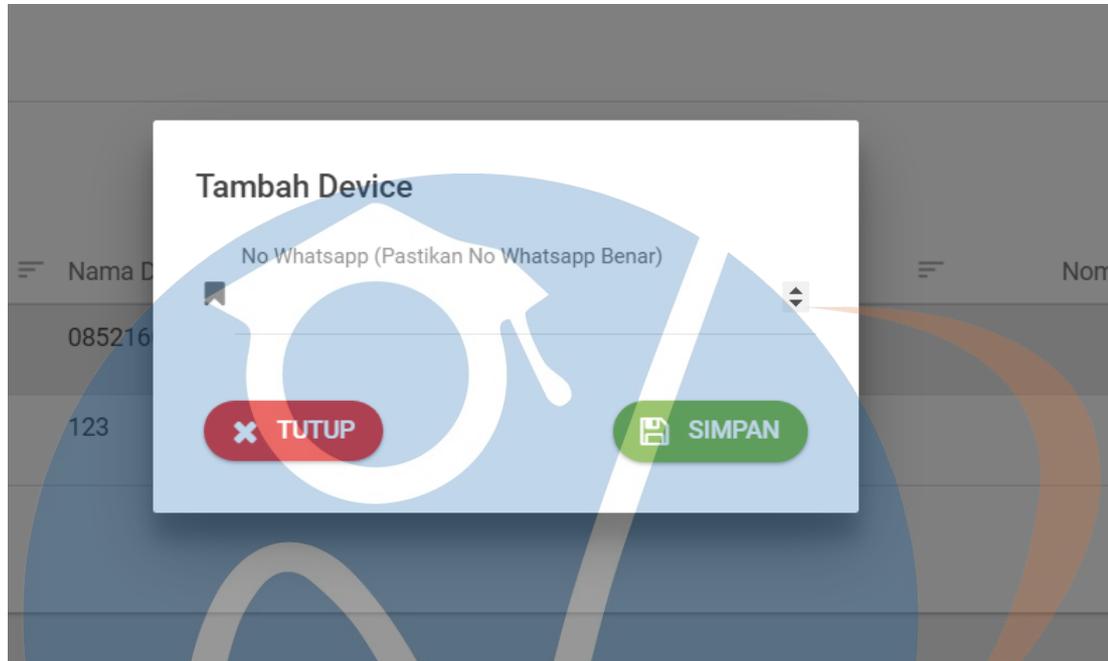
Gambar 4. 18 adalah tampilan awal ketika *user* berhasil melakukan *login*, setelah berhasil maka *user* akan diarahkan ke dalam *menu dashboard*. Pada halaman ini, *user* dapat melihat informasi dari akun *user* pribadi.



Gambar 4. 19 Tampilan Halaman - Device Watsap.id

Gambar 4.23 adalah halaman antarmuka dari *device broadcast* fungsinya adalah mendaftarkan nomor telepon user yang nantinya akan menjadi nomor

telepon pengirim. Didalam *menu* ini terdapat *api key* yang nantinya akan dapat menjembatani aplikasi yang dibuat peneliti dan aplikasi *whatsapp*.



Gambar 4. 20 Tampilan Halaman - Tambah Device Watsap.id

Gambar 4.24 adalah tampilan antarmuka dari penambahan nomor telepon pengguna, pengguna dapat melakukan penambahan nomor telepon dengan memasukkan nomor telepon pengguna setelah itu klik tombol simpan.

4.3.8. Memasang *API Key* Pada *Laravel*

Dalam tahap ini, didalam *file .env* dibaris terbawah tambahkan kode ini ke dalam *file .env*:

```
API_TOKEN_WATSAPP=url_api_key  
DOMAIN_SERVER_WATSAPP=https://wa.srv34.wapanel.s.com
```

4.3.9. Membuat *Traits Laravel*

Pada tahap ini, peneliti memanfaatkan salah satu fitur *laravel* yaitu *traits*. Penjelasan singkatnya adalah, *traits* ini digunakan untuk melakukan pemanggilan fungsi yang berulang-ulang sehingga peneliti tidak perlu melakukan penempatan kode yang berulang-ulang, untuk penjelasan lebih jelasnya dapat dilihat pada lampiran A4. Langkahnya adalah sebagai berikut:

buka terminal lalu tuliskan perintah ini

```
composer require mtownsend/laravel-make-trait
```

Perintah ini adalah untuk menambahkan paket *traits* kedalam aplikasi *laravel*. Selanjutnya tuliskan perintah ini, untuk membuat folder *Traits*

```
php artisan make:trait WhatsappTrait
```

Pada langkah selanjutnya buka *file WhatsappTrait.php* yang berada di folder *app/Traits/WhatsappTrait.php*, didalam *file* tersebut diperlukan merubah *code*-nya menjadi seperti ini:

```
<?php
namespace App\Traits;
use App\Models\Setting;
trait WhatsappTrait
{
    public function sendTextWhatsapp($phone, $message)
    {
        $sender = Setting::latest()->value('no_telpon');
        $sender = preg_replace('/[^0-9]/', '', $sender);
        $sender = (str_starts_with($sender, '0')) ? '62' .
substr($sender, 1) : $sender;

        $phone = preg_replace('/[^0-9]/', '', $phone);
        $phone = (str_starts_with($phone, '0')) ? '62' .
substr($phone, 1) : $phone;

        $token = env('API_TOKEN_WATSAPP');
        $payload = [
            'api_key' => $token,
            'sender' => $sender,
            'number' => $phone,
            'message' => $message
        ];

        $curl = curl_init();
        curl_setopt_array($curl, array(
            CURLOPT_URL => env('DOMAIN_SERVER_WATSAPP') .
"/send-message",
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_ENCODING => '',
            CURLOPT_MAXREDIRS => 10,
            CURLOPT_TIMEOUT => 0,
```

```

        CURLOPT_FOLLOWLOCATION => true,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => 'POST',
        CURLOPT_POSTFIELDS => json_encode($payload),
        CURLOPT_HTTPHEADER => array(
            'Content-Type: application/json'
        ),
    )); // tambahkan kurung tutup di sini

    $response = curl_exec($curl);
    curl_close($curl);

    return $response; // tambahkan ini untuk mengembalikan
    respons dari curl
}
}

```

Penulisan pada kode diatas adalah hasil dari ringkasan penelitian peneliti, untuk dapat melihat kode lengkapnya dapat dilihat pada lampiran A6.

4.3.10. Membuat *Task* Pengiriman Pesan

Dalam pembuatan pengiriman pesan secara terjadwal peneliti memanfaatkan paket *laravel* yang bernama *cronjob*, penjelasan singkatnya adalah *cronjob* dapat menjalankan suatu perintah secara otomatis dalam bentuk *script*, pengembang dapat menjalankan *script* yang akan dijalankan dan kapan perintah dijalankan. Untuk penjelasan lengkapnya dapat dilihat dalam lampiran A5. Langkah pertama yaitu buka terminal dan tuliskan perintah dibawah:

```
php artisan make:command Tagihan --command=tagihan:cron
```

Setelah perintah diatas dijalankan maka akan otomatis membuat folder *commands* dan file *tagihan.php*, file tersebut teletak pada *app/Console/Commands/Tagihan.php* dan kemudian diperlukan mengubah menjadi kode berikut

```

<?php

namespace App\Console\Commands;

use Carbon\Carbon;
use App\Models\Jadwal;
use App\Models\Laporan;
use App\Models\Setting;
use App\Traits\WatsappTrait;
use Illuminate\Console\Command;

class Tagihan extends Command
{

```

```

use WatsappTrait;
protected $signature = 'tagihan:cron';

protected $description = 'Mengirim Tagihan';

public function __construct()
{
    parent::__construct();
}

public function handle()
{
    try {
        $jadwalAktif = Jadwal::where('status', 1)-
>with('kontak')->get();

        foreach ($jadwalAktif as $jadwal) {
            $tanggal_jadwal = Carbon::parse($jadwal-
>tanggal_kirim);
            // $waktu_jadwal = $jadwal->>waktu_kirim-
>format('H:i:s');
            $waktu_jadwal = Carbon::parse($jadwal-
>>waktu_kirim)->format('H:i:s');

            $tanggal_hari_ini = Carbon::now();
            $waktu_hari_ini = $tanggal_hari_ini-
>format('H:i:s');

            if ($tanggal_jadwal-
>isSameDay($tanggal_hari_ini) && $waktu_jadwal <
$waktu_hari_ini) {
                $db_setting = Setting::first();
                $message = $db_setting->format_text;
                $phone = $jadwal->kontak->no_telpon;
                $payload = [
                    'phone' => $phone,
                    'message' => $message
                ];

                $tanggal_tagihan_berikutnya = $this-
>hitung_tanggal_tagihan_berikutnya($jadwal->tanggal_kirim);
                $this->add_laporan($jadwal);

                $laporan = Laporan::where('jadwal_id',
$jadwal->id)
                    ->orderBy('created_at', 'desc') // Order
by 'created_at' in descending order (latest first)
                    ->first();

                if ($laporan) {
                    $this->sendTextWatsapp($phone,
$message);

                    // $this->sendMessage($payload);
                    $jadwal->update([
                        'tanggal_kirim' =>
$tanggal_tagihan_berikutnya,
                    ]);
                    $laporan->update([
                        'tanggal_terkirim' => Carbon::now(),

```

```

        'status' => 'terkirim',
    });
    \Log::info("Berhasil Mengirim Tagihan "
. $jadwal->kontak->nama_lengkap . " di jalankan " . date('Y-m-d
H:i:s'));
    } else {
        \Log::info("Tidak ada laporan untuk
jadwal " . $jadwal->kontak->nama_lengkap . " di jalankan " .
date('Y-m-d H:i:s'));
    }
}
} catch (\Throwable $th) {
    \Log::info("Gagal Mengirim Tagihan di jalankan " .
date('Y-m-d H:i:s') . $th->getMessage());
}
}

```

Penulisan pada kode diatas adalah hasil dari ringkasan penelitian peneliti, untuk dapat melihat kode lengkapnya dapat dilihat pada lampiran A6.

4.3.11. Menjalankan *Task* Secara Otomatis

Pada tahap ini, peneliti perlu menjalankan tugas yang telah dibuat terhadap langkah sebelumnya. Tahap-tahapnya adalah sebagai berikut:

Pada folder *Commands* file *Tagihan.php* kita perlu mendefinisikan waktu yang diperlukan untuk menjalankan perintah tersebut. Buka file *routes/console.php* dan tambahkan kode berikut dibaris terbawah.

```

Schedule::command('tagihan:cron')
->everyMinute()
->onFailure(function (Throwable $exception) {
    \Log::error('Tagihan cron job failed: ' . $exception-
->getMessage());
});

```

Pada langkah selanjutnya, tuliskan perintah dibawah ini didalam *terminal* untuk menjalankan tugas yang ada pada *file routes/console.php*.

```

Php artisan schedule:work

```

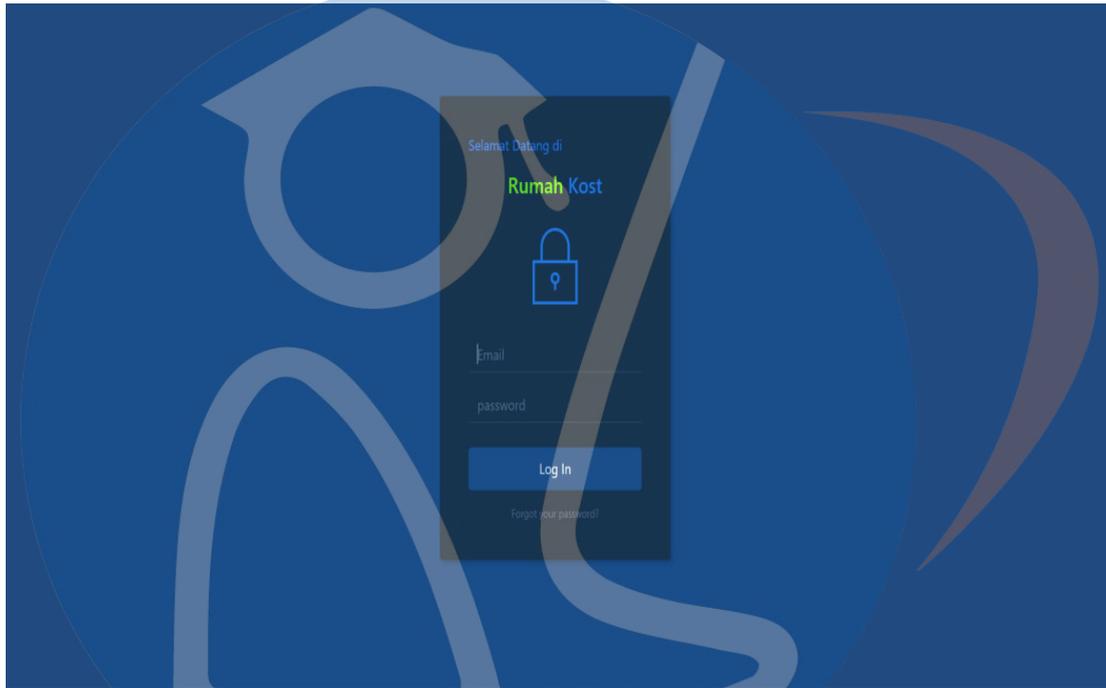
4.4. Evaluasi Sistem

Evaluasi sistem merupakan penilaian kinerja dan efektivitas dari aplikasi yang telah dibuat. Dalam penelitian ini, Proses evaluasi sistem menggunakan metode

blackbox testing dan usability testing untuk memastikan aplikasi memenuhi ekspektasi pengguna.

4.4.1. Tampilan Antarmuka Aplikasi

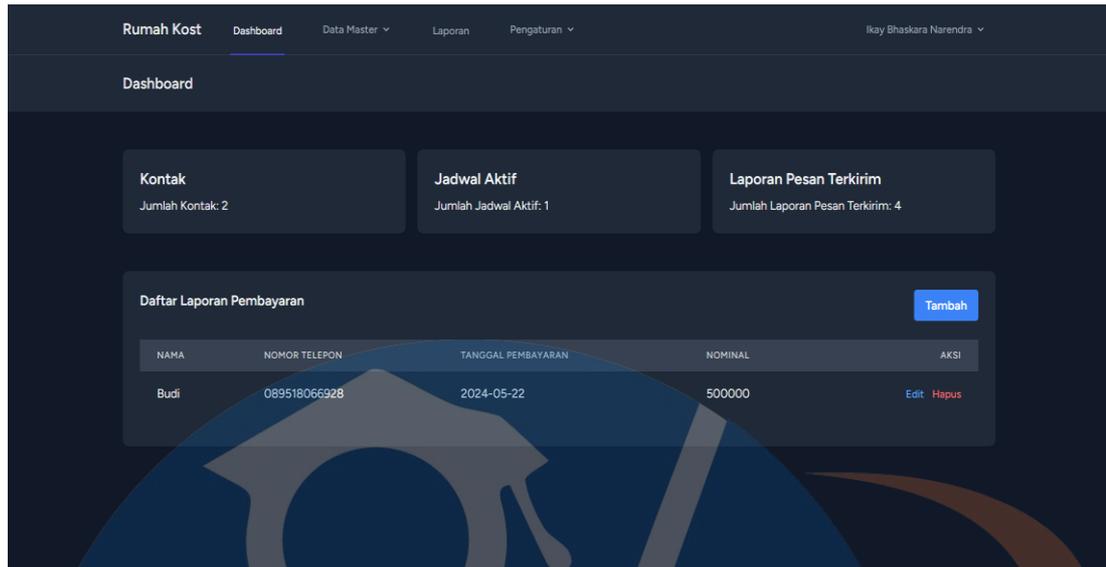
Pada tahap tampilan antarmuka aplikasi, peneliti akan menampilkan desain antarmuka aplikasi penagihan iuran rumah kost, yang mencakup antarmuka untuk peran admin dan pengelola kost.



Gambar 4. 21 Implementasi User Interface - Halaman Login

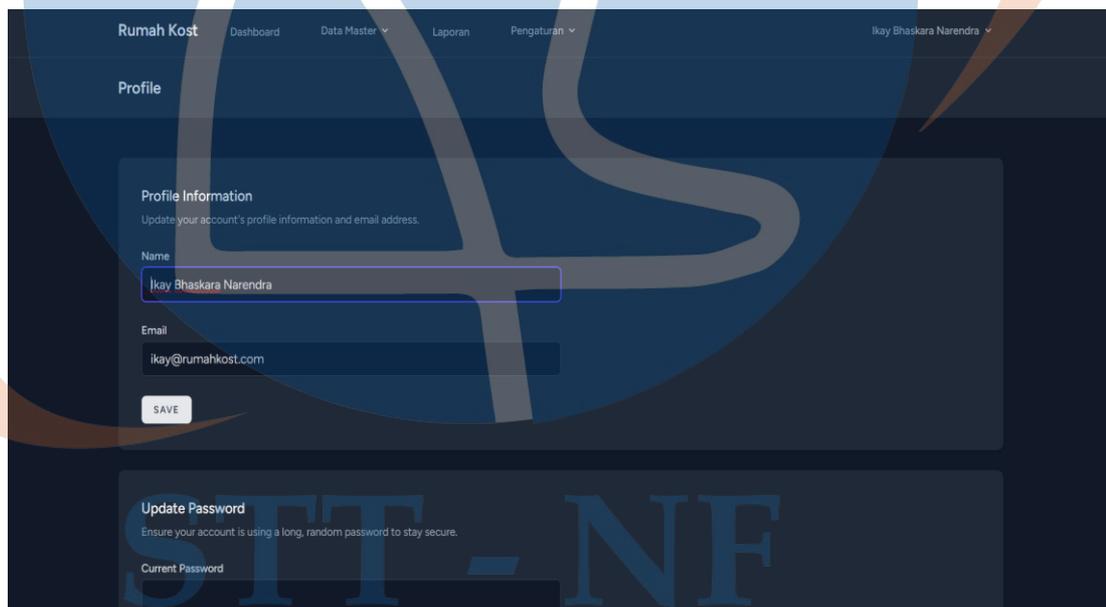
Gambar 4. 21 adalah tampilan awal dari *user interface* untuk melakukan login ke dalam aplikasi. Pada halaman ini pengguna diharuskan memasukkan data email dan password yang benar untuk bisa mengakses aplikasi.

STT - NF



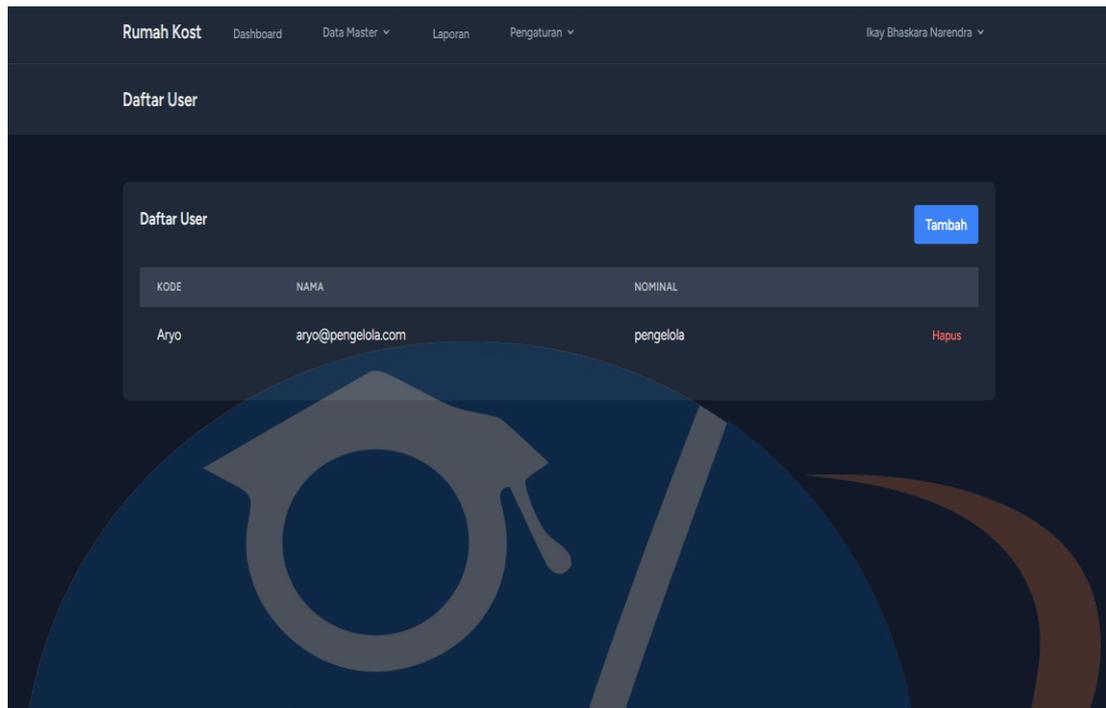
Gambar 4. 22 Implementasi User Interface - Halaman Dashboard

Gambar 4. 22 adalah *user interface* dari halaman *dashboard*. Halaman *dashboard* akan menampilkan informasi tertentu yang dibutuhkan oleh *user*. Halaman ini dapat diakses untuk peran admin dan pengelola.



Gambar 4. 23 Implementasi User Interface - Halaman Edit Profile

Gambar 4. 23 adalah *user interface* dari halaman *edit profile*. Didalam halaman ini *user* dapat melakukan perubahan data pribadi seperti nama, *email*, *password*, dan hapus akun.

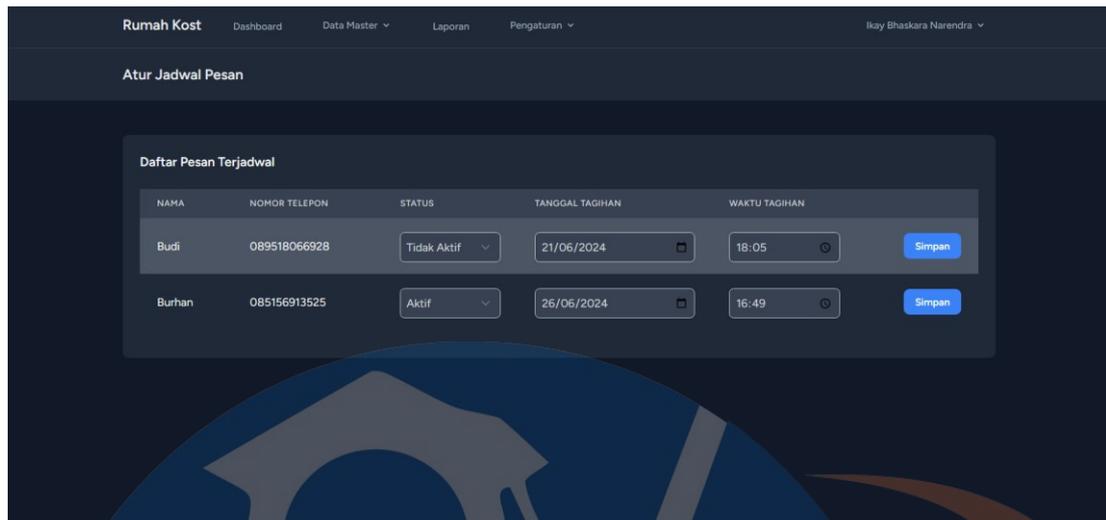


Gambar 4. 24 Implementasi User Interface - Halaman Daftar User



Gambar 4. 25 Implementasi User Interface - Halaman Tambah User

Gambar 4. 24 dan Gambar 4. 25 merupakan *user interface* dari halaman kelola *user*, yang hanya bisa diakses dengan peran admin. Halaman ini berfungsi untuk menambah *user* dengan *role* pengelola. disini peran admin dapat menambah dan menghapus *user* namun tidak dapat merubah data *user*.



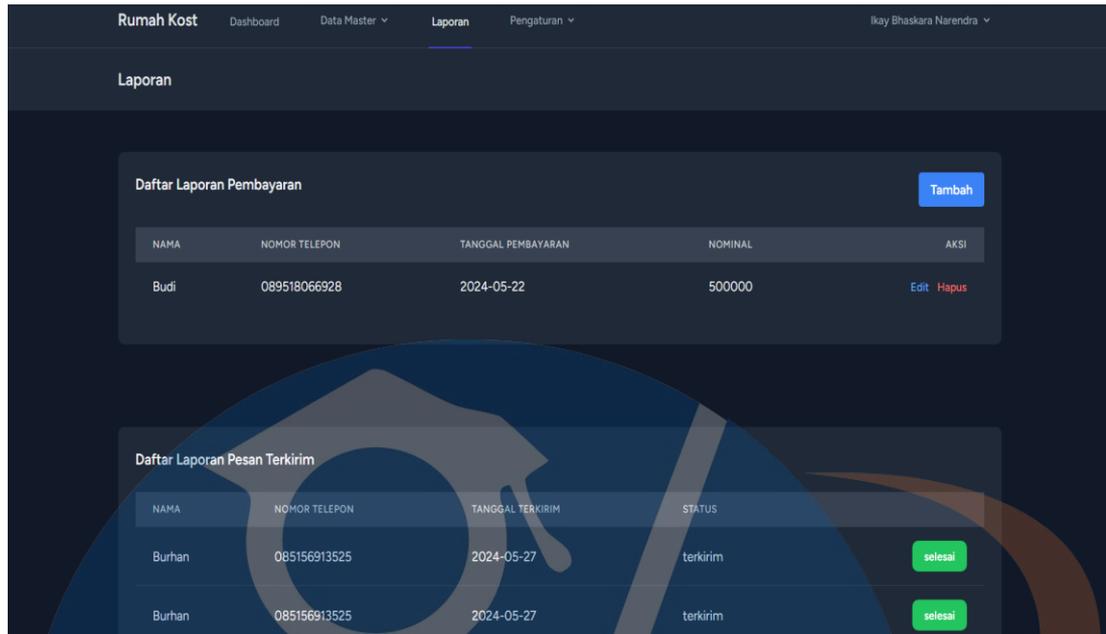
Gambar 4. 26 Implementasi User Interface - Halaman Jadwal

Gambar 4. 26 adalah *user interface* dari halaman jadwal. Halaman jadwal berfungsi untuk mengelola pengiriman pesan kepada penghuni kost, hal yang dapat dikelola meliputi status, tanggal, dan waktu pengiriman user. Peran admin dan pengelola dapat mengakses halaman ini.



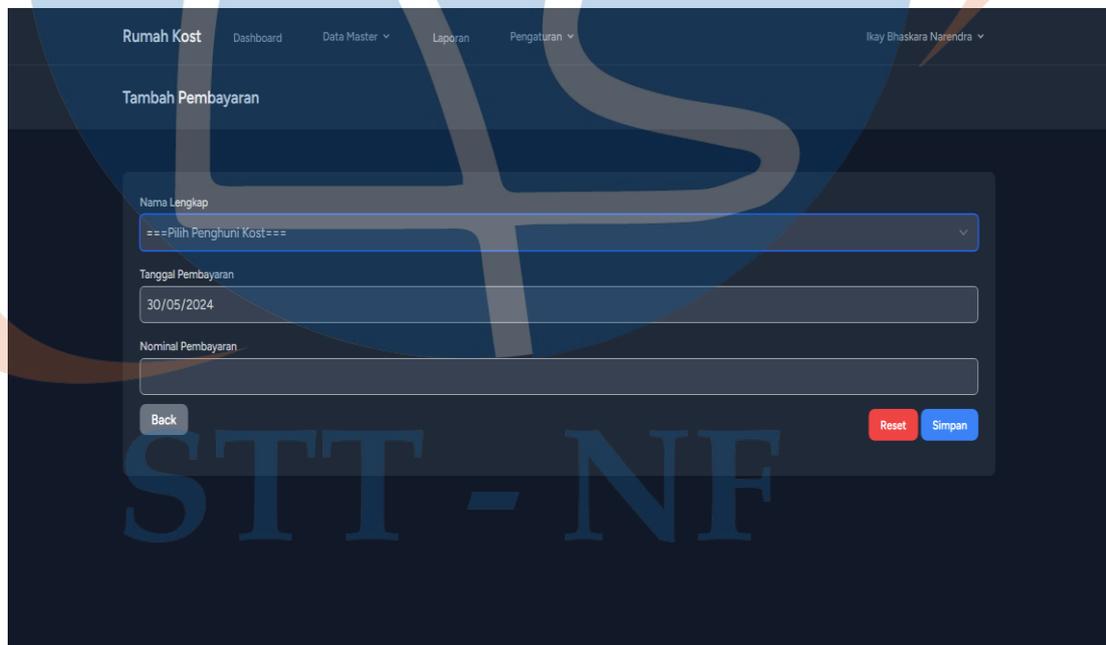
Gambar 4. 27 Implementasi User Interface - Halaman Perangkat

Gambar 4. 27 merupakan *user interface* dari halaman perangkat, pada halaman ini pengguna dengan peran admin dapat merubah nomor telepon pengirim dan format pesan teks pengiriman. Adapun dengan peran pengelola hanya dapat merubah format pesan teks pengiriman.



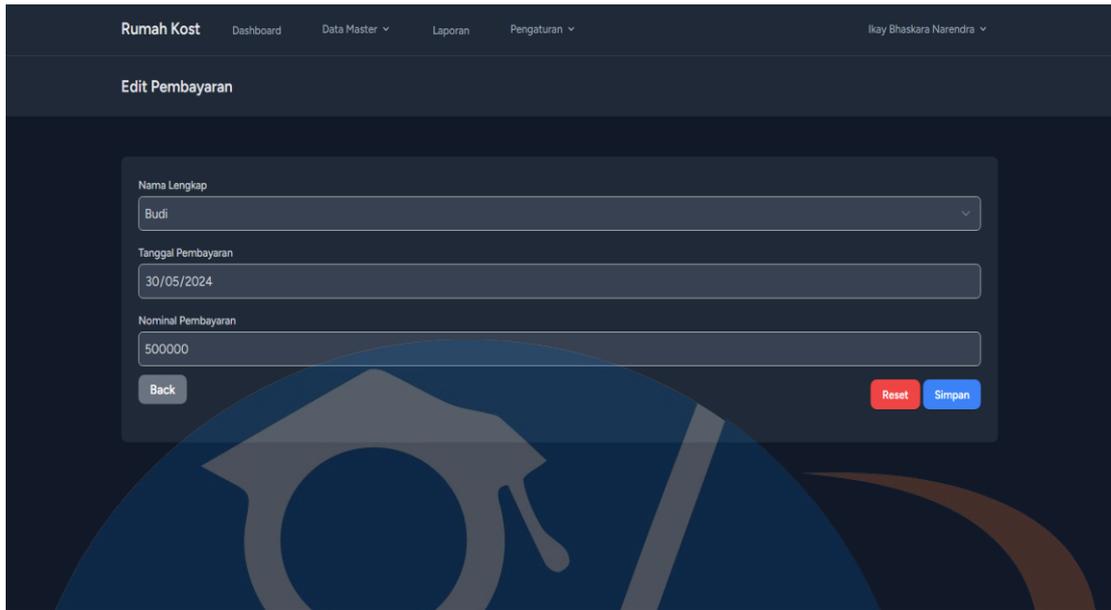
Gambar 4. 28 Implementasi User Interface - Halaman Daftar Laporan

Gambar 4. 28 merupakan *user interface* dari halaman daftar laporan. Pada halaman ini user dapat mengelola data laporan pembayaran dan melihat informasi pengiriman pesan, semua peran dapat mengakses halaman ini.



Gambar 4. 29 Implementasi User Interface - Halaman Tambah Laporan

Gambar 4. 29 merupakan *user interface* untuk menambah data laporan pembayaran. Peran admin dan pengelola dapat mengakses halaman ini untuk menambah data laporan.



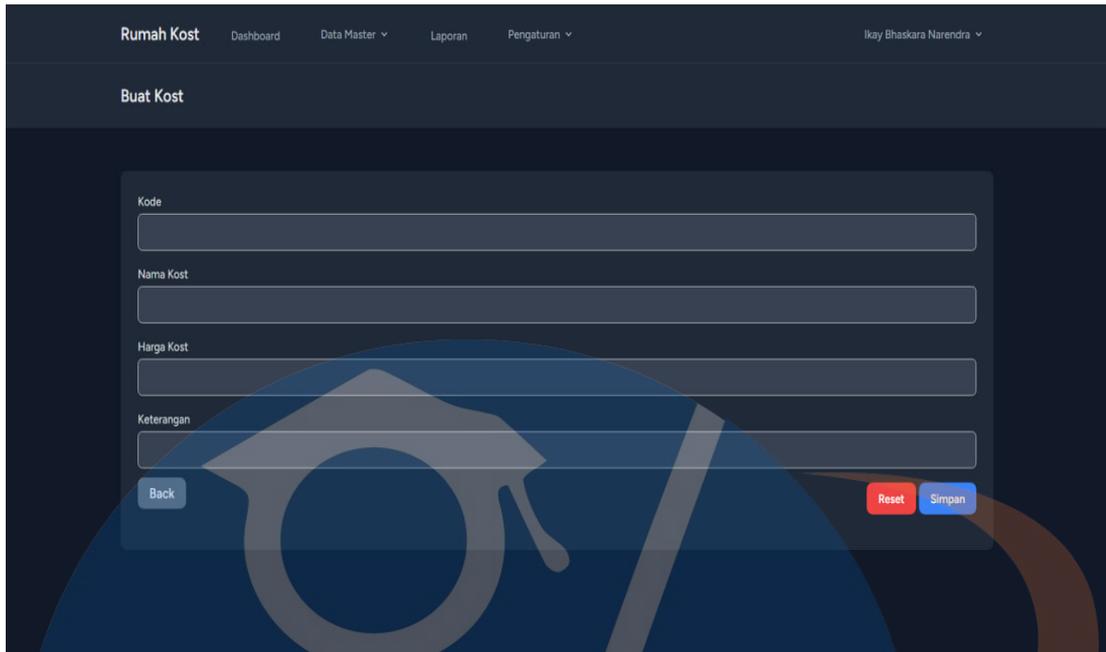
Gambar 4. 30 Implementasi User Interface - Halaman Edit Laporan

Gambar 4. 30 merupakan *user interface* untuk mengubah data laporan pembayaran. Peran admin dan pengelola dapat mengakses halaman ini untuk mengubah data laporan pembayaran.

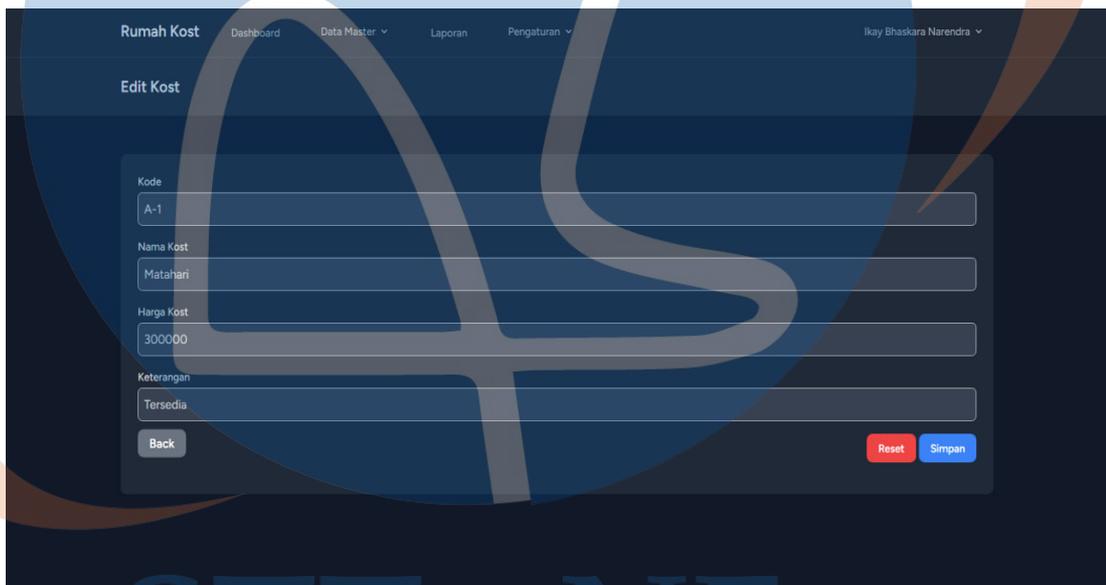


Gambar 4. 31 Implementasi User Interface - Halaman Daftar Kost

Gambar 4. 31 adalah *user interface* dari halaman daftar gambar, pada halaman ini peran admin dan pengelola dapat melihat informasi singkat dari data kost yang telah dibuat.

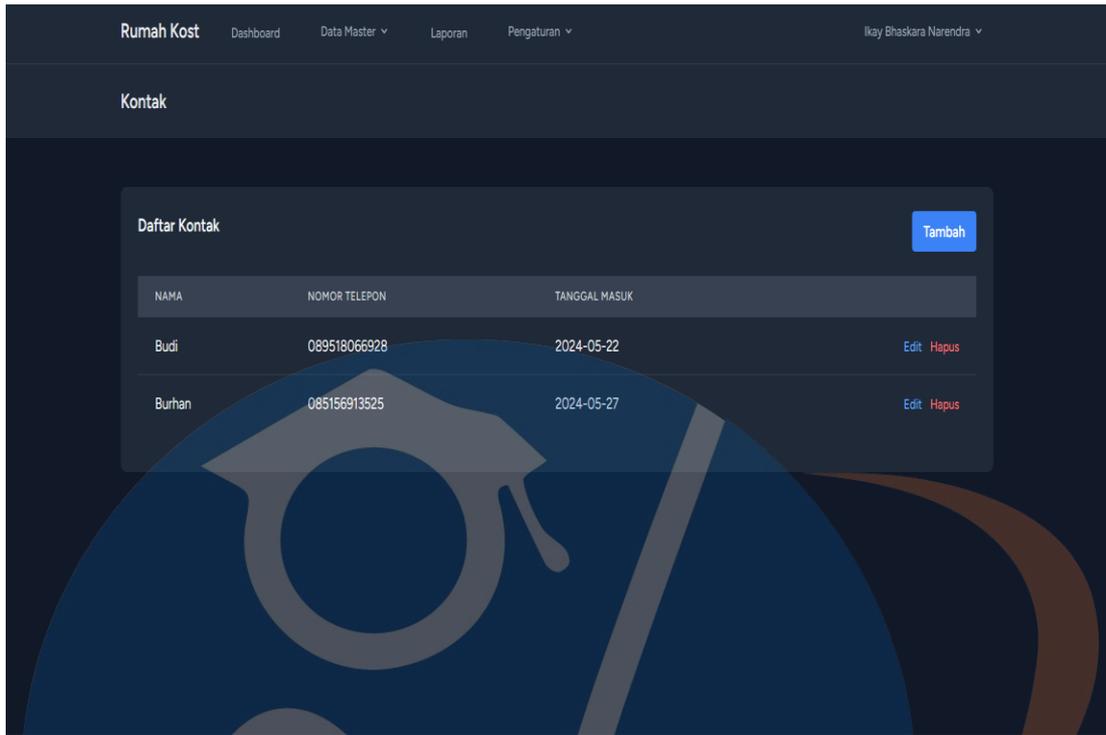


Gambar 4. 32 Implementasi User Interface - Halaman Tambah Data Kost



Gambar 4. 33 Implementasi User Interface - Halaman Mengubah Data Kost

Gambar 4. 32 dan Gambar 4. 33 adalah *user interface* dari halaman menambah data kost baru dan mengubah data kost yang telah ada, peran admin dan pengelola dapat mengakses halaman ini.



Gambar 4. 34 Implementasi User Interface - Halaman Daftar Kontak



Gambar 4. 35 Implementasi User Interface - Halaman Tambah Kontak

Rumah Kost Dashboard Data Master Laporan Pengaturan Ikay Bhaskara Narendra

Edit Kontak

NIK

Nama Lengkap
Budi

Nama Panggilan

Tempat Lahir

Tanggal Lahir
dd/mm/yyyy

Jenis Kelamin
Laki-laki

Agama

Gambar 4. 36 Implementasi User Interface - Halaman Merubah Kontak

Gambar 4. 34, Gambar 4. 35 dan Gambar 4. 36 adalah *user interface* dari halaman pengelolaan kontak. Peran admin dan pengelola dapat mengakses halaman ini. Halaman ini meliputi penambahan data kontak, mengubah data kontak, dan menghapus data kontak.

4.4.2. **Blackbox Testing**

Blackbox testing digunakan untuk mengetahui fungsionalitas aplikasi tanpa memperhatikan kode sumber. Tujuannya adalah untuk memastikan semua fitur aplikasi berfungsi sesuai dengan harapan yang telah ditetapkan. Berbagai skenario pengujian diterapkan dengan memberikan input tertentu dan melihat output yang dihasilkan, agar dapat menentukan apakah aplikasi memenuhi persyaratan yang telah ditetapkan.

4.4.2.1. **Melakukan Pengujian Fitur Login**

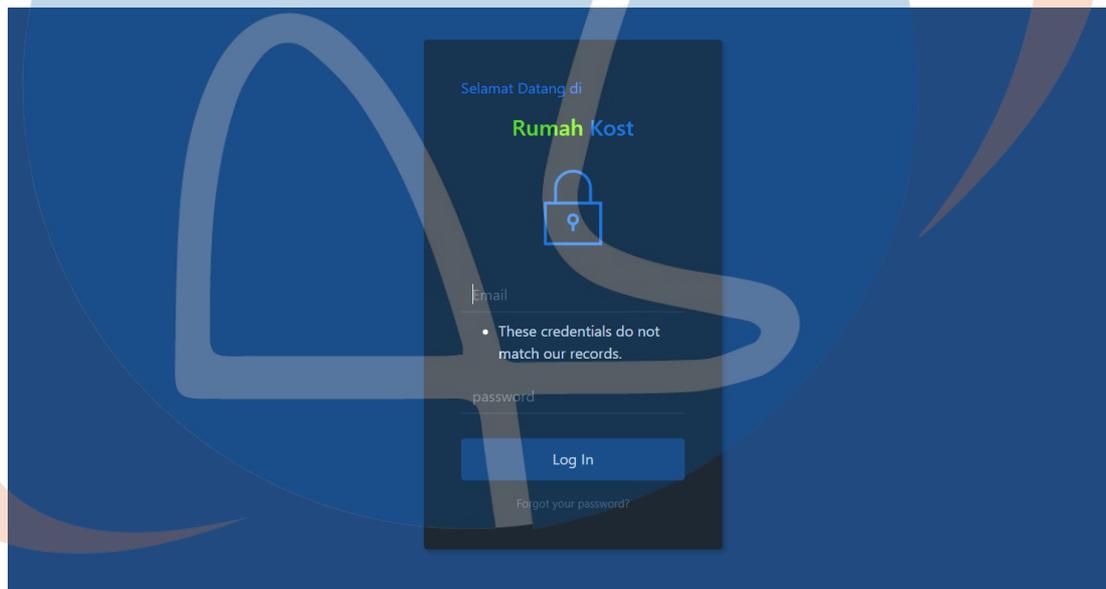
Tahap ini peneliti melakukan pengujian fitur *login* menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 27 Hasil Pengujian Blackbox - Login

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|-------------|--|-----------------------------|--------------|
| 1 | Login | Admin 1 | Pengujian proses login dengan data benar | Halaman utama ditampilkan | Ya |
| 2 | Login | Admin 1 | Pengujian proses login dengan data salah | Pesan kesalahan ditampilkan | Ya |
| 3 | Login | Admin 2 | Pengujian proses login dengan data benar | Halaman utama ditampilkan | Ya |
| 4 | Login | Admin 2 | Pengujian proses login dengan data salah | Pesan kesalahan ditampilkan | Ya |
| 5 | Login | Pengelola 1 | Pengujian proses login dengan data benar | Halaman utama ditampilkan | Ya |
| 6 | Login | Pengelola 1 | Pengujian proses login dengan data salah | Pesan kesalahan ditampilkan | Ya |
| 7 | Login | Pengelola 2 | Pengujian proses login dengan data benar | Halaman utama ditampilkan | Ya |
| 8 | Login | Pengelola 2 | Pengujian proses login dengan data salah | Pesan kesalahan ditampilkan | Ya |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|-------------|--|-----------------------------|--------------|
| 9 | Login | Pengelola 3 | Pengujian proses login dengan data benar | Halaman utama ditampilkan | Ya |
| 10 | Login | Pengelola 3 | Pengujian proses login dengan data salah | Pesan kesalahan ditampilkan | Ya |

Setelah melakukan pengujian proses *login* gagal, sistem aplikasi akan menampilkan pesan kesalahan di dalam halaman *login*. Jika dalam pengujian user memasukkan dengan data benar maka Gambar 4. 37 akan ditampilkan. Berikut adalah gambaran dari pesan kesalahan tersebut:



Gambar 4. 37 Hasil Pengujian Blackbox - Halaman Login

4.4.2.2. Melakukan Pengujian Menu Dashboard

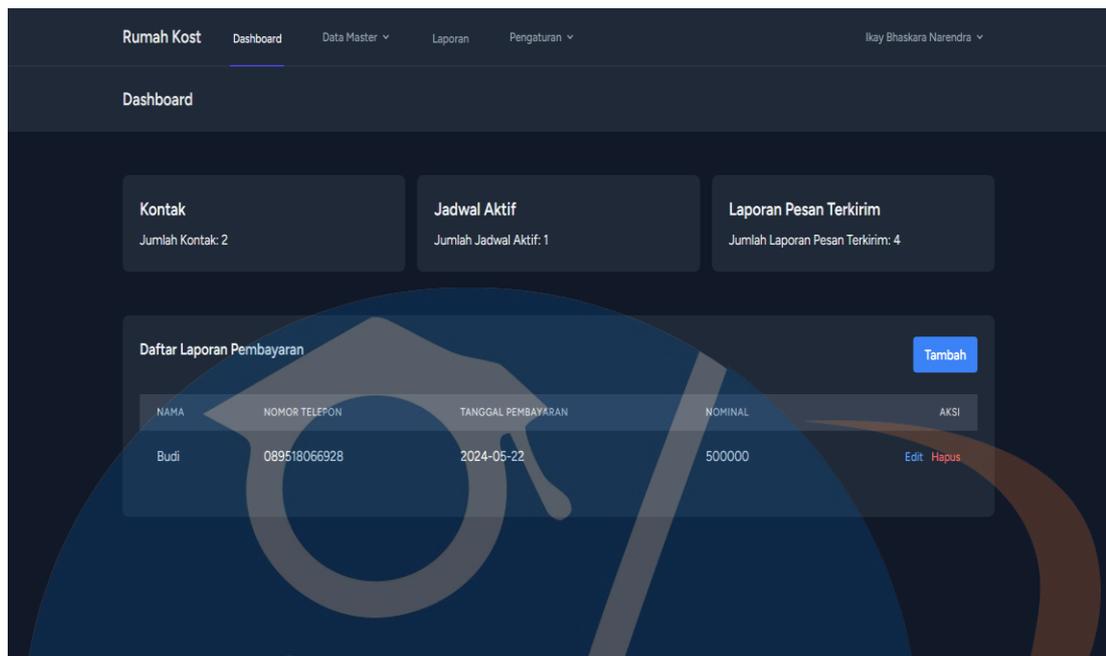
Tahap ini peneliti melakukan pengujian *menu dashboard* menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 28 Hasil Pengujian Blackbox – Melihat Dashboard

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|-------------|--|---------------------------------|--------------|
| 1 | Dashboard | Admin 1 | Pengujian proses melihat informasi dashboard | Informasi dashboard ditampilkan | Ya |
| 2 | Dashboard | Admin 2 | Pengujian proses melihat informasi dashboard | Informasi dashboard ditampilkan | Ya |
| 3 | Dashboard | Pengelola 1 | Pengujian proses melihat informasi dashboard | Informasi dashboard ditampilkan | Ya |
| 4 | Dashboard | Pengelola 2 | Pengujian proses melihat informasi dashboard | Informasi dashboard ditampilkan | Ya |
| 5 | Dashboard | Pengelola 3 | Pengujian proses melihat informasi dashboard | Informasi dashboard ditampilkan | Ya |

Berikut ini adalah gambaran hasil dari pengujian proses melihat halaman *dashboard* pada aplikasi sistem penagihan iuran rumah kost:

STT - NF



Gambar 4. 38 Hasil Pengujian Blackbox – Melihat Dashboard

4.4.2.3. Melakukan Pengujian Mengelola Data Kost

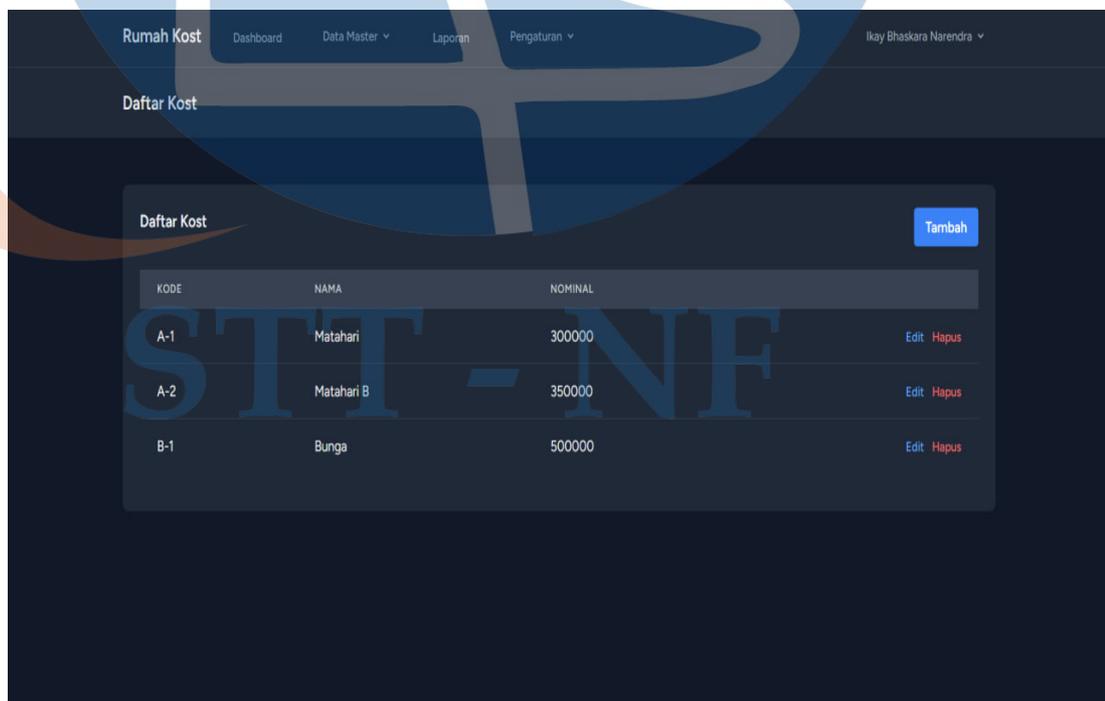
Tahap ini peneliti melakukan pengujian melihat daftar kost menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 29 Hasil Pengujian Blackbox - Mengelola Data Kost

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---------------------|---------|---|---|--------------|
| 1 | Mengelola data kost | Admin 1 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Ya |
| 2 | Mengelola data kost | Admin 2 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Ya |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---------------------|-------------|---|---|--------------|
| 3 | Mengelola data kost | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Ya |
| 4 | Mengelola data kost | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Ya |
| 5 | Mengelola data kost | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Ya |

Berikut ini adalah gambaran hasil dari pengujian proses mengelola data kost pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 39 Hasil Pengujian Blackbox - Mengelola Data Kost

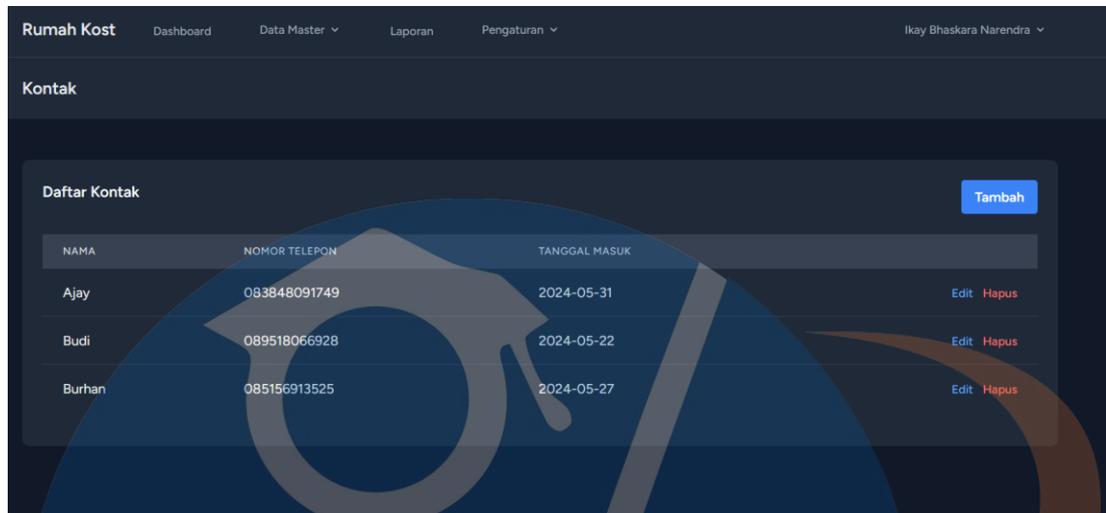
4.4.2.4. Melakukan Pengujian Mengelola Data Kontak

Tahap ini peneliti melakukan pengujian melihat daftar kontak menggunakan metode blackbox testing. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 30 Hasil Pengujian Blackbox - Mengelola Data Kontak

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|-----------------------|-------------|---|---|--------------|
| 1 | Mengelola data Kontak | Admin 1 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Ya |
| 2 | Mengelola data Kontak | Admin 2 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Ya |
| 3 | Mengelola data Kontak | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Ya |
| 4 | Mengelola data Kontak | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Ya |
| 5 | Mengelola data Kontak | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Ya |

Berikut ini adalah gambaran hasil dari pengujian proses melihat daftar data Kontak pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 40 Hasil Pengujian Blackbox - Mengelola Data Kontak

4.4.2.5. Melakukan Pengujian Mengelola Laporan Pembayaran

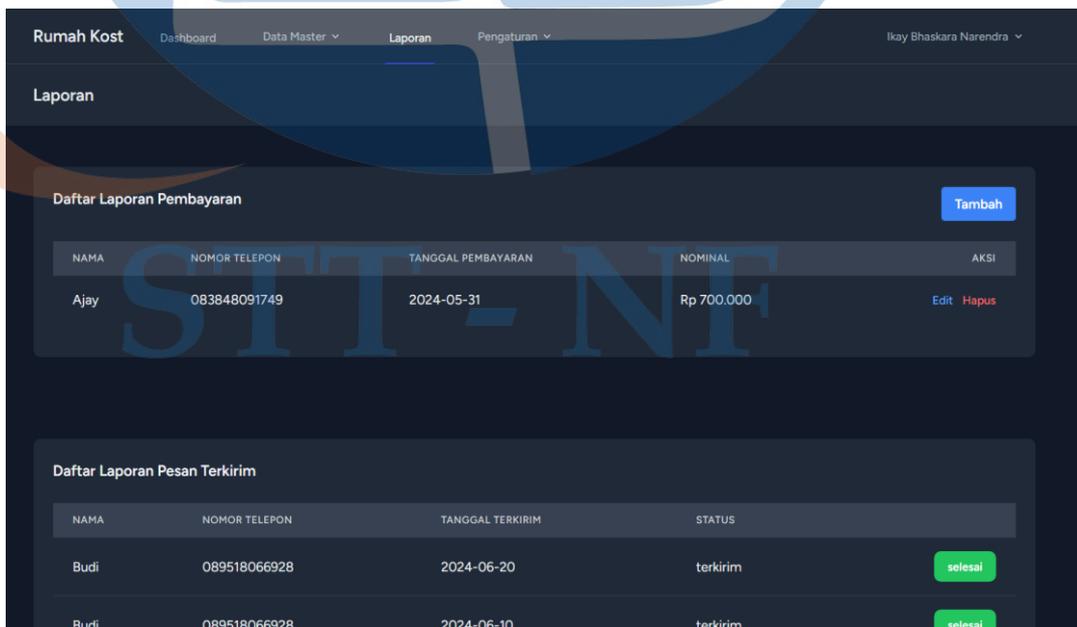
Tahap ini peneliti melakukan pengujian melihat daftar laporan pembayaran menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 31 Hasil Pengujian Blackbox - Mengelola Laporan Pembayaran

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|-------------------|---------|---|--|--------------|
| 1 | Mengelola Laporan | Admin 1 | Menguji proses menambah, mengubah, dan menghapus Laporan Pembayaran | Data berhasil ditambahkan, diubah, dihapus | Ya |
| 2 | Mengelola Laporan | Admin 2 | Menguji proses menambah, mengubah, dan menghapus Laporan Pembayaran | Data berhasil ditambahkan, diubah, dihapus | Ya |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|-------------------|-------------|---|--|--------------|
| 3 | Mengelola Laporan | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus Laporan Pembayaran | Data berhasil ditambahkan, diubah, dihapus | Ya |
| 4 | Mengelola Laporan | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus Laporan Pembayaran | Data berhasil ditambahkan, diubah, dihapus | Ya |
| 5 | Mengelola Laporan | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus Laporan Pembayaran | Data berhasil ditambahkan, diubah, dihapus | Ya |

Berikut ini adalah gambaran hasil dari pengujian proses melihat daftar laporan pembayaran pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 41 Hasil Pengujian Blackbox - Mengelola Laporan Pembayaran

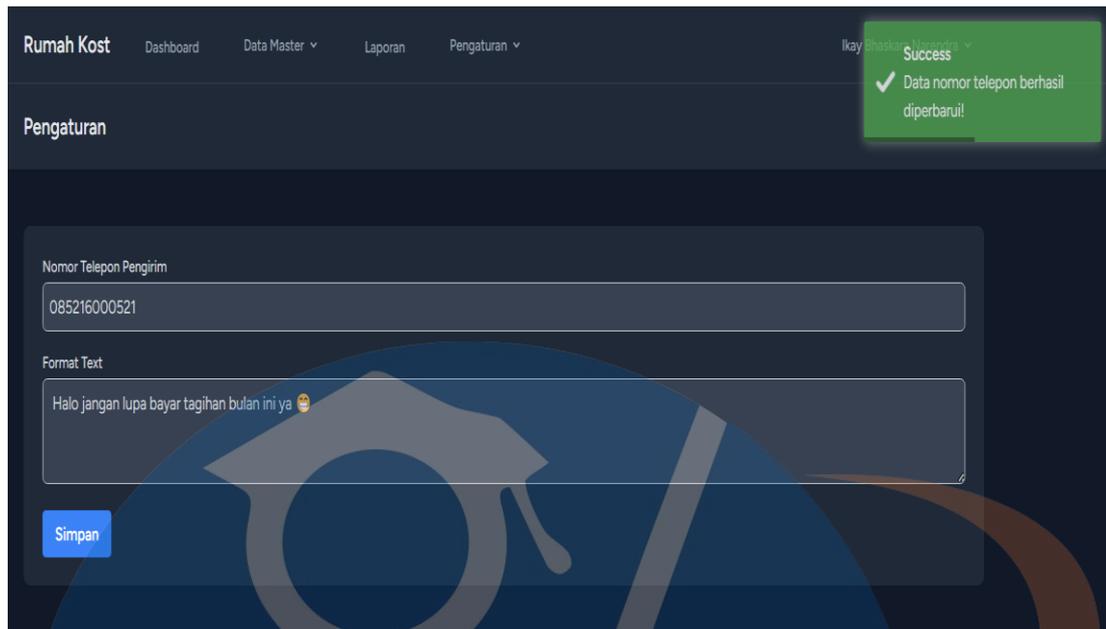
4.4.2.6. Melakukan Pengujian Mengelola Perangkat

Tahap ini peneliti melakukan pengujian mengelola perangkat menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 32 Hasil Pengujian Blackbox - Mengelola Perangkat

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|--|-------------|---|--|--------------|
| 1 | Menu pengaturan > perangkat, peran admin | Admin 1 | Pengujian proses merubah formulir telepon dan format teks | Formulir telepon dan format teks berhasil diperbarui | Ya / Tidak |
| 2 | Menu pengaturan > perangkat, peran admin | Admin 2 | Pengujian proses merubah formulir telepon dan format teks | Formulir telepon dan format teks berhasil diperbarui | Ya / Tidak |
| 3 | Menu pengaturan > perangkat, peran pengelola | Pengelola 1 | Pengujian proses merubah formulir format teks | Formulir format teks berhasil diperbarui | Ya |
| 4 | Menu pengaturan > perangkat, peran pengelola | Pengelola 2 | Pengujian proses merubah formulir format teks | Formulir format teks berhasil diperbarui | Ya |
| 5 | Menu pengaturan > perangkat, peran pengelola | Pengelola 3 | Pengujian proses merubah formulir format teks | Formulir format teks berhasil diperbarui | Ya |

Berikut ini adalah gambaran hasil dari pengujian pengaturan perangkat peran admin pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 42 Hasil Pengujian Blackbox - Mengelola Perangkat

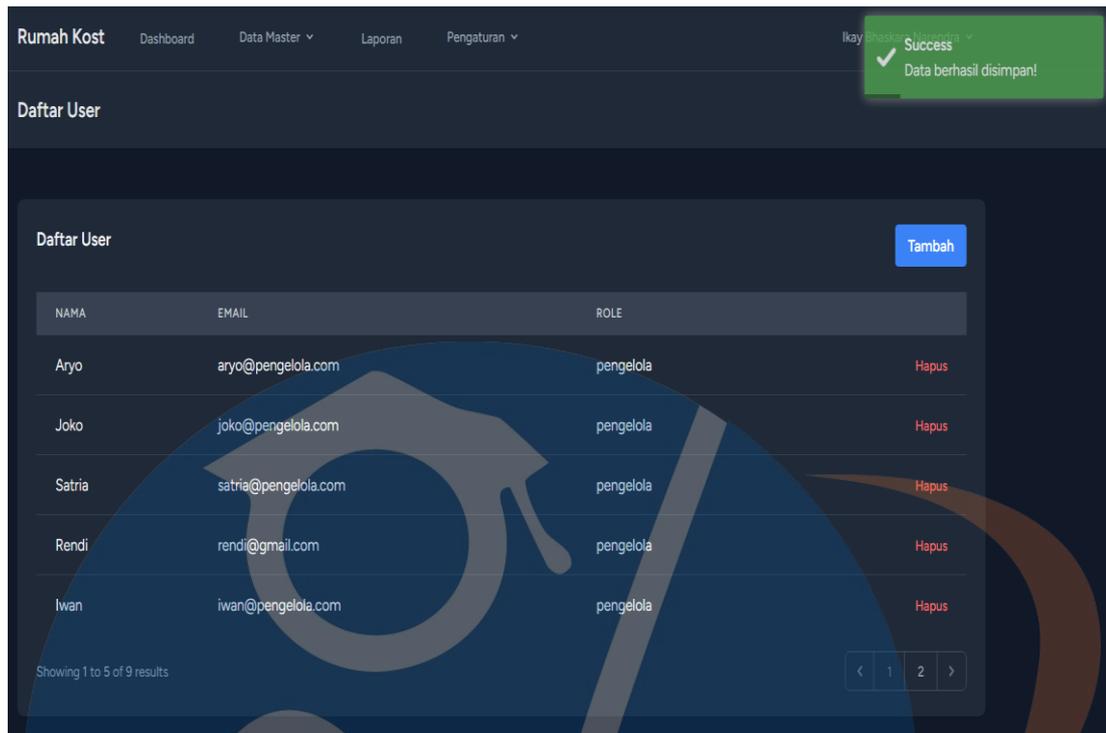
4.4.2.7. Melakukan Pengujian Mengelola Data User

Tahap ini peneliti melakukan pengujian mengelola data *user* menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 33 Hasil Pengujian Blackbox - Mengelola User

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|---------|--|--|--------------|
| 1 | Mengelola User | Admin 1 | Menguji proses menambah, dan menghapus data user | Data berhasil ditambahkan, dan dihapus | Ya |
| 2 | Mengelola User | Admin 2 | Menguji proses menambah, dan menghapus data user | Data berhasil ditambahkan, dan dihapus | Ya |

Berikut ini adalah gambaran hasil dari pengujian mengelola data *user* pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 43 Hasil Pengujian Blackbox - Mengelola User

4.4.2.8. Melakukan Pengujian Mengubah *Profile* Pribadi

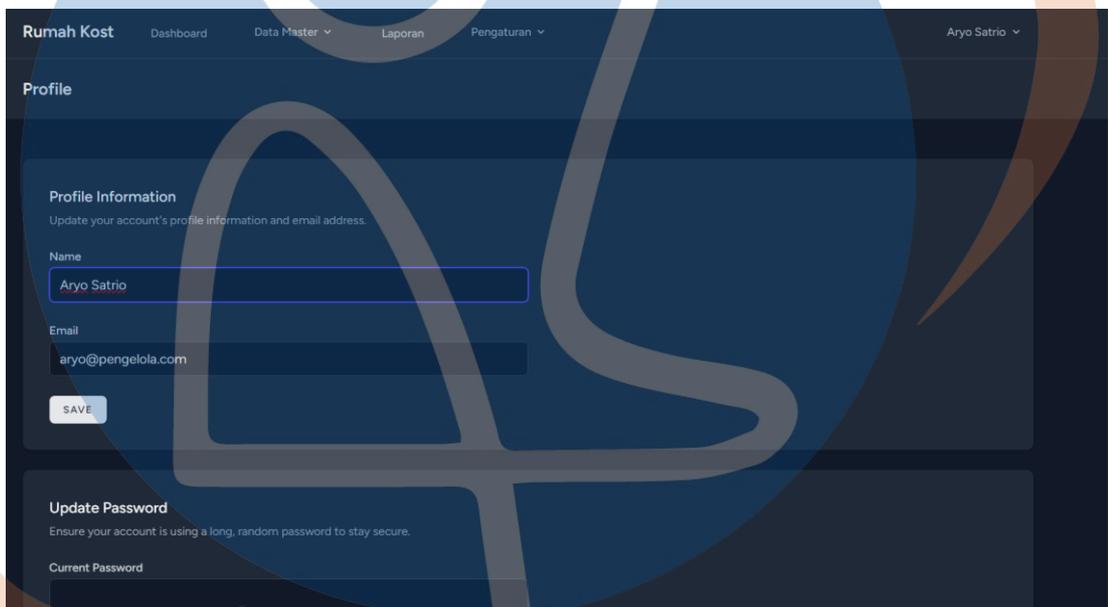
Tahap ini peneliti melakukan pengujian mengubah *profile* pribadi menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 34 Hasil Pengujian Blackbox - Mengedit Profile

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|-------------|--|---------------------------------------|--------------|
| 1 | Edit Profile | Admin 1 | Pengujian proses merubah informasi pribadi | Informasi pribadi berhasil diperbarui | Ya |
| 2 | Edit Profile | Admin 2 | Pengujian proses merubah informasi pribadi | Informasi pribadi berhasil diperbarui | Ya |
| 3 | Edit Profile | Pengelola 1 | Pengujian proses merubah informasi pribadi | Informasi pribadi berhasil diperbarui | Ya |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|-------------|--|---------------------------------------|--------------|
| 4 | Edit Profile | Pengelola 2 | Pengujian proses merubah informasi pribadi | Informasi pribadi berhasil diperbarui | Ya |
| 5 | Edit Profile | Pengelola 3 | Pengujian proses merubah informasi pribadi | Informasi pribadi berhasil diperbarui | Ya |

Berikut ini adalah gambaran hasil dari pengujian mengubah *profile* pribadi pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 44 Hasil Pengujian Blackbox - Mengedit Profile

STT - NF

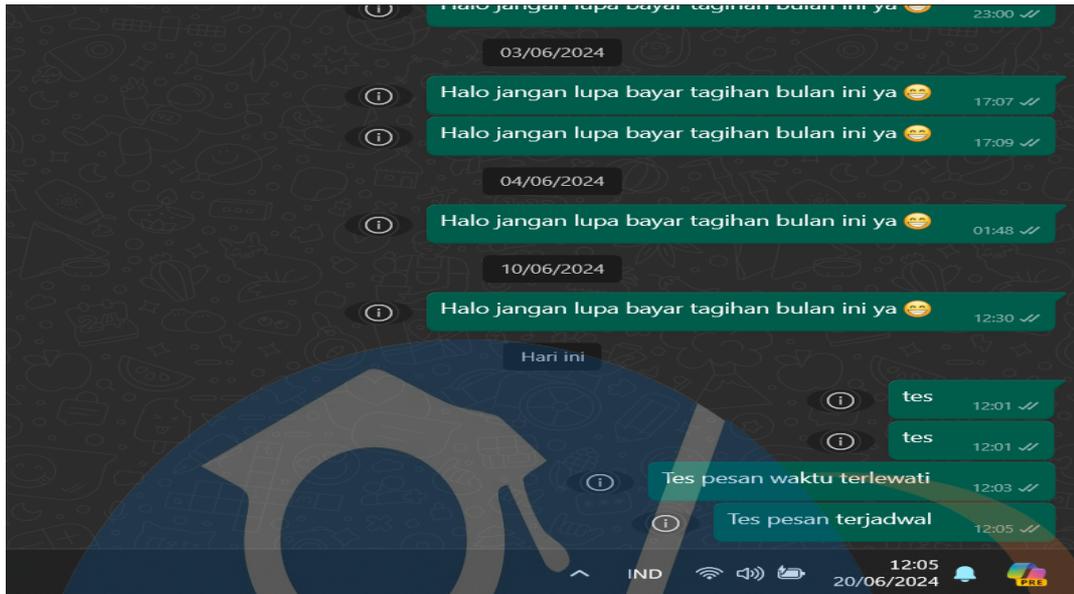
4.4.2.10. Melakukan Pengujian Pengiriman Pesan Waktu Tepat

Tahap ini peneliti melakukan pengujian pengiriman pesan waktu tepat menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 35 Hasil Pengujian Blackbox – Pengiriman Pesan - Waktu Tepat

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|-------------|--|-----------------------|--------------|
| 1 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu tepat | Admin 1 | Pesan dikirim dengan waktu yang telah ditentukan | Pesan terkirim | Ya |
| 2 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu tepat | Admin 2 | Pesan dikirim dengan waktu yang telah ditentukan | Pesan terkirim | Ya |
| 3 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu tepat | Pengelola 1 | Pesan dikirim dengan waktu yang telah ditentukan | Pesan terkirim | Ya |
| 4 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu tepat | Pengelola 2 | Pesan dikirim dengan waktu yang telah ditentukan | Pesan terkirim | Ya |
| 5 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu tepat | Pengelola 3 | Pesan dikirim dengan waktu yang telah ditentukan | Pesan terkirim | Ya |

Berikut ini adalah gambaran hasil dari pengujian pengiriman pesan waktu tepat pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 45 Hasil Pengujian Blackbox – Pengiriman Pesan - Waktu Tepat

4.4.2.11. Melakukan Pengujian Pengiriman Pesan Waktu Terlewati

Tahap ini peneliti melakukan pengujian pengiriman pesan waktu terlewati menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 36 Hasil Pengujian Blackbox - Pengiriman Pesan – Waktu Terlewati

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|--|---------|---|-----------------------|--------------|
| 1 | Mengirim pesan dengan jadwal aktif, tanggal yang tepat, dan waktu yang telah terlewati | Admin 1 | Pesan dikirim dengan waktu yang telah terlewati | Pesan terkirim | Ya / Tidak |
| 2 | Mengirim pesan dengan jadwal aktif, tanggal yang tepat, dan waktu yang telah terlewati | Admin 2 | Pesan dikirim dengan waktu yang telah terlewati | Pesan terkirim | Ya / Tidak |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|-------------|--|-----------------------|--------------|
| 3 | Mengirim pesan dengan jadwal aktif, tanggal yang tepat, dan waktu yang telah melewati | Pengelola 1 | Pesan dikirim dengan waktu yang telah melewati | Pesan terkirim | Ya / Tidak |
| 4 | Mengirim pesan dengan jadwal aktif, tanggal yang tepat, dan waktu yang telah melewati | Pengelola 2 | Pesan dikirim dengan waktu yang telah melewati | Pesan terkirim | Ya / Tidak |
| 5 | Mengirim pesan dengan jadwal aktif, tanggal yang tepat, dan waktu yang telah melewati | Pengelola 3 | Pesan dikirim dengan waktu yang telah melewati | Pesan terkirim | Ya / Tidak |

Berikut ini adalah gambaran hasil dari pengujian pengiriman pesan waktu melewati pada aplikasi sistem penagihan iuran rumah kost:

STT - NF



Gambar 4. 46 Hasil Pengujian Blackbox - Pengiriman Pesan – Waktu Terlewati

4.4.2.12. Melakukan Pengujian Pengiriman Pesan Tanggal Terlewati

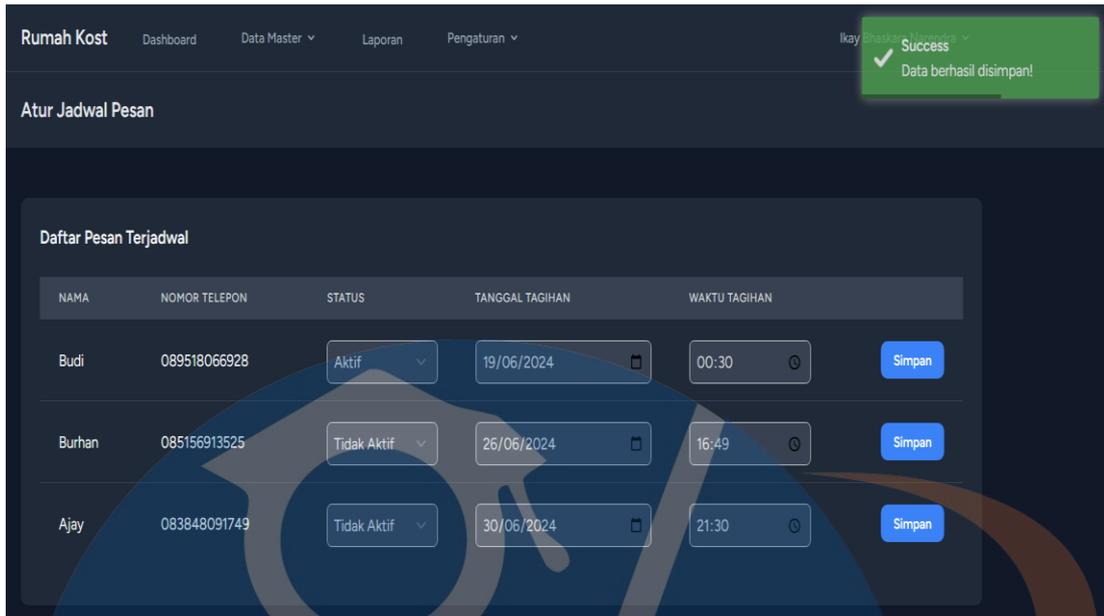
Tahap ini peneliti melakukan pengujian pengiriman pesan tanggal terlewati menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 37 Hasil Pengujian Blackbox – Pengiriman Pesan – Tanggal Terlewati

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|--|---------|---|-----------------------|--------------|
| 1 | Mengirim pesan dengan jadwal aktif, tanggal yang telah terlewati, dan waktu yang tepat | Admin 1 | Tanggal telah terlewati, pesan tidak terkirim | Pesan tidak terkirim | Ya |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|-------------|--|-----------------------|--------------|
| 2 | Mengirim pesan dengan jadwal aktif, tanggal yang telah melewati, dan waktu yang tepat | Admin 2 | Tanggal telah melewati, pesan tidak terkirim | Pesan tidak terkirim | Ya |
| 3 | Mengirim pesan dengan jadwal aktif, tanggal yang telah melewati, dan waktu yang tepat | Pengelola 1 | Tanggal telah melewati, pesan tidak terkirim | Pesan tidak terkirim | Ya |
| 4 | Mengirim pesan dengan jadwal aktif, tanggal yang telah melewati, dan waktu yang tepat | Pengelola 2 | Tanggal telah melewati, pesan tidak terkirim | Pesan tidak terkirim | Ya |
| 5 | Mengirim pesan dengan jadwal aktif, tanggal yang telah melewati, dan waktu yang tepat | Pengelola 3 | Tanggal telah melewati, pesan tidak terkirim | Pesan tidak terkirim | Ya |

Berikut ini adalah gambaran hasil dari pengujian pengiriman pesan tanggal melewati pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 47 Hasil Pengujian Blackbox – Pengiriman Pesan – Tanggal Terlewati

4.4.2.13. Melakukan Pengujian Pengiriman Pesan Status Tidak Aktif

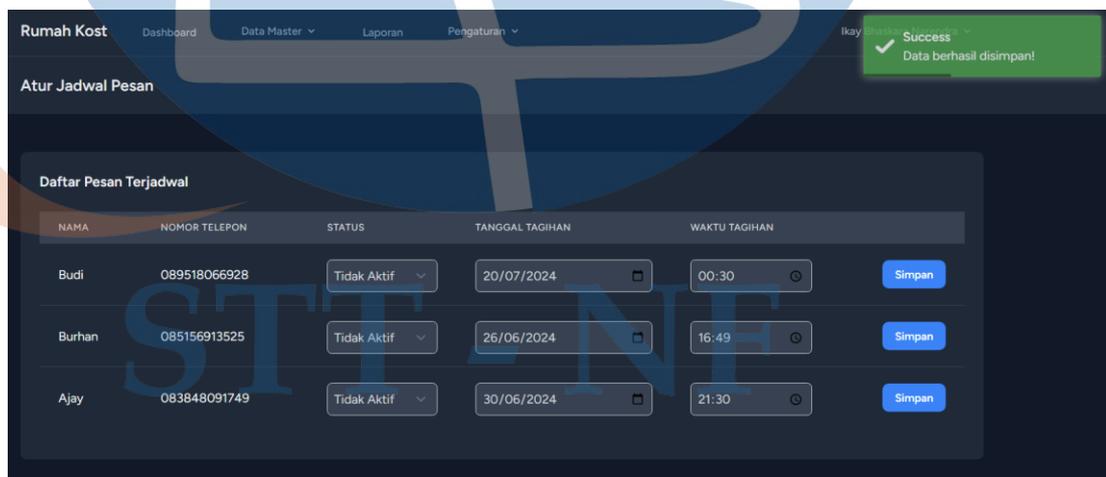
Tahap ini peneliti melakukan pengujian pengiriman pesan status tidak aktif menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 38 Hasil Pengujian Blackbox – Pengiriman Pesan – Jadwal Tidak Aktif

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|---------|--|-----------------------|--------------|
| 1 | Mengirim pesan dengan jadwal tidak aktif, tanggal dan waktu tepat | Admin 1 | Jadwal tidak aktif, pesan tidak terkirim | Pesan tidak terkirim | Ya |
| 2 | Mengirim pesan dengan jadwal tidak aktif, tanggal dan waktu tepat | Admin 2 | Jadwal tidak aktif, pesan tidak terkirim | Pesan tidak terkirim | Ya |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|---|-------------|--|-----------------------|--------------|
| 3 | Mengirim pesan dengan jadwal tidak aktif, tanggal dan waktu tepat | Pengelola 1 | Jadwal tidak aktif, pesan tidak terkirim | Pesan tidak terkirim | Ya |
| 4 | Mengirim pesan dengan jadwal tidak aktif, tanggal dan waktu tepat | Pengelola 2 | Jadwal tidak aktif, pesan tidak terkirim | Pesan tidak terkirim | Ya |
| 5 | Mengirim pesan dengan jadwal tidak aktif, tanggal dan waktu tepat | Pengelola 3 | Jadwal tidak aktif, pesan tidak terkirim | Pesan tidak terkirim | Ya |

Berikut ini adalah gambaran hasil dari pengujian pengiriman pesan status tidak aktif pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 48 Hasil Pengujian Blackbox – Pengiriman Pesan – Jadwal Tidak Aktif

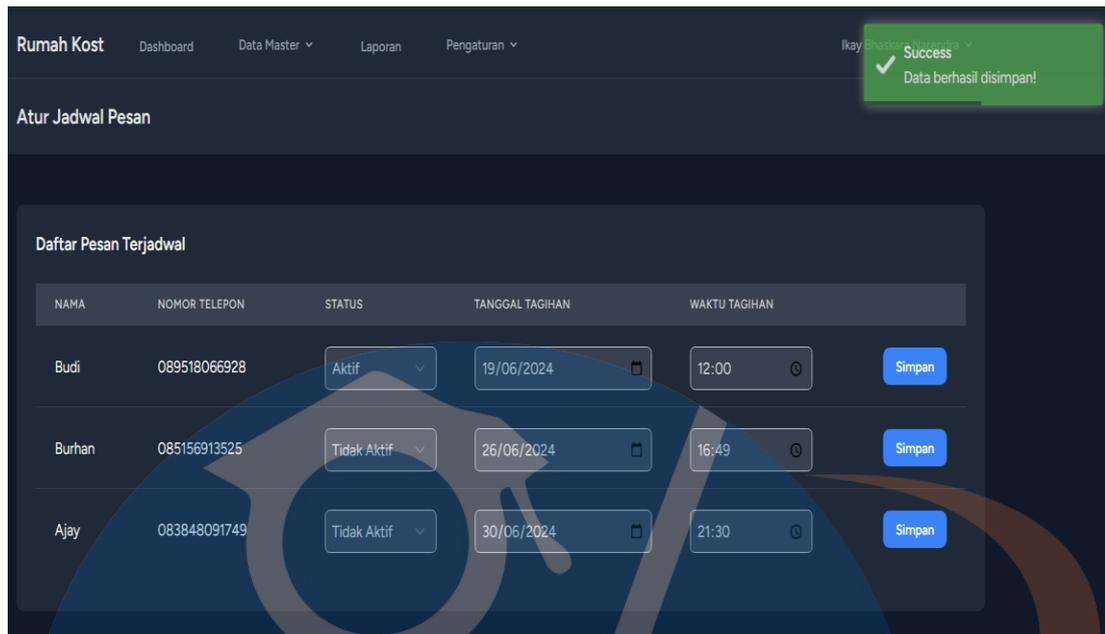
4.4.2.14. Melakukan Pengujian Pengiriman Pesan Tanggal dan Jam Salah

Tahap ini peneliti melakukan pengujian pengiriman pesan tanggal dan jam salah menggunakan metode *blackbox testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 39 Hasil Pengujian Blackbox - Pengiriman Pesan – Tanggal dan Waktu Salah

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|--|-------------|---|-----------------------|--------------|
| 1 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu yang salah | Admin 1 | Tanggal dan waktu salah, pesan tidak terkirim | Pesan tidak terkirim | Ya |
| 2 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu yang salah | Admin 2 | Tanggal dan waktu salah, pesan tidak terkirim | Pesan tidak terkirim | Ya |
| 3 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu yang salah | Pengelola 1 | Tanggal dan waktu salah, pesan tidak terkirim | Pesan tidak terkirim | Ya |
| 4 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu yang salah | Pengelola 2 | Tanggal dan waktu salah, pesan tidak terkirim | Pesan tidak terkirim | Ya |
| 5 | Mengirim pesan dengan jadwal aktif, tanggal dan waktu yang salah | Pengelola 3 | Tanggal dan waktu salah, pesan tidak terkirim | Pesan tidak terkirim | Ya |

Berikut ini adalah gambaran hasil dari pengujian pengiriman pesan tanggal dan jam salah pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 49 Hasil Pengujian Blackbox – Pengiriman Pesan – Jadwal Tidak Aktif

4.4.3. Menghitung Hasil Pengujian *Blackbox Testing*

Dalam pengujian *blackbox* terdapat 13 skenario pengujian dan total dari pengujian tersebut terdapat 67 pengujian. Pada pengujian ini melibatkan responden, yang terdiri dari 2 responden admin dan 3 responden pengelola. Total keseluruhan dari pengujian berjumlah 226 pengujian. Dengan perhitungan persentase keberhasilan sebagai berikut:

1. Melakukan Pengujian Fitur Login

- a. Total Pengujian: 10 Pengujian
- b. Lulus : $(10 / 10) \times 100 = 100\%$
- c. Tidak Lulus : $(0/10) \times 100 = 0\%$

2. Melakukan Pengujian Menu Dashboard

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

3. Melakukan Pengujian Mengelola Kost

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

4. Melakukan Pengujian Mengelola Kontak

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

5. Melakukan Pengujian Mengelola Laporan Pembayaran

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

6. Melakukan Pengujian Pengaturan Perangkat

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

7. Melakukan Pengujian Mengelola Data User

- a. Total Pengujian: 2 Pengujian
- b. Lulus : $(2/2) \times 100 = 100\%$
- c. Tidak Lulus : $(0/2) \times 100 = 0\%$

8. Melakukan Pengujian Mengubah Profile Pribadi

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

9. Melakukan Pengujian Pengiriman Pesan Waktu Tepat

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(10/10) \times 100 = 100\%$
- c. Tidak Lulus : $(0/10) \times 100 = 0\%$

10. Melakukan Pengujian Pengiriman Pesan Waktu Terlewat

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

11. Melakukan Pengujian Pengiriman Pesan Tanggal Terlewat

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

12. Melakukan Pengujian Pengiriman Pesan Status Tidak Aktif

- Total Pengujian: 5 Pengujian
- Lulus : $(5/5) \times 100 = 100\%$
- Tidak Lulus : $(0/5) \times 100 = 0\%$

13. Melakukan Pengujian Pengiriman Pesan Tanggal dan Jam Salah

- Total Pengujian: 5 Pengujian
- Lulus : $(5/5) \times 100 = 100\%$
- Tidak Lulus : $(0/5) \times 100 = 0\%$

Berdasarkan hasil perhitungan dari seluruh skenario pengujian, pengujian aplikasi pada sistem penagihan iuran rumah kost menggunakan metode *blackbox testing* mendapatkan persentase 100% keberhasilan.

4.4.4. Usability Testing

Usability testing dilakukan untuk menilai kemudahan pengguna dan pemahaman sistem oleh pengguna. Pengujian ini mengharuskan pengguna menyelesaikan daftar tugas yang telah ditetapkan dalam menggunakan aplikasi. Harapan nantinya adalah hasil dari pengujian ini dapat memberikan pandangan penting untuk pengembangan aplikasi lebih lanjut dan penyempurnaan aplikasi.

4.4.4.1. Melakukan Pengujian Fitur Login

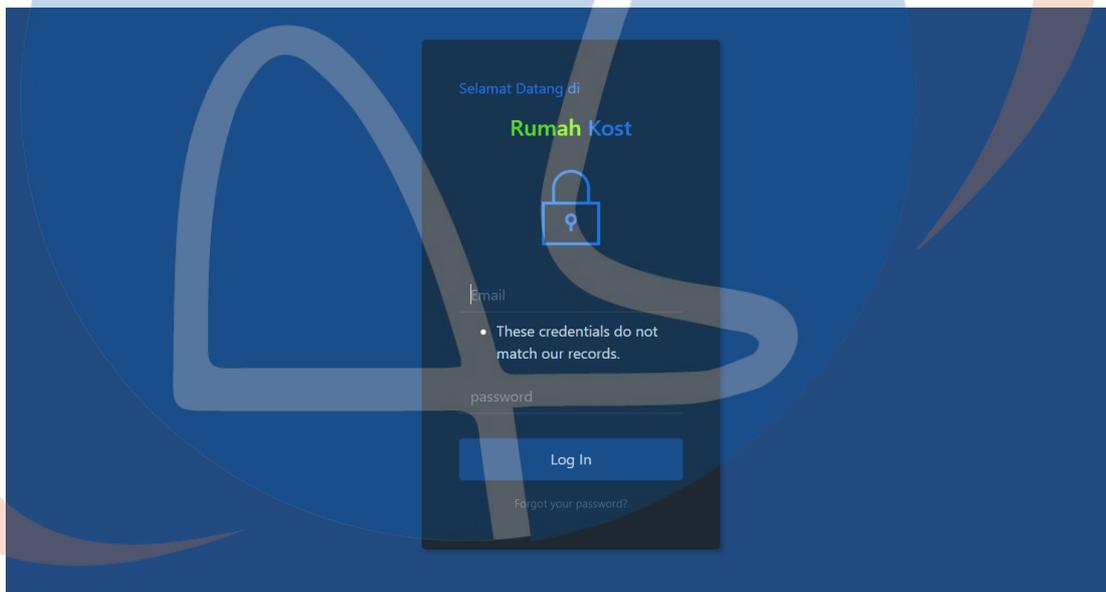
Tahap ini peneliti melakukan pengujian fitur *login* menggunakan metode *usability testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 40 Hasil Pengujian Usability Testing - Login

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|------------------|---------|--|---|-----------------|
| 1 | Login | Admin 1 | Menguji proses login dengan data benar | User berhasil login dan melihat dashboard | Lulus |
| 2 | Login | Admin 2 | Menguji proses login dengan data benar | User berhasil login dan melihat dashboard | Lulus |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|-------------|--|---|--------------|
| 3 | Login | Pengelola 1 | Menguji proses login dengan data benar | User berhasil login dan melihat dashboard | Lulus |
| 4 | Login | Pengelola 2 | Menguji proses login dengan data benar | User berhasil login dan melihat dashboard | Lulus |
| 5 | Login | Pengelola 3 | Menguji proses login dengan data benar | User berhasil login dan melihat dashboard | Lulus |

Berikut ini adalah gambaran hasil dari pengujian proses fitur *login* pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 50 Hasil Pengujian Usability Testing - Login

STT - NF

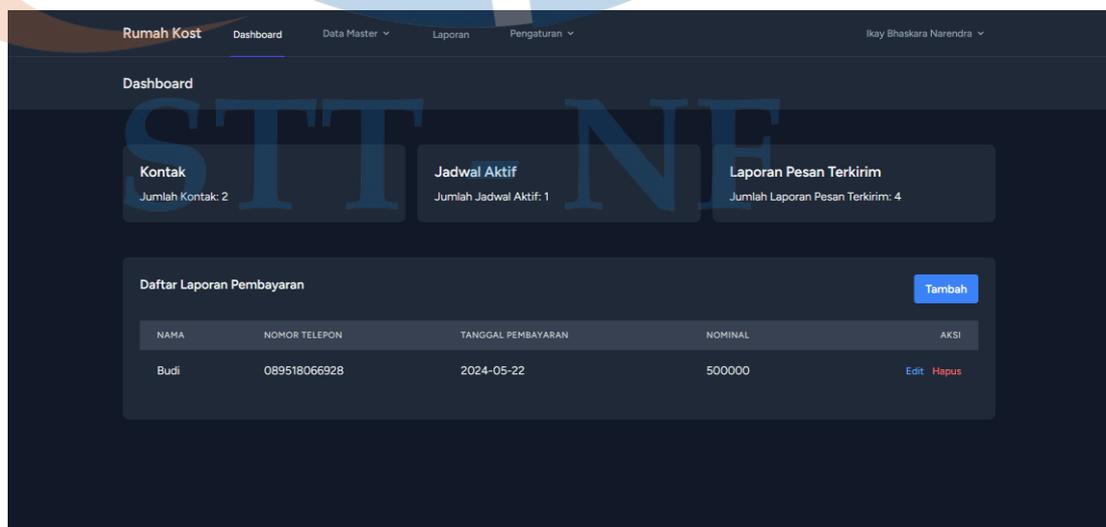
4.4.4.3. Melakukan Pengujian Mengakses *Dashboard*

Tahap ini peneliti melakukan pengujian mengakses dashboard menggunakan metode *usability testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 41 Hasil Pengujian Usability Testing - Mengakses Dashboard

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|---------------------|-------------|------------------------------------|--------------------------------|-----------------|
| 1 | Mengakses Dashboard | Admin 1 | Menguji akses ke halaman dashboard | Dashboard berhasil ditampilkan | Lulus |
| 2 | Mengakses Dashboard | Admin 2 | Menguji akses ke halaman dashboard | Dashboard berhasil ditampilkan | Lulus |
| 3 | Mengakses Dashboard | Pengelola 1 | Menguji akses ke halaman dashboard | Dashboard berhasil ditampilkan | Lulus |
| 4 | Mengakses Dashboard | Pengelola 2 | Menguji akses ke halaman dashboard | Dashboard berhasil ditampilkan | Lulus |
| 5 | Mengakses Dashboard | Pengelola 3 | Menguji akses ke halaman dashboard | Dashboard berhasil ditampilkan | Lulus |

Berikut ini adalah gambaran hasil dari pengujian proses mengakses dashboard pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 51 Hasil Pengujian Usability Testing - Mengakses Dashboard

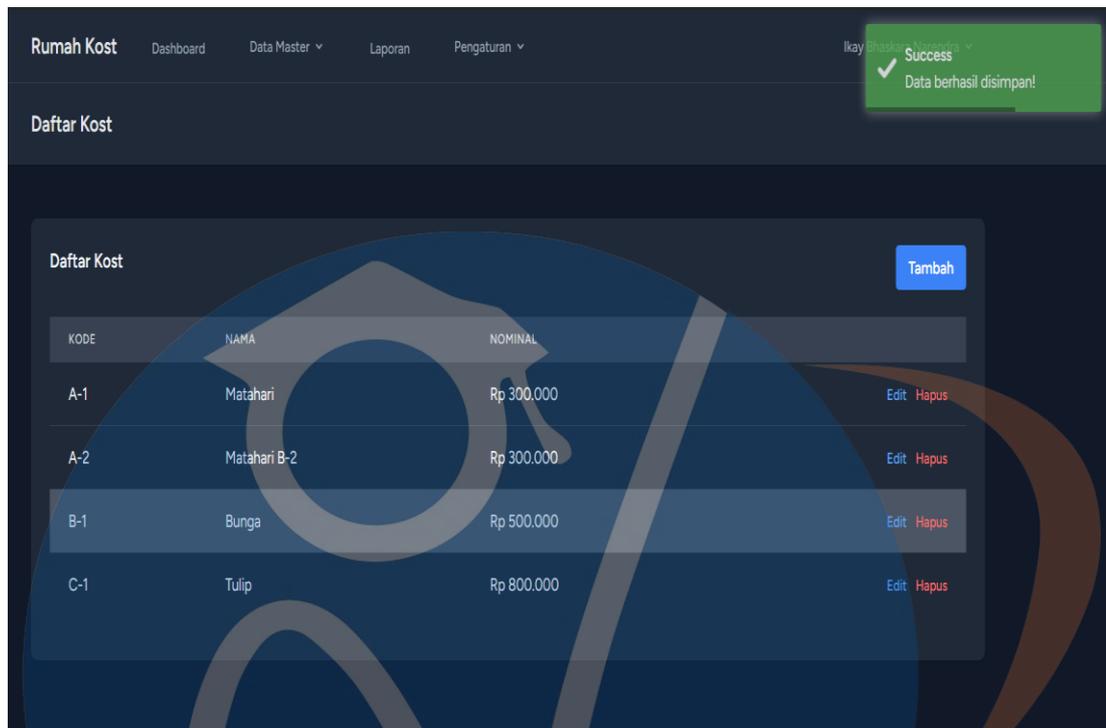
4.4.4.4. Melakukan Pengujian Mengelola Kost

Tahap ini peneliti melakukan pengujian mengelola kost menggunakan metode *usability testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 42 Hasil Pengujian Usability Testing - Mengelola Kost

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|------------------|-------------|---|---|-----------------|
| 1 | Mengelola Kost | Admin 1 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Lulus |
| 2 | Mengelola Kost | Admin 2 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Lulus |
| 3 | Mengelola Kost | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Lulus |
| 4 | Mengelola Kost | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Lulus |
| 5 | Mengelola Kost | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus kost | Data kost berhasil ditambahkan, diubah, dihapus | Lulus |

Berikut ini adalah gambaran hasil dari pengujian proses mengelola kost pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 52 Hasil Pengujian Usability Testing - Mengelola Kost

4.4.4.5. Melakukan Pengujian Mengelola Kontak

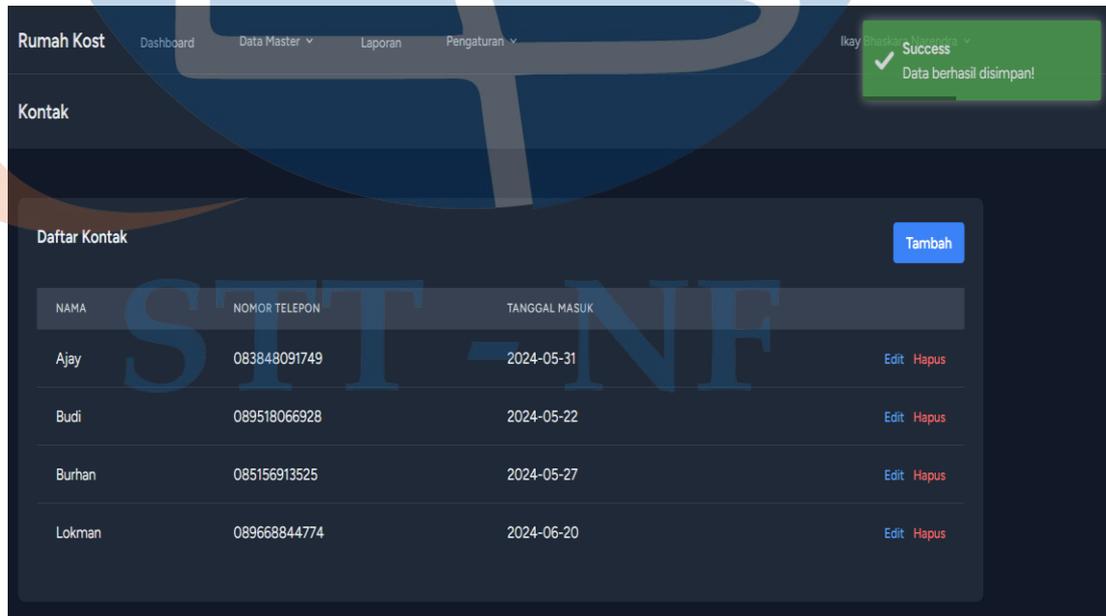
Tahap ini peneliti melakukan pengujian mengelola kontak menggunakan metode *usability testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 43 Hasil Pengujian Usability Testing - Mengelola Kontak

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|------------------|---------|---|---|-----------------|
| 1 | Mengelola Kontak | Admin 1 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Lulus |
| 2 | Mengelola Kontak | Admin 2 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Lulus |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|------------------|-------------|---|---|--------------|
| 3 | Mengelola Kontak | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Lulus |
| 4 | Mengelola Kontak | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Lulus |
| 5 | Mengelola Kontak | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus kontak | Data kontak berhasil ditambahkan, diubah, dihapus | Lulus |

Berikut ini adalah gambaran hasil dari pengujian proses mengelola kontak pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 53 Hasil Pengujian Usability Testing - Mengelola Kontak

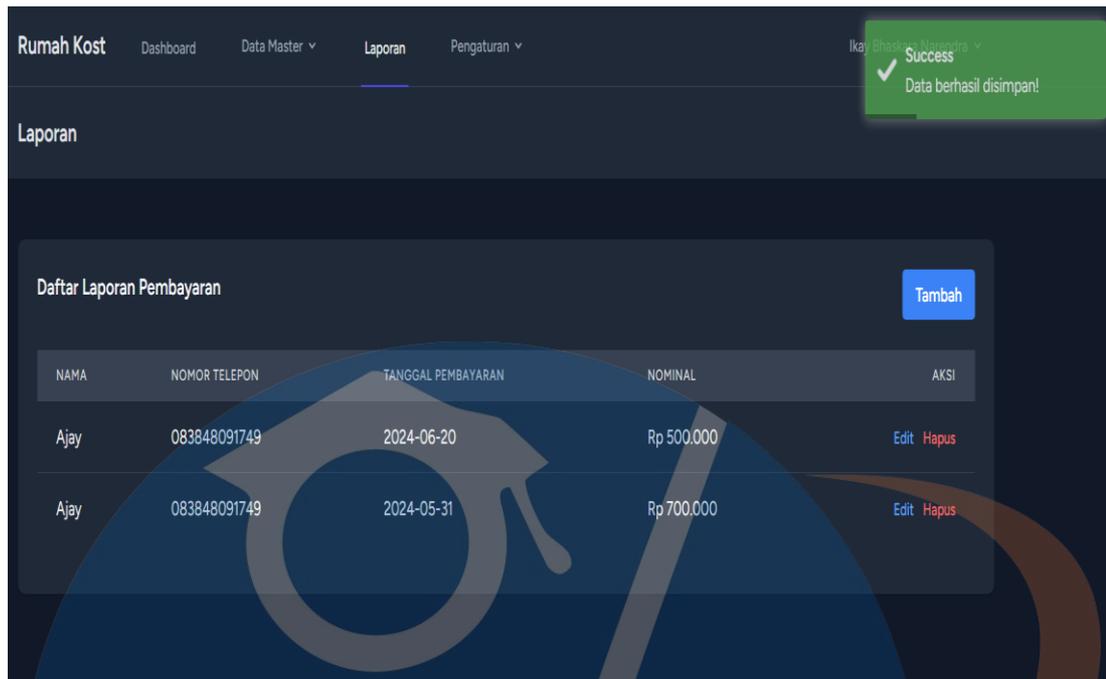
4.4.4.7. Melakukan Pengujian Mengelola Laporan

Tahap ini peneliti melakukan pengujian mengelola laporan menggunakan metode *usability testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 44 Hasil Pengujian Usability Testing - Mengelola Laporan

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|-------------------|-------------|--|---|-----------------|
| 1 | Mengelola Laporan | Admin 1 | Menguji proses menambah, mengubah, dan menghapus laporan | Laporan berhasil ditambahkan, diubah, dihapus | Lulus |
| 2 | Mengelola Laporan | Admin 2 | Menguji proses menambah, mengubah, dan menghapus laporan | Laporan berhasil ditambahkan, diubah, dihapus | Lulus |
| 3 | Mengelola Laporan | Pengelola 1 | Menguji proses menambah, mengubah, dan menghapus laporan | Laporan berhasil ditambahkan, diubah, dihapus | Lulus |
| 4 | Mengelola Laporan | Pengelola 2 | Menguji proses menambah, mengubah, dan menghapus laporan | Laporan berhasil ditambahkan, diubah, dihapus | Lulus |
| 5 | Mengelola Laporan | Pengelola 3 | Menguji proses menambah, mengubah, dan menghapus laporan | Laporan berhasil ditambahkan, diubah, dihapus | Lulus |

Berikut ini adalah gambaran hasil dari pengujian proses mengelola laporan pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 54 Hasil Pengujian Usability Testing - Mengelola Laporan

4.4.4.8. Melakukan Pengujian Mengelola Perangkat

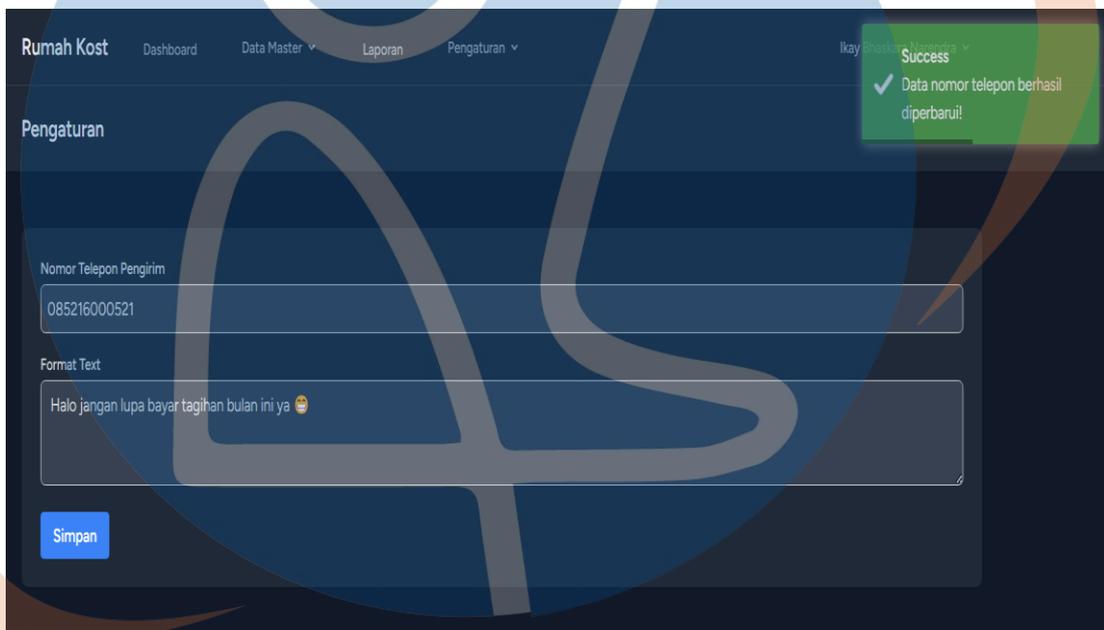
Tahap ini peneliti melakukan pengujian mengelola perangkat menggunakan metode *usability testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 45 Hasil Pengujian Usability Testing - Mengelola Perangkat

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|---------------------|-------------|---|---|-----------------|
| 1 | Mengelola perangkat | Admin 1 | Menguji proses merubah formulir telepon dan format teks | Formulir telepon dan teks berhasil diubah | Lulus |
| 2 | Mengelola perangkat | Admin 2 | Menguji proses merubah formulir telepon dan format teks | Formulir telepon dan teks berhasil diubah | Lulus |
| 3 | Mengelola perangkat | Pengelola 1 | Menguji proses merubah formulir format teks | Formulir teks berhasil diubah | Lulus |

| No | Fitur yang diuji | User | Kondisi Prasyarat / | Hasil yang diharapkan | Keberhasilan |
|----|---------------------|-------------|---|-------------------------------|--------------|
| 4 | Mengelola perangkat | Pengelola 2 | Menguji proses merubah formulir format teks | Formulir teks berhasil diubah | Lulus |
| 5 | Mengelola perangkat | Pengelola 3 | Menguji proses merubah formulir format teks | Formulir teks berhasil diubah | Lulus |

Berikut ini adalah gambaran hasil dari pengujian proses mengelola perangkat pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 55 Hasil Pengujian Usability Testing - Mengelola Perangkat

STT - NF

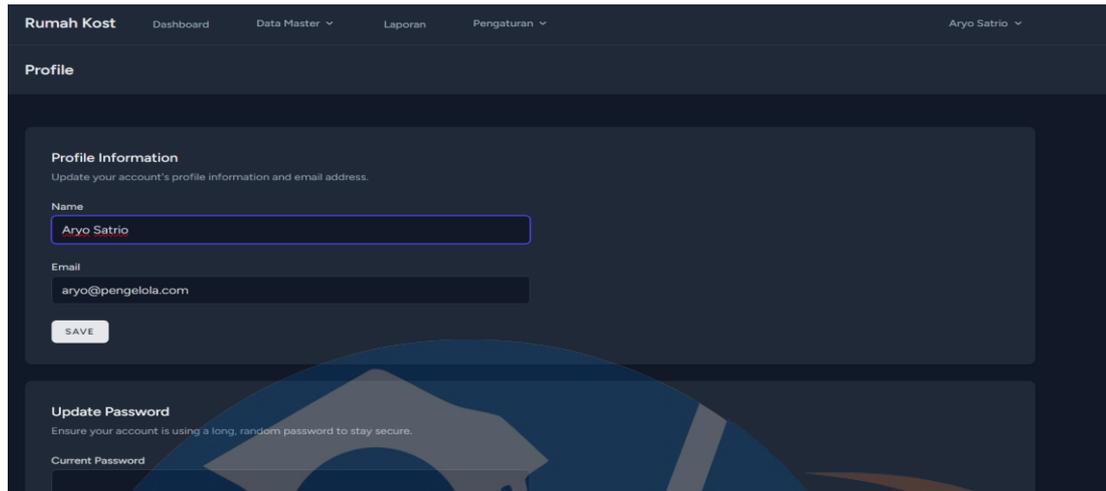
4.4.4.10. Melakukan Pengujian Mengubah *Profile*

Tahap ini peneliti melakukan pengujian mengubah *profile* menggunakan metode *usability testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 46 Hasil Pengujian Usability Testing - Mengubah *Profile*

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|-------------------------------|-------------|--|-----------------------------------|-----------------|
| 1 | Melakukan <i>Edit Profile</i> | Admin 1 | Menguji proses merubah informasi pribadi | Informasi pribadi berhasil diubah | Lulus |
| 2 | Melakukan <i>Edit Profile</i> | Admin 2 | Menguji proses merubah informasi pribadi | Informasi pribadi berhasil diubah | Lulus |
| 3 | Melakukan <i>Edit Profile</i> | Pengelola 1 | Menguji proses merubah informasi pribadi | Informasi pribadi berhasil diubah | Lulus |
| 4 | Melakukan <i>Edit Profile</i> | Pengelola 2 | Menguji proses merubah informasi pribadi | Informasi pribadi berhasil diubah | Lulus |
| 5 | Melakukan <i>Edit Profile</i> | Pengelola 3 | Menguji proses merubah informasi pribadi | Informasi pribadi berhasil diubah | Lulus |

Berikut ini adalah gambaran hasil dari pengujian proses mengubah *profile* pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 56 Hasil Pengujian Usability Testing - Mengubah Profile

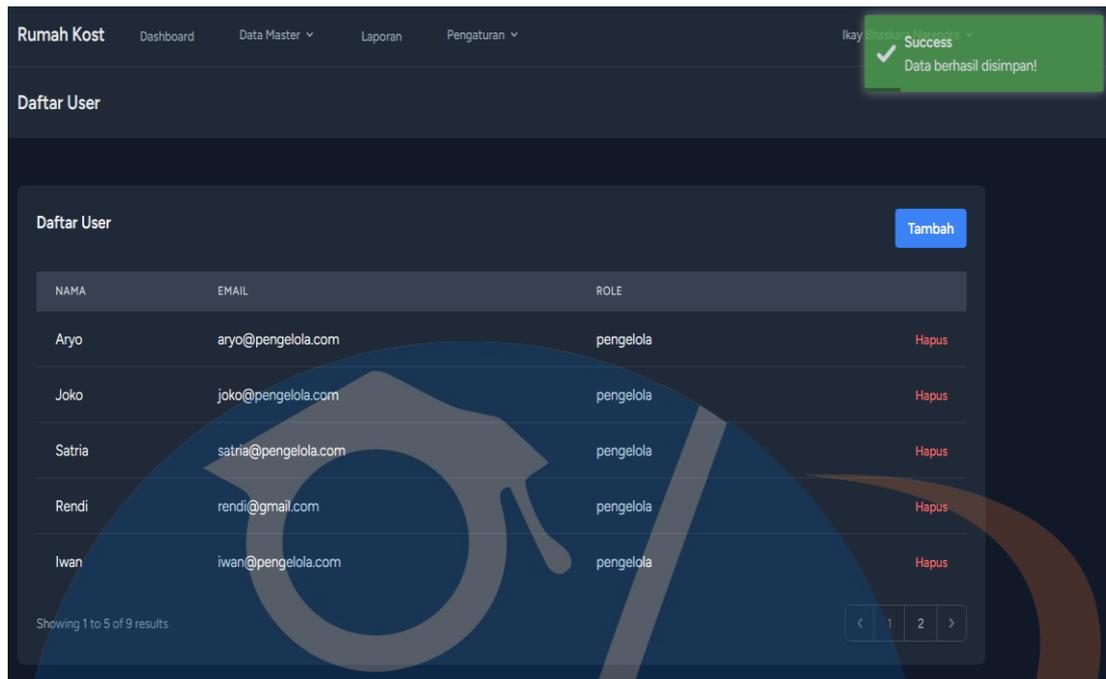
4.4.4.11. Melakukan Pengujian Mengelola User

Tahap ini peneliti melakukan pengujian mengelola *user* menggunakan metode *usability testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 47 Hasil Pengujian Usability Testing - Mengelola User

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|------------------|---------|--|------------------------------------|-----------------|
| 1 | Mengelola user | Admin 1 | Menguji proses menambah dan menghapus user | User berhasil ditambahkan, dihapus | Lulus |
| 2 | Mengelola user | Admin 2 | Menguji proses menambah dan menghapus user | User berhasil ditambahkan, dihapus | Lulus |

Berikut ini adalah gambaran hasil dari pengujian proses mengelola *user* pada aplikasi sistem penagihan iuran rumah kost:



Gambar 4. 57 Hasil Pengujian Usability Testing - Mengelola User

4.4.4.12. Melakukan Pengujian Melihat Laporan Pengiriman Pesan

. Tahap ini peneliti melakukan pengujian melihat laporan pengiriman pesan menggunakan metode *usability testing*. Berikut adalah hasil dari pengujian tersebut:

Tabel 4. 48 Hasil Pengujian Usability Testing – Melihat Status Pengiriman Pesan

| No | Fitur yang diuji | User | Deskripsi pengujian | Kriteria keberhasilan | Hasil pengujian |
|----|-----------------------------------|-------------|---|--|-----------------|
| 1 | Memeriksa status pengiriman pesan | Admin 1 | Menguji pemeriksaan status pengiriman pesan di menu laporan | Status pengiriman pesan berhasil diperiksa | Lulus |
| 2 | Memeriksa status pengiriman pesan | Admin 2 | Menguji pemeriksaan status pengiriman pesan di menu laporan | Status pengiriman pesan berhasil diperiksa | Lulus |
| 3 | Memeriksa status pengiriman pesan | Pengelola 1 | Menguji pemeriksaan status pengiriman pesan di menu laporan | Status pengiriman pesan berhasil diperiksa | Lulus |

| No | Fitur yang diuji | User | Kondisi / Prasyarat | Hasil yang diharapkan | Keberhasilan |
|----|-----------------------------------|-------------|---|--|--------------|
| 4 | Memeriksa status pengiriman pesan | Pengelola 2 | Menguji pemeriksaan status pengiriman pesan di menu laporan | Status pengiriman pesan berhasil diperiksa | Lulus |
| 5 | Memeriksa status pengiriman pesan | Pengelola 3 | Menguji pemeriksaan status pengiriman pesan di menu laporan | Status pengiriman pesan berhasil diperiksa | Lulus |

Berikut ini adalah gambaran hasil dari pengujian proses melihat laporan pengiriman pesan pada aplikasi sistem penagihan iuran rumah kost:

| Daftar Laporan Pesan Terkirim | | | |
|-------------------------------|---------------|------------------|----------|
| NAMA | NOMOR TELEPON | TANGGAL TERKIRIM | STATUS |
| Budi | 089518066928 | 2024-06-20 | terkirim |
| Budi | 089518066928 | 2024-06-20 | terkirim |
| Budi | 089518066928 | 2024-06-10 | terkirim |
| Budi | 089518066928 | 2024-06-04 | terkirim |
| Budi | 089518066928 | 2024-06-03 | terkirim |

Showing 1 to 5 of 64 results

Gambar 4. 58 Hasil Pengujian Usability Testing – Melihat Status Pengiriman Pesan

4.4.5. Menghitung Hasil Pengujian Usability Testing

Pada pengujian *usability testing* terdiri dari 9 skenario pengujian dengan total keseluruhan yaitu 42 pengujian. Dari total pengujian yang telah dilakukan, terdapat 42 lulus pengujian dan tidak ada pengujian yang tidak lulus. Sehingga persentase keberhasilan dari pengujiannya yaitu 100%. Dengan perhitungan persentase keberhasilan sebagai berikut:

1. Melakukan Pengujian Fitur *Login*

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

2. Melakukan Pengujian Mengakses *Dashboard*

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

3. Melakukan Pengujian Mengelola Kost

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

4. Melakukan Pengujian Mengelola Kontak

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

5. Melakukan Pengujian Mengelola Laporan

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

6. Melakukan Pengujian Mengelola Perangkat

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

7. Melakukan Pengujian Mengubah *Profile*

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$

8. Melakukan Pengujian Mengelola *User*

- a. Total Pengujian: 2 Pengujian
- b. Lulus : $(2/2) \times 100 = 100\%$
- c. Tidak Lulus : $(0/2) \times 100 = 0\%$

9. Melakukan Pengujian Melihat Laporan Pengiriman Pesan

- a. Total Pengujian: 5 Pengujian
- b. Lulus : $(5/5) \times 100 = 100\%$
- c. Tidak Lulus : $(0/5) \times 100 = 0\%$



STT - NF

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian, pembahasan, dan uji coba yang telah dilakukan, dapat disimpulkan sebagai berikut:

1. Berdasarkan penelitian yang telah dilakukan mengenai perancangan sistem penagihan iuran rumah kost menggunakan *whatsapp API*, dapat disimpulkan bahwa penelitian ini berhasil merancang dan mengimplementasikan sistem penagihan yang efektif dan terstruktur. Sistem ini dirancang dengan menggunakan *framework laravel* dan memanfaatkan berbagai alat bantu pemodelan seperti *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Dalam penelitian ini, implementasi sistem penagihan iuran rumah kost memanfaatkan layanan pihak ketiga untuk menghubungkan *whatsapp API* dengan *website* penagihan iuran rumah kost. Layanan pihak ketiga berperan sebagai perantara dalam pengiriman pesan kepada penghuni rumah kost.
2. Pengujian *blackbox testing* dan *usability testing* telah dilakukan untuk memastikan bahwa semua fungsi sistem bekerja dengan baik dan dapat digunakan oleh pengguna. Berdasarkan hasil pengujian dalam penelitian ini seluruh pengujian mendapatkan persentase 100% dan terbukti efektif, dengan seluruh fungsi bekerja dengan baik dan dapat digunakan oleh pengguna guna memastikan efektivitas dalam penggunaan sistem penagihan iuran rumah kost.

5.2. Saran

Sistem penagihan iuran rumah kost ini masih mempunyai banyak kekurangan yang perlu dikembangkan dan diperbaiki. Sehingga perlu dilakukan pengembangan lebih lanjut yang sesuai dengan perkembangan teknologi dimasa yang akan datang. Beberapa saran yang disarankan oleh penulis antara lain sebagai berikut:

1. Aplikasi ini diharapkan dapat digunakan melalui android sehingga lebih mudah digunakan dimana saja.
2. Tidak ada fitur pencarian didalam aplikasi ini sehingga tidak cocok dalam penggunaan skala besar.

DAFTAR REFERENSI

- [1] Suminten, I. Dewi Sintawati, and W. Indrarti, “Perancangan Sistem Informasi Sewa Rumah Kost Melalui Aplikasi Berbasis Web,” *IJCCS*, vol. Vol. 17, No. 2, pp. 1–12, 2023.
- [2] E. Giovana Asti, “Analisis Konseptual Faktor-Faktor Penggunaan Aplikasi WhatsApp pada Manajemen Pendidikan,” 2023.
- [3] A. G. Gani, P. Furtuna Dewi, and A. Sugiharto, “SISTEM INFORMASI POINT OF SALE BERBASIS WEB PADA DAPUR CARINGIN TILU BANDUNG.”
- [4] Y. Wahyudin and D. N. Rahayu, “Analisis Metode Pengembangan Sistem Informasi Berbasis Website: A Literatur Review,” *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 15, no. 3, pp. 26–40, Oct. 2020, doi: 10.35969/interkom.v15i3.74.
- [5] S. Utomo and M. A. Hamdani, “SISTEM INFORMASI GEOGRAFIS (SIG) PARIWISATA KOTA BANDUNG MENGGUNAKAN GOOGLE MAPS API DAN PHP,” 2021. [Online]. Available: <http://jurnal.unnur.ac.id/index.php/jurnalfiki>
- [6] D. Purnama Sari, R. Wijanarko, and J. X. Menoreh Tengah, “Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus Di Rumah Kamera Semarang),” vol. 2, no. 1, pp. 32–36, 2019.
- [7] D. Syifani and A. Dores, “APLIKASI SISTEM REKAM MEDIS DI PUSKESMAS KELURAHAN GUNUNG,” *Teknologi Informatika dan Komputer*, vol. 9, no. 1, 2018.
- [8] E. Zuliarso and H. Februariyanti, “Pemanfaatan Instant Messaging untuk Aplikasi Layanan Akademik,” *Jurnal Teknologi Informasi DINAMIK*, vol. 18, no. 2, pp. 112–121, 2013.
- [9] E. Julio, M. A. I. Pakereng, and I. Artikel, “Implementasi API Payment Gateway Menggunakan Arsitektur Microservice,” *JURNAL INFORMATIKA*, vol. 8, no. 2, 2021, [Online]. Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>

- [10] A. Izzah *et al.*, “PENGEMBANGAN WEB COMPANY PROFILE TERINTEGRASI DENGAN API WHATSAPP (STUDI KASUS: AGEN SEMBAKO AL-BARCAH)”, [Online]. Available: <https://api.whatsapp.com/send?ph>
- [11] Y. Nugraha, “Information System Development With Comparison of Waterfall and Prototyping Models,” 2020.
- [12] Yohanes Mada Masa, Emerensiana Ngaga, Yovinia Carmeneja Hoar Siki, and Donatus Joseph Manehat, “RANCANG BANGUN SISTEM INFORMASI RUMAH KOST DI KOTA KUPANG BERBASIS WEB,” *Jurnal In Create (Inovasi dan Kreasi dalam Teknologi Informasi)*, 2023.
- [13] S. Aditia, M. Najamuddin Dwi Miharja, and A. Aguswin, “IMPLEMENTASI SISTEM KEHADIRAN PRAKTIKUM BERBASIS QRCODE DENGAN WHATSAPP GATEWAY MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT (RAD),” *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, vol. 9, pp. 82–88, 2023.
- [14] Y. D. Arimbi, D. Kartinah, A. Nila, and W. Della, “RANCANGAN SISTEM INFORMASI KOST PUTRI MALIKA BERBASIS WEBSITE MENGGUNAKAN FRAMEWORK LARAVEL DAN MYSQL,” vol. 1, [Online]. Available: <https://www.teamstart.my.id/>.
- [15] A. Mike, G. Hutauruk, B. Parga Zen, and A. Utami, “Penerapan Metode Agile Pada Website Indekost Sruntul Menggunakan Framework Laravel,” *Jurnal Ilmiah MEDIA SISFO*, vol. 17, no. 2, 2023, doi: 10.33998/mediasisfo.2023.1.2.1370.
- [16] Okpatrioka, “Research And Development (R&D) Penelitian Yang Inovatif Dalam Pendidikan,” *DHARMA ACARIYA NUSANTARA : Jurnal Pendidikan, Bahasa dan Budaya*, vol. 1, pp. 86–100, 2023.
- [17] M. Rijal Fadli, “Memahami desain metode penelitian kualitatif,” vol. 21, no. 1, pp. 33–54, 2021, doi: 10.21831/hum.v21i1.
- [18] Y. Dwi Wijaya and M. Wardah Astuti, “PENGUJIAN BLACKBOX SISTEM INFORMASI PENILAIAN KINERJA KARYAWAN PT INKA (PERSERO) BERBASIS EQUIVALENCE PARTITIONS BLACKBOX TESTING OF PT INKA (PERSERO) EMPLOYEE PERFORMANCE

ASSESSMENT INFORMATION SYSTEM BASED ON EQUIVALENC
PARTITIONS,” *Jurnal Digital Teknologi Informasi*, vol. 4, p. 2021.

- [19] M. T. Abdillah *et al.*, “Implementasi Black box Testing dan Usability Testing pada Website Sekolah MI Miftahul Ulum Warugunung Surabaya,” *Jurnal Ilmu Komputer dan Desain Komunikasi Visual*, vol. 8, no. 1, 2023.



STT - NF

LAMPIRAN

A1. Composer

- Langkah *Install Composer* : <https://santrikoding.com/tutorial-laravel-10-2-install-dan-menjalankan-laravel-10>

A2. Laragon

- Langkah *Install Laragon* : <https://www.hops.id/hobi/pr-2942596598/cara-install-laragon-di-windows-10-pemula-wajib-tahu>

A3. Otentikasi Laravel

- Langkah *Implementasi* : <https://medium.com/@akhmadshaleh/instalasi-laravel-breeze-53fa200dfdcc>

A4. Penjelasan Traits

- Tentang *Traits* : <https://wahyuivan.medium.com/hindari-menulis-kode-berulang-kali-di-laravel-dengan-menggunakan-trait-92fb1710b326>

A5. Penjelasan CronJob

- Tentang *CronJob* : <https://www.jagoanhosting.com/blog/laravel-scheduler/>

A6. Source Code

- *Application* : https://github.com/rizkiarch/rumah_kost

A7. Video demo

- *Youtube* : <https://www.youtube.com/watch?v=whR4LaSVDR8>

A.8. Lampiran Hasil Wawancara

- *Spreadsheets* :
<https://docs.google.com/spreadsheets/d/1ou3l6f32xym6MReN1b14DhJONL17xGvakzYv14jRmmE/edit?usp=sharing>

STT - NF