

**LAPORAN PRAKTIK KERJA LAPANGAN**  
**SISTEM INFORMASI PEMBAYARAN DAN MANAJEMEN INVOICE**  
**RUMAH SUSUN KOTA SEMARANG**



**Disusun oleh :**  
**Labiba Adinda Zahwana**  
**24060121140111**

**DEPARTEMEN INFORMATIKA**  
**FAKULTAS SAINS DAN MATEMATIKA**  
**UNIVERSITAS DIPONEGORO**

**2025**

## HALAMAN PENGESAHAN

Nama : Labiba Adinda Zahwana

NIM : 24060121140111

Judul PKL : Sistem Informasi Pembayaran dan Manajemen Invoice Rumah Susun Kota  
Semarang

Telah diseminarkan dan dinyatakan lulus pada tanggal 06 Januari 2025

Semarang, 06 Januari 2025

Menyetujui,

Koordinator PKL



Sandy Kurniawan, S.Kom., M.Kom.

NIP. 199603032024061003

Dosen Pembimbing,



Prajanto Wahyu Adi, M.Kom

NIP. 198803222020121010

Mengetahui,

Ketua Departemen Informatika



Dr. Aris Sugiharto S.Si., M.Kom.

NIP. 197108111997021004

## ABSTRAK

Aplikasi **SIPARUS** adalah platform berbasis web yang dikembangkan untuk menghadapi tantangan dalam pengelolaan sistem pembayaran dan manajemen invoice di Rumah Susun Kota Semarang. Aplikasi ini dirancang untuk pengelolaan data pembayaran serta invoice penghuni yang sebelumnya tidak terintegrasi secara optimal. Pengembangan aplikasi ini menggunakan metode *Extreme Programming*, dengan penerapan teknologi terkini seperti *framework* Next.js 14 untuk *frontend*, Laravel 11 untuk *backend*, dan MySQL sebagai basis data, sehingga menawarkan kecepatan, keamanan, dan efisiensi dalam pengelolaan data penyewa dan pemilik rusun. Platform ini menyediakan dua peran pengguna, yaitu penyewa yang mencari dan memesan unit, serta pemilik yang dapat mengelola rusun secara mandiri.

*Keywords:* *Extreme Programming, Web-based Application, Payment Management*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah Subahanahu Wata'ala, yang atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan Praktik Kerja Lapangan dengan judul "Sistem Informasi Pembayaran dan Manajemen Invoice Rumah Susun Kota Semarang" ini dengan baik. Laporan ini disusun sebagai hasil dari pengalaman penulis selama melaksanakan kegiatan Praktik Kerja Lapangan (PKL) di Dinas Komunikasi, Informatika, Statistik, dan Persandian Kota Semarang dilaksanakan mulai dari 02 Januari 2024 hingga 02 Maret 2024.

Penulis menyadari bahwa penyusunan laporan ini tidak lepas dari dukungan, semangat, dan bimbingan dari berbagai pihak. Bersama ini, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Dr. Aris Sugiharto S.Si., M.Kom., selaku Ketua Departemen Ilmu Komputer/Informatika yang telah membantu dalam proses pengarahan PKL.
2. Bapak Prajanto Wahyu Adi, M.Kom selaku Dosen Pembimbing PKL yang telah memberikan arahan, pengetahuan, dan bimbingan yang berharga sepanjang penyusunan laporan ini.
3. Bapak Sandy Kurniawan, S.Kom., M.Kom., selaku Koordinator PKL yang telah mendukung keberjalanan kegiatan PKL.
4. Kedua orang tua penulis yang senantiasa memberikan dukungan kepada penulis.
5. Seluruh pihak yang telah membantu kelancaran dalam pelaksanaan kegiatan PKL.

Terakhir, penulis menyadari bahwa laporan ini belum sempurna, dan penulis selalu terbuka untuk masukan dan saran yang membangun. Semua saran yang diberikan akan sangat penulis hargai untuk perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat bagi semua pihak.

Semarang, 01 Januari 2025

Labiba Adinda Zahwana

## DAFTAR ISI

<b>HALAMAN PENGESAHAN .....</b>	<b>ii</b>
<b>ABSTRAK.....</b>	<b>iii</b>
<b>KATA PENGANTAR .....</b>	<b>iv</b>
<b>DAFTAR ISI.....</b>	<b>v</b>
<b>DAFTAR GAMBAR .....</b>	<b>vii</b>
<b>DAFTAR TABEL.....</b>	<b>ix</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Rumusan Masalah .....	2
1.3    Tujuan.....	3
1.4    Manfaat.....	3
1.5    Ruang Lingkup .....	3
1.6    Sistematika Penulisan.....	4
<b>BAB II TINJAUAN PERUSAHAAN .....</b>	<b>7</b>
2.1    Profil Instansi .....	7
2.2    Visi Instansi .....	7
2.3    Misi Instansi .....	8
2.4    Tugas dan Fungsi Diskominfo Kota Semarang.....	8
2.5    Struktur Organisasi Instansi .....	9
<b>BAB III LANDASAN TEORI.....</b>	<b>11</b>
3.1    Aplikasi Berbasis Web .....	11
3.2    Metode Pengembangan Perangkat Lunak <i>Agile</i> .....	11
3.3 <i>Extreme Programming</i> .....	12
3.4 <i>Unified Modelling Language (UML)</i> .....	12
3.5 <i>Use Case Diagram</i> .....	13
3.6 <i>Sequence Diagram</i> .....	14
3.7 <i>Class Diagram</i> .....	15
3.8    Bahasa Pemrograman dan Database.....	16
3.8.1 <i>Framework laravel</i> .....	16
3.8.2 <i>Framework Next.js</i> .....	16
3.8.3    Database MySQL & Phpmyadmin.....	17
3.8.4    Bahasa JavaScript.....	17
3.8.5    Bahasa PHP .....	18

3.8.6	Bahasa HTML & CSS .....	18
3.8.7	<i>Black Box Testing</i> .....	19
3.8.2	<i>White Box Testing</i> .....	19
<b>BAB IV ANALISIS &amp; RANCANGAN.....</b>		<b>20</b>
4.1	Analisis Kebutuhan .....	20
4.1.1	Deskripsi Umum Aplikasi SIPARUS .....	20
4.1.2	Karakteristik Pengguna .....	20
4.1.3	Kebutuhan Perangkat Lunak .....	21
4.1.4	Kebutuhan Fungsional.....	21
4.1.5	Kebutuhan Non Fungsional .....	23
4.2	Desain Aplikasi .....	23
4.2.1	<i>Diagram Use Case</i> .....	24
4.2.2	Diagram <i>Sequence</i> .....	24
4.2.3	Diagram <i>Class</i> .....	42
4.2.4	Perancangan Antarmuka.....	43
<b>BAB V HASIL DAN PEMBAHASAN .....</b>		<b>54</b>
5.1	Implementasi .....	54
5.1.1	Implementasi Sistem .....	54
5.1.2	Implementasi Database.....	54
5.1.3	Implementasi Antarmuka .....	55
5.2	Pengujian .....	80
5.2.1	Persiapan Prosedural .....	81
5.2.2	Perencanaan Pengujian.....	82
5.2.3	Pelaksanaan Pengujian .....	82
<b>BAB VI PENUTUP .....</b>		<b>90</b>
6.1	Kesimpulan.....	90
6.2	Saran.....	90
<b>DAFTAR PUSTAKA .....</b>		<b>91</b>
<b>LAMPIRAN-LAMPIRAN.....</b>		<b>94</b>

## DAFTAR GAMBAR

Gambar 2. 1 Struktur Organisasi Dikominfo Kota Semarang.....	10
Gambar 3. 1 Contoh Sequence Diagram .....	14
Gambar 3. 2 Contoh Class Diagram.....	15
Gambar 4.2.1. 1 Use Case SIPARUS.....	24
Gambar 4.2.2. 1 Sequence Diagram Login atau Sign In oleh user.....	25
Gambar 4.2.2. 2 Sequence Diagram Login atau Sign In oleh admin .....	26
Gambar 4.2.2. 3 Sequence Diagram Logout oleh user .....	27
Gambar 4.2.2. 4 Sequence Diagram Logout oleh admin.....	28
Gambar 4.2.2. 5 Sequence Diagram register user .....	29
Gambar 4.2.2. 6 Sequence Diagram Lihat List Rusun oleh admin .....	30
Gambar 4.2.2. 7 Sequence Diagram Lihat List Rusun oleh User.....	31
Gambar 4.2.2. 8 Sequence Diagram Lihat Detail Rusun oleh user .....	32
Gambar 4.2.2. 9 Sequence Diagram Lihat Detail Rusun oleh admin.....	33
Gambar 4.2.2. 10 Sequence Diagram Lihat Daftar Transaksi oleh user .....	34
Gambar 4.2.2. 11 Sequence Diagram Lihat Daftar Transaksi oleh admin .....	35
Gambar 4.2.2. 12 Sequence Diagram Hapus Transaksi oleh admin .....	36
Gambar 4.2.2. 13 Sequence Diagram Tambah Rusun oleh admin.....	37
Gambar 4.2.2. 14 Sequence Diagram Hapus Rusun oleh admin.....	38
Gambar 4.2.2. 15 . Sequence Diagram Booking Kamar .....	39
Gambar 4.2.2. 16 Sequence Diagram Beri Bukti Pembayaran .....	40
Gambar 4.2.2. 17 Sequence Diagram Verifikasi Pembookingan Kamar oleh admin .....	41
Gambar 4.2.3. 1 Diagram Class .....	42
Gambar 4.2.4.1. 1 Wireframe antarmuka Sign Up.....	44
Gambar 4.2.4.1. 2 Wireframe antarmuka Sign In .....	44
Gambar 4.2.4.1. 3 Wireframe antarmuka Logged In.....	45
Gambar 4.2.4.1. 4 Wireframe antarmuka Landing Page .....	46
Gambar 4.2.4.1. 5 Wireframe antarmuka Detail Rusun .....	47
Gambar 4.2.4.1. 6 Wireframe antarmuka Checkout.....	48
Gambar 4.2.4.1. 7 Wireframe antarmuka Booking Sukses .....	49
Gambar 4.2.4.1. 8 Wireframe antarmuka Dashboard Overview .....	49
Gambar 4.2.4.1. 9 Wireframe antarmuka Transactions.....	50
Gambar 4.2.4.2. 1 Wireframe antarmuka Sign In .....	50
Gambar 4.2.4.2. 2 Wireframe antarmuka Dashboard.....	51
Gambar 4.2.4.2. 3 Wireframe antarmuka Listings .....	52
Gambar 4.2.4.2. 4 Wireframe antarmuka Create Listings.....	52
Gambar 4.2.4.2. 5 Wireframe antarmuka Edit Listings.....	53
Gambar 4.2.4.2. 6 Wireframe antarmuka Transactions.....	53
Gambar 5.1.2. 1 Tabel users.....	54
Gambar 5.1.2. 2 Tabel transaction.....	55
Gambar 5.1.2. 3 Tabel sessions.....	55
Gambar 5.1.2. 4 Tabel migrations .....	55
Gambar 5.1.2. 5 Tabel listings.....	55
Gambar 5.1.3.1.1. 1 Halaman Landing Page 1.....	56
Gambar 5.1.3.1.1. 2 Halaman Landing Page 2.....	57
Gambar 5.1.3.1.1. 3 Halaman Landing Page 3.....	57
Gambar 5.1.3.1.1. 4 Halaman Landing Page 4.....	58

Gambar 5.1.3.1.2. 1 Halaman Sign Up.....	58
Gambar 5.1.3.1.2. 2 Halaman Sign Up berhasil.....	59
Gambar 5.1.3.1.2. 3 Halaman auto direct dan auto masuk akun setelah berhasil Sign Up .....	59
Gambar 5.1.3.1.2. 4 Halaman Notifikasi Gagal Buat Akun menggunakan Email yang sama .....	60
Gambar 5.1.3.1.3. 1 Halaman Sign In berhasil .....	60
Gambar 5.1.3.1.3. 2 Halaman Signed In .....	61
Gambar 5.1.3.1.3. 3 Halaman Sign In gagal .....	61
Gambar 5.1.3.1.4. 1 Halaman Navbar untuk Logout .....	62
Gambar 5.1.3.1.4. 2 Halaman direct ke Login Page.....	62
Gambar 5.1.3.1.5. 1 Halaman Logged In .....	63
Gambar 5.1.3.1.6. 1 Detail Rusun .....	65
Gambar 5.1.3.1.6. 2 Detail Rusun set booking.....	65
Gambar 5.1.3.1.6. 3 Halaman booking gagal karena belum sign in.....	66
Gambar 5.1.3.1.7. 1 Halaman Checkout .....	67
Gambar 5.1.3.1.8. 1 Halaman Booking Sukses dan menunggu konfirmasi dari admin .....	68
Gambar 5.1.3.1.8. 2 Halaman booking sudah disetujui (approved) .....	68
Gambar 5.1.3.1.9. 1 Halaman Overview .....	69
Gambar 5.1.3.1.10. 1 Halaman dashboard transaksi yang telah dilakukan .....	70
Gambar 5.1.3.2.1. 2 Halaman Sign In admin .....	71
Gambar 5.1.3.2.2. 1 Halaman Landing Page admin.....	72
Gambar 5.1.3.2.2. 2 Halaman Landing Page berhasil approved .....	72
Gambar 5.1.3.2.3. 1 Halaman Navbar Logout .....	73
Gambar 5.1.3.2.3. 2 Halaman Redirect Sign In page .....	73
Gambar 5.1.3.2.4. 1 Halaman Listings .....	74
Gambar 5.1.3.2.4. 2 Halaman tambah listings rusun.....	75
Gambar 5.1.3.2.4. 3 Halaman redirect ke edit listing page dengan pop up Created .....	75
Gambar 5.1.3.2.4. 4 Halaman daftar listings dengan listings yang sudah ditambahkan .....	75
Gambar 5.1.3.2.4. 5 Halaman listings delete listing.....	76
Gambar 5.1.3.2.4. 6 Halaman listings berhasil deleted .....	76
Gambar 5.1.3.2.4. 7 Halaman edit listing.....	77
Gambar 5.1.3.2.4. 8 Halaman berhasil edit listing .....	77
Gambar 5.1.3.2.5. 1 Halaman informasi transaksi .....	78
Gambar 5.1.3.2.5. 2 Halaman pop up approved oleh admin .....	79
Gambar 5.1.3.2.5. 3 Halaman pop up transaksi berhasil approved .....	79
Gambar 5.1.3.2.5. 4 Halaman edit transactions untuk menghapus transaksi .....	80
Gambar 5.1.3.2.5. 5 Halaman berhasil deleted transactions.....	80
Gambar 5.2.3.1. 1 Register berhasil .....	83
Gambar 5.2.3.1. 2 Register gagal .....	83
Gambar 5.2.3.1. 3 Login berhasil .....	84
Gambar 5.2.3.1. 4 Login gagal .....	84
Gambar 5.2.3.1. 5 Logout berhasil .....	85
Gambar 5.2.3.1. 6 Cek Info Login.....	85
Gambar 5.2.3.2. 1 Cek Listings Rusun.....	86
Gambar 5.2.3.2. 2 Cek detail listing rusun .....	86
Gambar 5.2.3.3. 1 Cek ketersediaan rusun .....	87
Gambar 5.2.3.3. 2 Booking dengan melakukan transaksi .....	87
Gambar 5.2.3.3. 3 Transaksi yang telah dilakukan .....	88
Gambar 5.2.3.3. 4 Detail transaksi .....	88

## DAFTAR TABEL

Tabel 2. 1 Profil Instansi .....	7
Tabel 3. 1 Komponen Use Case Diagram .....	14
Tabel 4.1.2. 1 Karakteristik Pengguna .....	20
Tabel 4.1.4. 1 Kebutuhan Fungsional.....	21
Tabel 4.1.5. 1 Kebutuhan Non Fungsional.....	23
Tabel 4.2.2. 1 Skenario Login atau Sign In oleh user .....	25
Tabel 4.2.2. 2 Skenario Login atau Sign In oleh admin .....	26
Tabel 4.2.2. 3 Skenario Logout oleh user.....	27
Tabel 4.2.2. 4 Skenario Logout oleh admin .....	28
Tabel 4.2.2. 5 Skenario Register atau Sign Up oleh user .....	28
Tabel 4.2.2. 6 Skenario melihat List rusun untuk akun admin .....	30
Tabel 4.2.2. 7 Skenario melihat List rusun untuk akun user .....	30
Tabel 4.2.2. 8 Skenario melihat detail rusun untuk akun user.....	31
Tabel 4.2.2. 9 Skenario melihat detail rusun dan edit rusun untuk akun admin.....	32
Tabel 4.2.2. 10 Skenario melihat daftar transaksi untuk akun user.....	33
Tabel 4.2.2. 11 Skenario melihat daftar transaksi untuk akun admin.....	34
Tabel 4.2.2. 12 Skenario menghapus transaksi user untuk akun admin .....	35
Tabel 4.2.2. 13 Skenario menambah rusun untuk akun admin.....	36
Tabel 4.2.2. 14 Skenario menghapus rusun untuk akun admin .....	37
Tabel 4.2.2. 15 Skenario membooking kamar untuk akun user .....	38
Tabel 4.2.2. 16 Skenario beri bukti pembayaran untuk akun user .....	39
Tabel 4.2.2. 17 Skenario verifikasi pembookingan kamar oleh admin .....	41

## **BAB I**

### **PENDAHULUAN**

#### 1.1 Latar Belakang

Magang merupakan sebuah proses pembelajaran yang dilakukan di bawah bimbingan seorang ahli melalui pengalaman langsung dalam dunia kerja. Selain itu, magang juga merupakan sebuah kesempatan untuk mempraktikkan pengetahuan dan keterampilan yang telah diperoleh guna menyelesaikan permasalahan nyata di lingkungan sekitar. Tujuan program magang adalah untuk menciptakan jalur yang menghubungkan antara dunia pendidikan dan dunia kerja. Peserta program ini berasal dari berbagai latar belakang, mulai dari siswa SMA, SMK, mahasiswa, hingga para profesional muda. Program magang ini berpotensi memberikan manfaat besar bagi para pesertanya, seperti meningkatkan kualitas praktikum mahasiswa, memperluas peluang karier, serta menyediakan pengalaman dan wawasan berharga mengenai dunia kerja.

Pemerintah Kota Semarang, melalui Dinas Komunikasi, Informatika, Statistik, dan Persandian, bertanggung jawab dalam mengelola informasi publik, data statistik, komunikasi, dan keamanan data di berbagai bidang pelayanan masyarakat. Salah satu aspek penting yang dikelola adalah penyediaan layanan pembayaran dan pengelolaan invoice, khususnya untuk rumah susun yang menjadi bagian dari fasilitas publik yang disediakan pemerintah bagi masyarakat. Dalam mengelola pembayaran sewa rumah susun, Diskominfo dihadapkan pada tantangan yang cukup signifikan dalam pengelolaan sistem secara efisien.

Saat ini, proses pembayaran sewa rumah susun dan pembuatan invoice masih dilakukan secara manual, yang mengakibatkan beberapa kendala dalam pengelolaan administrasi keuangan. Penghuni rumah susun atau pihak ketiga yang menggunakan fasilitas dinas harus melalui prosedur pembayaran yang memakan waktu, baik melalui pembayaran langsung di kantor atau transfer yang tidak terintegrasi dengan sistem pencatatan otomatis. Kondisi ini menyebabkan terjadinya keterlambatan dalam pencatatan transaksi, serta tingginya risiko kesalahan pencatatan dan ketidakakuratan dalam pengelolaan data pembayaran. Tidak adanya sistem terpusat juga menyulitkan

dalam memantau pembayaran dan melacak riwayat transaksi secara transparan dan real-time.

Pengelolaan invoice juga menjadi tantangan utama. Pembuatan dan pengiriman invoice dilakukan satu per satu oleh petugas secara manual, yang tidak hanya memakan waktu tetapi juga meningkatkan potensi kesalahan. Selain itu, pelacakan status pembayaran untuk setiap invoice sulit dilakukan, mengakibatkan dinas mengalami kesulitan dalam memonitor invoice mana saja yang telah dibayar dan yang belum, serta menghambat pembuatan laporan keuangan yang akurat. Proses manual ini juga memperpanjang waktu penyusunan laporan keuangan yang dibutuhkan manajemen untuk pengambilan keputusan berbasis data yang cepat dan tepat.

Untuk mengatasi permasalahan ini, pengembangan sistem informasi berbasis web yang terintegrasi untuk pembayaran dan manajemen invoice menjadi kebutuhan mendesak bagi Dinas Komunikasi, Informatika, Statistik, dan Persandian Kota Semarang. Sistem informasi ini akan memfasilitasi proses pembayaran sewa rumah susun secara online, sehingga penghuni dapat melakukan pembayaran dengan lebih mudah dan aman, serta langsung tercatat di dalam sistem tanpa keterlambatan atau risiko kesalahan. Selain itu, fitur manajemen invoice dalam sistem ini memungkinkan pembuatan invoice secara otomatis. Dengan demikian, sistem informasi ini diharapkan dapat meningkatkan efisiensi dan transparansi pengelolaan pembayaran serta mendukung pengambilan keputusan yang lebih efektif dan efisien, serta memberikan pengalaman pelayanan yang lebih baik bagi masyarakat.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diungkapkan, rumusan masalah dari kegiatan Praktik Kerja Lapangan yang dilaksanakan di Diskominfo Semarang adalah sebagai berikut:

1. Bagaimana merancang dan membangun aplikasi berbasis web SIPARUS untuk mendukung pengelolaan pembayaran dan manajemen invoice rumah susun di Kota Semarang?
2. Bagaimana penerapan model proses eXtreme Programming (XP) dengan pendekatan berorientasi objek dalam pengembangan aplikasi SIPARUS?

3. Metode pengujian apa saja yang dapat digunakan untuk memastikan bahwa aplikasi SIPARUS sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan?

### 1.3 Tujuan

Tujuan dari pelaksanaan kegiatan Praktik Kerja Lapangan di Diskominfo Semarang adalah:

1. Merancang dan mengembangkan aplikasi berbasis web SIPARUS sebagai solusi sistem informasi pembayaran dan manajemen *invoice* rumah susun di Kota Semarang.
2. Mengimplementasikan proses pengembangan perangkat lunak menggunakan model *eXtreme Programming (XP)* untuk meningkatkan kolaborasi, fleksibilitas, dan efisiensi dalam pengembangan perangkat lunak.
3. Melakukan pengujian perangkat lunak secara menyeluruh untuk memastikan kualitas aplikasi SIPARUS dan kesesuaianya dengan kebutuhan pengguna.

### 1.4 Manfaat

Manfaat dilaksanakannya kegiatan Praktik Kerja Lapangan yang dilakukan di Diskominfo Semarang adalah sebagai berikut:

1. Mendapatkan pengalaman serta kesempatan untuk mengaplikasikan ilmu informatika, khususnya dalam bidang pengembangan perangkat lunak berbasis web.
2. Memahami proses pengembangan perangkat lunak di dunia kerja nyata sebagai persiapan untuk menghadapi tantangan profesional di masa depan.

### 1.5 Ruang Lingkup

1. Proses pengembangan sistem terdiri dari tahapan analisis, desain, implementasi, dan pengujian untuk memastikan sistem bekerja secara efektif dalam mendukung pengelolaan pembayaran dan manajemen invoice rumah susun.
2. Data yang terdapat dalam database mencakup informasi terkait penghuni, detail pembayaran, pengaturan pembayaran, riwayat transaksi, dan notifikasi. Hal ini

- bertujuan untuk mendukung kelancaran pengelolaan pembayaran sewa rumah susun dan memperkuat pengawasan finansial.
3. Data dalam sistem mencakup semua informasi yang telah tersimpan sejak awal penerapan sistem, termasuk riwayat transaksi dan perubahan yang terjadi seiring waktu, guna memberikan rekam jejak yang lengkap dan akurat terkait setiap pembayaran serta pemrosesan invoice.
  4. Teknologi yang digunakan dalam pengembangan sistem ini difokuskan pada lingkungan berbasis web agar penghuni dan administrator dapat mengaksesnya dengan mudah, meningkatkan kenyamanan dalam melakukan pembayaran dan manajemen invoice.
  5. Tahap pengujian menggunakan teknik black box untuk memastikan bahwa semua fungsi sistem bekerja sesuai spesifikasi, khususnya dalam proses pembayaran dan manajemen invoice agar bebas dari kesalahan input dan output yang mungkin terjadi.
  6. Sistem Informasi Pembayaran dan Manajemen Invoice Rumah Susun Kota Semarang dirancang sebagai solusi yang terintegrasi dan berbasis web untuk mendukung pengelolaan pembayaran sewa rumah susun secara efisien. Sistem ini memungkinkan administrator untuk mengelola data pembayaran penghuni.

## 1.6 Sistematika Penulisan

Gambaran dari pembahasan penyusunan implementasi “Sistem Informasi Pembayaran dan Manajemen Invoice Rumah Susun Kota Semarang” disesuaikan dengan sistematika penulisan, antara lain sebagai berikut.

### BAB I

### PENDAHULUAN

Bab ini membahas latar belakang, rumusan masalah, tujuan, manfaat, ruang lingkup, serta sistematika penulisan laporan Praktik Kerja Lapangan (PKL)

### BAB II

### TINJAUAN PERUSAHAAN

Bab ini berisikan informasi tentang instansi tempat Praktik Kerja Lapangan (PKL) yaitu Dinas Komunikasi, Informatika, Statistik, dan Persandian (Diskominfo) Kota Semarang yang meliputi profil instansi, visi, misi, dan struktur organisasi.

### **BAB III** LANDASAN TEORI

Bab ini berisikan penjelasan tentang metodologi dan landasan teori yang digunakan dalam pembuatan Sistem Informasi Pembayaran dan Manajemen Invoice Rumah Susun Kota Semarang, termasuk topik-topik seperti magang, basis data, manajemen basis data, rekayasa perangkat lunak, metode *Extreme Programming*, dengan penerapan teknologi terkini seperti *framework* Next.js 14 untuk *frontend*, Laravel 11 untuk *backend*, dan MySQL.

### **BAB IV** ANALISIS DAN PERANCANGAN

Bab ini berisi tentang analisis kebutuhan, identifikasi kebutuhan, analisis masalah, desain sistem, dan perancangan proses yang digunakan dalam pengembangan Sistem Informasi Pembayaran dan Manajemen Invoice Rumah Susun Kota Semarang.

### **BAB V** HASIL DAN PEMBAHASAN

Bab ini membahas mengenai implementasi sistem, implementasi *database*, implementasi antarmuka sistem, persiapan prosedural dalam pengujian, dan rencana pengujian sistem dalam pembangunan Sistem Informasi Pembayaran dan Manajemen Invoice Rumah Susun Kota Semarang.

## BAB VI

## PENUTUP

Bab ini berisikan pembahasan tentang kesimpulan dari Praktik Kerja Lapangan yang telah dilaksanakan serta saran penulis dalam pengembangan sistem selanjutnya. Informasi yang ditulis diantaranya adalah kesimpulan dan saran

## **BAB II**

### **TINJAUAN PERUSAHAAN**

#### **2.1 Profil Instansi**

Berdasarkan Peraturan Walikota Nomor 76 Tahun 2016, Dinas Komunikasi, Informatika, Statistik, dan Persandian Kota Semarang, yang disebut sebagai Diskominfo Kota Semarang, merupakan instansi pelaksana tugas pemerintahan yang dipimpin oleh seorang Kepala Dinas. Tugasnya adalah mendukung Walikota Semarang dalam menjalankan urusan pemerintahan di bidang Komunikasi, Informatika, Statistik, dan Persandian yang merupakan kewenangan daerah, serta melaksanakan tugas pembantuan yang diberikan oleh pemerintah. Tugas pembantuan Diskominfo ini merujuk pada penugasan dari Pemerintah Pusat kepada daerah, di mana Diskominfo bertanggung jawab untuk melaksanakan sebagian urusan pemerintahan yang menjadi wewenang Pemerintah Pusat. Sebaliknya, Diskominfo juga dapat ditugaskan oleh Pemerintah Daerah Provinsi untuk melaksanakan sebagian urusan pemerintahan yang menjadi kewenangan Daerah Provinsi.

**Tabel 2. 1 Profil Instansi**

Nama Instansi	Dinas Komunikasi, Informatika, Statistik, dan Persandian (Diskominfo) Kota Semarang
Kepala Dinas	Soenarto, S.Kom, MM
Alamat	Jl. Pemuda 148 Semarang, Jawa Tengah
Kode Pos	50132
Email	Diskominfo@semarangkota.go.id
Nomor Telepon	(024) 3549446

#### **2.2 Visi Instansi**

Terwujudnya Kota Semarang yang Semakin Hebat berlandaskan Pancasila dalam Bingkai NKRI Yang Ber-Bhineka Tunggal Ika

### **2.3 Misi Instansi**

1. Meningkatkan kualitas dan kapasitas Sumber Daya Manusia (SDM) agar dapat mencapai tingkat keunggulan dan produktivitas yang mendukung kesejahteraan dan keadilan sosial.
2. Meningkatkan potensi ekonomi lokal yang memiliki daya saing tinggi, serta mendorong stimulasi pembangunan industri melalui riset dan inovasi, dengan landasan pada prinsip demokrasi ekonomi Pancasila.
3. Menjamin kemerdekaan masyarakat dalam menjalankan ibadah, pemenuhan hak dasar, perlindungan kesejahteraan sosial, dan hak asasi manusia, dengan prinsip keadilan bagi seluruh lapisan masyarakat.
4. Mewujudkan infrastruktur berkualitas yang memperhatikan aspek lingkungan guna mendukung kemajuan kota.
5. Menjalankan reformasi birokrasi pemerintahan secara dinamis, serta penyusunan produk hukum yang sesuai dengan nilai-nilai Pancasila, dalam kerangka Negara Kesatuan Republik Indonesia.

### **2.4 Tugas dan Fungsi Diskominfo Kota Semarang**

Tugas dan fungsi Dinas Komunikasi, Informatika, Statistik, dan Persandian Kota Semarang, sebagaimana diatur dalam Peraturan Walikota Semarang Nomor 76 Tahun 2016, mencakup hal-hal berikut:

Kepala Diskominfo bertanggung jawab membantu Walikota dalam menjalankan urusan pemerintahan di Bidang Komunikasi dan Informatika, Bidang Statistik, dan Bidang Persandian, yang merupakan kewenangan daerah, serta pelaksanaan tugas pembantuan yang ditugaskan kepada daerah.

Saat menjalankan tugasnya, Diskominfo melaksanakan fungsi-fungsi berikut:

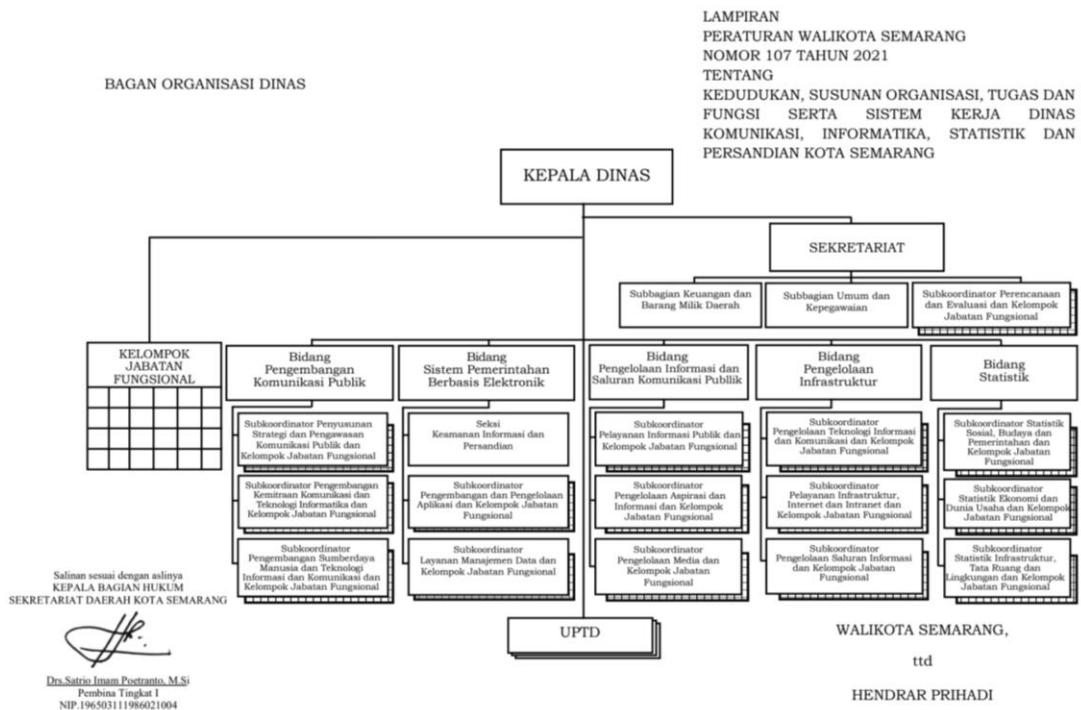
- a. Perumusan kebijakan di Bidang Pengembangan Komunikasi Publik, Bidang Layanan *E-Government*, Bidang Pengelolaan Informasi dan Saluran Komunikasi Publik, Bidang Pengelolaan Infrastruktur, dan Bidang Statistik;

- b. Perumusan rencana strategis sesuai dengan visi dan misi Walikota;
- c. Pengkoordinasian tugas-tugas dalam pelaksanaan program dan kegiatan di Kesekretariatan, Bidang Pengembangan Komunikasi Publik, Bidang Layanan *E-Government*, Bidang Pengelolaan Informasi dan Saluran Komunikasi Publik, Bidang Pengelolaan Infrastruktur, dan Bidang Statistik;
- d. Pembinaan pegawai sesuai dengan tanggung jawabnya;
- e. Penyusunan Sasaran Kerja Pegawai;
- f. Kerjasama di Bidang Pengembangan Komunikasi Publik, Bidang Layanan *E-Government*, Bidang Pengelolaan Informasi dan Saluran Komunikasi Publik, Bidang Pengelolaan Infrastruktur, dan Bidang Statistik;
- g. Pelaksanaan tugas kesekretariatan Diskominfo;
- h. Pelaksanaan program dan kegiatan di Bidang Pengembangan Komunikasi Publik, Bidang Layanan *E-Government*, Bidang Pengelolaan Informasi dan Saluran Komunikasi Publik, Bidang Pengelolaan Infrastruktur, dan Bidang Statistik;
- i. Penilaian kinerja Pegawai;
- j. Monitoring dan evaluasi program dan kegiatan di Bidang Pengembangan Komunikasi Publik, Bidang Layanan *E-Government*, Bidang Pengelolaan Informasi dan Saluran Komunikasi Publik, Bidang Pengelolaan Infrastruktur, dan Bidang Statistik;
- k. Pelaporan pelaksanaan program dan kegiatan; dan
- l. Pelaksanaan fungsi lain yang diberikan oleh Walikota terkait dengan tugas dan fungsinya.

## **2.5 Struktur Organisasi Instansi**

Menurut Peraturan Walikota Nomor 76 Tahun 2016, Dinas Komunikasi, Informatika, Statistik, dan Persandian Kota Semarang memiliki Struktur Organisasi yang terdiri dari Kepala Dinas, Sekretariat, Kelompok Jabatan Fungsional, dan lima bidang, yakni Bidang Pengembangan Komunikasi Publik, Bidang Layanan *E-*

*Government*, Bidang Pengelolaan Informasi dan Saluran Komunikasi Publik, Bidang Pengelolaan Infrastruktur, serta Bidang Statistik. Setiap bidang memiliki beberapa seksi yang mendukung pelaksanaan tugasnya. Gambar 2.1 menunjukkan tata letak Struktur Organisasi ini. Pkl dilaksanakan dibawah bidang Dinas Komunikasi, Informatika, Statistik, dan Persandian dibawah pengawasan / supervisi Bapak Hanry Sugiharto.



**Gambar 2. 1 Struktur Organisasi Dikominfo Kota Semarang**

## **BAB III**

### **LANDASAN TEORI**

#### **3.1 Aplikasi Berbasis Web**

Aplikasi adalah program yang dibuat oleh pemakai yang ditujukan untuk melakukan suatu tugas khusus (Kadir, 2003). Menurut Kadir (2008:3) program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi berbasis web merupakan program yang dapat diakses menggunakan web browser melalui jaringan internet tanpa memerlukan instalasi di perangkat pengguna (Ovan dkk., 2020). Salah satu kelebihan utama aplikasi berbasis web adalah aksesibilitasnya yang mudah dan fleksibel. Pengguna dapat mengakses aplikasi berbasis web dari berbagai perangkat selama browser yang digunakan kompatibel dengan aplikasi tersebut. Selain itu, pembaruan aplikasi berbasis web tidak perlu melakukan proses instalasi ulang, yang menghemat ruang penyimpanan, meminimalisir pembajakan, dan memudahkan pemeliharaan. Namun, aplikasi berbasis web juga memiliki beberapa kelemahan. Salah satunya adalah ketergantungan pada koneksi internet. Tanpa koneksi yang stabil, pengguna mungkin mengalami kesulitan dalam mengakses atau menggunakan aplikasi secara optimal.

#### **3.2 Metode Pengembangan Perangkat Lunak *Agile***

Pengembangan perangkat lunak *Agile* adalah cara yang lebih fleksibel untuk mengembangkan perangkat lunak dibandingkan dengan pendekatan yang berbasis rencana (plan-driven approach). Fleksibilitas ini terlihat dari kemampuannya untuk mengubah persyaratan pada setiap fase dalam siklus pengembangan perangkat lunak (Dönmez, Grote, & Brusoni, 2016). Metode *agile* adalah metode pengembangan incremental yang berfokus pada perkembangan cepat, merilis perangkat lunak secara bertahap, mengurangi *overhead* proses, dan menghasilkan kode berkualitas tinggi dengan melibatkan pelanggan secara langsung dalam prosesnya. Beberapa model pengembangan perangkat lunak yang termasuk dalam metode *agile* adalah *Extreme Programming*, *Adaptive Software Development*, *Dynamic Systems Development*

*Method, Scrum, dan Agile Modeling* (Mahendra & Yanto, 2018). Dalam penelitian ini, model yang akan digunakan adalah model *Extreme Programming*.

### 3.3 *Extreme Programming*

*Extreme programming (XP)* adalah metodologi dalam pengembangan *agile software development methodologies* yang berfokus pada pengkodean (*coding*) yang menjadi aktivitas utama dalam semua tahapan pada siklus pengembangan perangkat lunak (Gumelar dkk., 2017). Metode *XP* merupakan metode yang responsif terhadap perubahan (Sulianta, 2019). Dalam *XP* terdapat iterasi yang bisa dilakukan berulang kali sesuai dengan kebutuhan. *XP* menawarkan tahapan dalam waktu yang singkat dan berulang untuk bagian - bagian yang berbeda sesuai dengan fokus yang akan dicapai. Tahapan pengembangan perangkat lunak dengan *XP* meliputi: *planning* (perencanaan), *design* (perancangan), *coding* (pengkodean) dan *testing* (pengujian) (Suryantara, 2017).

Berikut adalah tahapan dari Extreme programming :

1. *Planning* (Perencanaan): Tahap perencanaan dimulai dengan pemahaman konteks bisnis dari aplikasi, mendefinisikan output, fitur yang ada pada aplikasi, fungsi dari aplikasi yang dibuat, serta alur pengembangan aplikasi (Suryantara, 2017).
2. *Design* (Perancangan): Pada tahap fokus pada design aplikasi secara sederhana, alat untuk mendesign pada tahap ini dapat menggunakan CRC (*Class Responsibility Collaborator*) (Suryantara, 2017).
3. *Coding* (Pengkodean): *Coding* atau pengkodean merupakan penerjemahan dari perancangan dalam bahasa pemrograman yang dikenali oleh komputer (Melinda dkk., 2018).
4. *Testing* (Pengujian): Sistem yang telah dibangun harus diuji terlebih dahulu agar dapat menemukan kesalahan - kesalahan (Kumala dkk., 2018).

### 3.4 *Unified Modelling Language (UML)*

*Unified Modelling Language* atau UML adalah bahasa pemodelan berorientasi objek yang diciptakan pada tahun 1997 oleh perusahaan Object Management Group. UML biasa digunakan untuk pemodelan sistem berbasis objek. UML dapat digunakan untuk membuat rancangan semua jenis model perangkat lunak dimana perangkat lunak yang dibuat menggunakan bahasa pemrograman apapun, dengan sistem operasi dan

jaringan apapun, serta dapat dijalankan pada perangkat keras apapun (Najib & Zain, 2020; Yuniarti dkk., 2022).

Dalam UML, terdiri dari 13 macam diagram yang dikelompokkan menjadi 3 kategori sebagai berikut.

### 1. *Structure diagrams*

Kategori ini merupakan kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan dan terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram*, dan *deployment diagram*.

### 2. *Behaviour diagrams*

*Behaviour diagrams* merupakan kumpulan diagram yang menggambarkan aktivitas sistem yang terjadi dimana terdiri dari *use case diagram*, *state machine diagram*, dan *activity diagram*.

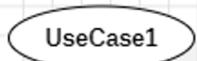
### 3. *Interaction diagrams*

*Interaction diagrams* adalah kumpulan diagram untuk menggambarkan interaksi sistem dengan sistem lain atau antar subsistem dalam suatu sistem, terdiri dari *sequence diagram*, *communication diagram*, *timing diagram*, dan *interaction overview diagram*.

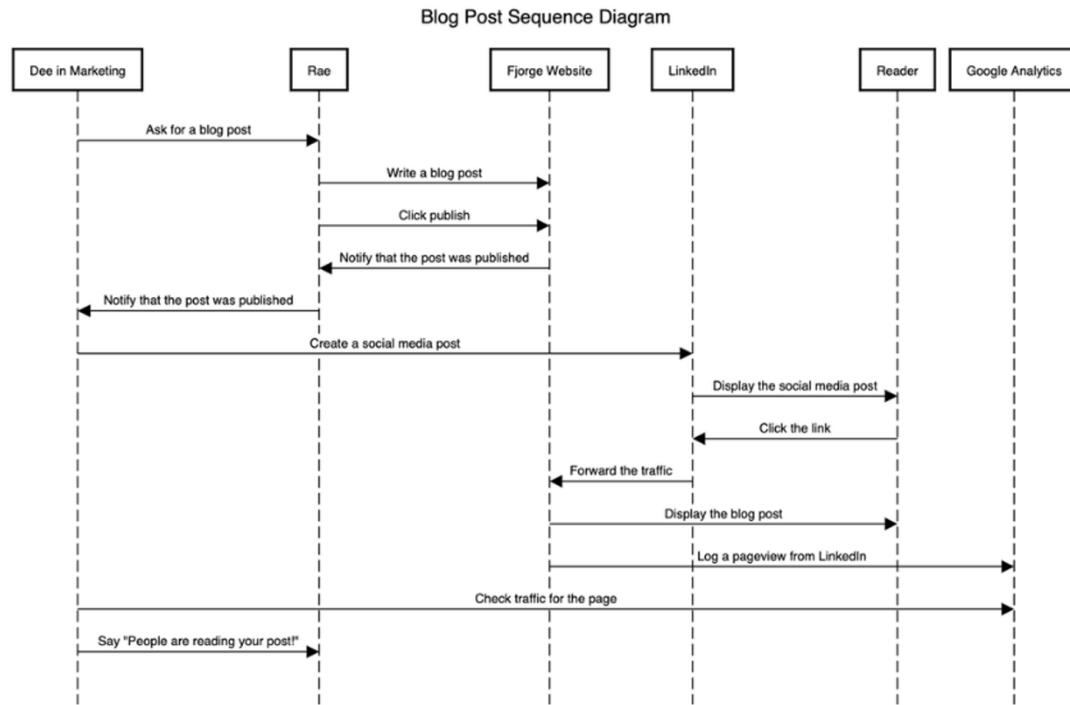
## 3.5 *Use Case Diagram*

Sebuah *use case diagram* menyatakan visualisasi interaksi yang terjadi antara pengguna dengan sistem (Kurniawan, 2018). Secara sederhana, *use case diagram* digunakan untuk memahami fungsi apa saja yang ada di dalam sistem dan siapa saja yang dapat mengetahui fungsi tersebut. Langkah langkah sebelum melakukan perancangan atau pemodelan dalam rekayasa perangkat lunak, adalah dengan melakukan rekayasa kebutuhan. Kebutuhan sendiri adalah suatu kondisi yang dibutuhkan oleh user untuk mengatasi masalah. Sedangkan rekayasa kebutuhan adalah proses mewujudkan serangkaian layanan yang dibutuhkan oleh pemangku kepentingan atas suatu sistem (Sommerville, 2011). Contoh komponen *use case diagram* terdapat pada tabel 3.1.

Tabel 3. 1 Komponen Use Case Diagram

Simbol	Komponen	Keterangan
 Aktor	Aktor	Melambangkan entitas yang berinteraksi dengan <i>use case</i> , bisa berupa pengguna, sistem lain, atau perangkat lain saat berkomunikasi dengan fungsionalitas sistem.
	Use case	Abstraksi dan hubungan antara sistem dengan entitas yang berinteraksi (aktor).
	Asosiasi	Abstraksi dari penghubung antara aktor dengan <i>use case</i> .
	Generalisasi	Representasi dari penyesuaian aktor yang mengkhususkan diri untuk berpartisipasi dalam <i>use case</i> tertentu.
	<<include>>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	<<extend>>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

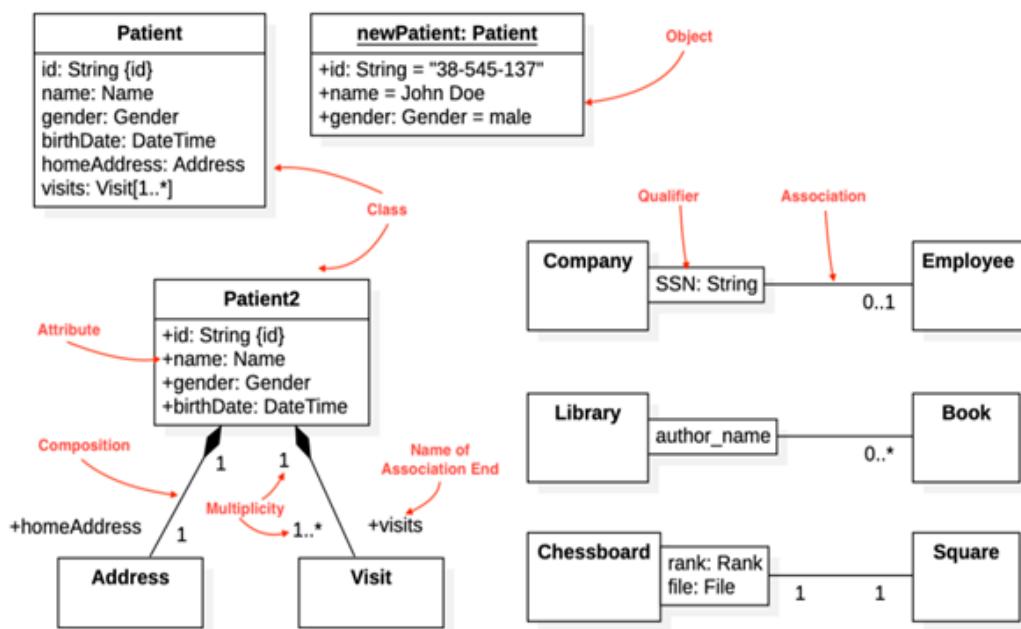
### 3.6 Sequence Diagram



Gambar 3. 1 Contoh Sequence Diagram

*Sequence Diagram* seperti yang tertera pada Gambar 3.1 Contoh Sequence Diagram adalah sebuah representasi visual yang menunjukkan hasil interaksi yang terjadi antara objek objek yang saling bertukar pesan. Diagram ini menggambarkan interaksi antara objek dengan menyoroti urutan waktu dari pesan yang dikirim. *Sequence Diagram* memiliki dimensi vertikal yang menggambarkan waktu dan dimensi horizontal yang menggambarkan objek-objek yang terlibat dalam interaksi tersebut (Nugroho dkk., 2017).

### 3.7 Class Diagram



Gambar 3. 2 Contoh Class Diagram

Menurut Wira dkk. (2019), *class diagram* atau diagram kelas yang tertera dalam Gambar 3.2 adalah representasi umum yang menggambarkan kelas-kelas yang terlibat dalam desain sistem dari perspektif struktur sistem, yang membantu dalam memperjelas fungsi-fungsinya. Gambar di atas menunjukkan hubungan antara entitas dalam sistem, termasuk atribut utama yang digunakan di setiap kelas.

*Class Diagram* umumnya mencakup kelas, antarmuka, kolaborasi, serta hubungan seperti ketergantungan, generalisasi, dan asosiasi. Seperti diagram lainnya, *Class Diagram* juga dapat mencakup catatan dan batasan, serta paket atau subsistem untuk mengelompokkan elemen model menjadi bagian yang lebih besar. *Class*

*Diagram* yang baik dirancang untuk fokus pada satu aspek desain statis sistem, mencakup elemen-elemen penting yang relevan, memberikan detail sesuai dengan tingkat abstraksi, dan menghindari elemen yang berlebihan atau terlalu minimalis (Booch dkk., 1998).

## 3.8 Bahasa Pemrograman dan Database

### 3.8.1 Framework laravel

Laravel merupakan sebuah *framework* web yang dibangun dengan bahasa pemrograman PHP, bersifat *open-source*, dan tidak berbayar. *Framework* ini dikembangkan oleh Taylor Otwell dan dirancang khusus untuk memudahkan pengembangan aplikasi web dengan menggunakan pola MVC (*Model-View Controller*). Salah satu perbedaan utama dari struktur MVC pada Laravel adalah adanya komponen routing yang menghubungkan permintaan dari pengguna dengan *controller*, sehingga *controller* tidak langsung menerima permintaan tersebut (Tahir dkk., 2019).

### 3.8.2 Framework Next.js

Next.js adalah kerangka kerja berbasis React yang dirancang untuk mempermudah pengembangan aplikasi web modern. *Framework* ini menawarkan berbagai fitur seperti *Server-Side Rendering* (SSR), pembuatan *Static Site Generation* (SSG), pemisahan kode otomatis, dan sistem routing berbasis halaman. Fitur-fitur ini memungkinkan pengembang membangun aplikasi yang responsif dan ramah SEO (Thakkar, 2020).

Menurut Thakkar (2020), salah satu keunggulan Next.js adalah kemudahannya dalam menangani konfigurasi webpack dan Babel, sehingga cocok untuk pengembang yang ingin fokus pada logika aplikasi tanpa khawatir tentang pengaturan dasar. Selain itu, Next.js juga menyediakan dukungan untuk pemuatan dinamis yang membantu meningkatkan performa aplikasi.

Libby (2023) menekankan bahwa Next.js menjadi pilihan populer untuk pengembangan situs e-commerce berkat kemampuan skalabilitas dan keamanan yang ditawarkannya. *Framework* ini memanfaatkan kekuatan JavaScript dan Node.js untuk membangun aplikasi yang cepat, handal, dan mudah dikembangkan.

Dengan komunitas aktif dan ekosistem yang terus berkembang, Next.js terus menjadi kerangka kerja pilihan untuk membangun aplikasi web yang modern dan efisien (Thakkar, 2020; Libby, 2023).

### **3.8.3 Database MySQL & Phpmyadmin**

MySQL adalah salah satu jenis database server yang menggunakan SQL (*Structured Query Language*) sebagai bahasa untuk mengakses database. MySQL merupakan sebuah DBMS (*Database Management System*) yang dijalankan dengan perintah SQL untuk menyimpan dan memanipulasi data, serta menentukan keterkaitan antar masing-masing tabel (Najib & Zain, 2020; Yuniarti dkk., 2022). Cara kerja MySQL yaitu diawali dengan membuat database yang dapat memodifikasi, menyimpan data, dan menentukan keterkaitan tabel yang ada, kemudian perangkat pengguna membuat *request* dengan perintah spesifik menggunakan bahasa SQL, setelah itu server akan menerima dan menjalankan perintah yang kemudian menampilkan informasi yang diminta pengguna.

Menurut Sitinjak dkk. (2020), PHPMyAdmin adalah aplikasi open-source untuk memudahkan manajemen DBMS MySQL. Dengan menggunakan PHPMyAdmin dapat mengelola database seperti membuat database, membuat tabel, meng-insert, meng-update, dan menghapus data dengan lebih mudah dan efisien karena dapat melakukannya dengan GUI tanpa perlu mengetikkan query SQL secara manual. Selain dapat mengelola database dengan lebih efisien, PHPMyAdmin memiliki beberapa kelebihan lain seperti tampilan User Interface yang mudah digunakan, dapat dijalankan di berbagai server dan sistem operasi, dapat menambahkan banyak user dan mengelola banyak server, dapat mengekspor data ke berbagai format seperti CSV, SQL, XML, dan lainnya, serta dapat mengeksekusi statement dan query SQL yang kompleks.

### **3.8.4 Bahasa JavaScript**

JavaScript adalah bahasa pemrograman yang digunakan untuk meningkatkan interaktivitas halaman web dengan mengubah tampilan dokumen HTML pada browser menjadi lebih dinamis. Fungsionalitas yang disediakan oleh JavaScript memungkinkan pembuatan program dengan antarmuka web, tanpa memerlukan proses kompilasi karena cukup dengan interpreter. Browser

seperti Netscape Navigator dan Internet Explorer menyediakan interpreter JavaScript, namun tidak semua browser memiliki fitur ini. JavaScript memberikan kemudahan dalam pengembangan aplikasi berbasis web dengan antarmuka yang interaktif, memberikan solusi atas tantangan untuk membuat halaman web yang lebih dinamis dan menarik bagi pengguna, serta menghindari kebosanan pengunjung akibat halaman web yang statis (Yani & Saputra, 2018).

### 3.8.5 Bahasa PHP

PHP adalah singkatan dari *PHP Hypertext Preprocessor*, sebuah bahasa scripting yang dijalankan di server. Ini berarti PHP memungkinkan pengolahan data di server dan dapat digunakan dalam dokumen HTML untuk menghasilkan konten yang dinamis. Penggunaan PHP memungkinkan pembuatan situs web yang lebih dinamis daripada sekadar kumpulan halaman statis, karena kontennya dapat diperbarui secara otomatis. Keuntungan menggunakan PHP termasuk kesederhanaan bahasa dan mesin scripting, siklus pengembangan yang lebih singkat, konektivitas ke server basis data, serta sifat open-source yang tidak tergantung pada platform tertentu (Yani & Saputra, 2018).

### 3.8.6 Bahasa HTML & CSS

HTML adalah bahasa standar web yang dikelola oleh W3C (*World Wide Web Consortium*) dan terdiri dari tag-tag yang membentuk setiap elemen dari sebuah website. Sebagai penyusun struktur halaman web, HTML memungkinkan penempatan elemen-elemen layout yang diinginkan. Biasanya disimpan dalam file berekstensi .html, HTML dapat ditulis menggunakan berbagai text editor, mulai dari yang sederhana seperti Notepad hingga yang khusus dan lebih canggih seperti Notepad++, Sublime Text, dan sejenisnya. Editor yang lebih canggih dapat mengenali setiap unsur HTML dan menampilkan mereka dengan warna yang berbeda, memudahkan pembacaan dan penulisan skrip HTML (Sari & Suhendi, 2020).

CSS merupakan dokumen web yang bertanggung jawab dalam mengatur elemen-elemen HTML dengan berbagai properti sehingga dapat ditampilkan dengan gaya yang diinginkan. Meskipun sederhana dalam strukturnya, CSS tidak dianggap sebagai bahasa pemrograman oleh sebagian orang karena hanya berupa kumpulan aturan yang mengatur gaya elemen HTML. Cara kerja CSS

melibatkan pemilihan elemen HTML yang akan dimodifikasi, diikuti dengan pemberian properti yang sesuai untuk mencapai tampilan yang diinginkan. Aturan dalam CSS terdiri dari tiga bagian utama, yaitu selector untuk memilih elemen yang akan diatur, property yang merupakan aturan yang diberikan, dan value sebagai nilai dari aturan yang diberikan (Sari & Suhendi, 2020).

### **3.8.7 *Black Box Testing***

Salah satu metode dalam pengujian perangkat lunak yaitu dengan metode pengujian *black box* yang merupakan salah satu metode pengujian yang dapat memastikan sistem telah beroperasi sesuai kebutuhan tanpa melihat struktur program dari dalam sistem. Pengujian ini berfokus pada persyaratan fungsional perangkat lunak yang telah ditentukan di awal proses perancangan perangkat lunak. Pengujian *black box* berusaha menemukan kesalahan dalam beberapa kategori yaitu fungsi yang tidak benar atau hilang, kesalahan antarmuka, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan kinerja, serta insiasi dan kesalahan terminasi. Tahapan dalam pengujian *black box* yaitu diawali dengan membuat test case kebutuhan yang akan diuji berdasarkan fungsi-fungsi sistem yang telah dikembangkan dan hasil pengujian menjadi indikator bahwa sistem sesuai dengan kebutuhan pengguna. Pada proses pengujian, penguji akan memasukkan test data untuk diproses oleh sistem dan melihat hasil keluarannya. Kemudian hasil keluaran sistem tersebut menunjukkan apakah sistem tervalidasi atau tidak (Pressman, 2010).

### **3.8.2 *White Box Testing***

*White box testing* adalah pengujian perangkat lunak pada tingkat alur kode program, apakah masukan dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan. (Cholifah, Yulianingsih, & Sagita, 2018). *White box testing* juga merupakan metode pengujian perangkat lunak yang didasarkan pada pengujian design program secara prosedural, secara struktural, pengujian berbasis logika atau pengujian berbasis kode (Irawan, 2017)

## **BAB IV**

### **ANALISIS & RANCANGAN**

#### **4.1 Analysis Kebutuhan**

Analisis kebutuhan merupakan tahap awal dalam pengembangan sistem yang bertujuan untuk mengidentifikasi dan memahami kebutuhan yang diperlukan serta menganalisis permasalahan sistem yang akan dibangun. Tahapan ini meliputi identifikasi kebutuhan perangkat lunak, kebutuhan fungsional dan nonfungsional, serta karakteristik pengguna. Proses ini penting untuk mendeskripsikan secara umum aplikasi yang akan dikembangkan sehingga dapat memenuhi kebutuhan sistem secara menyeluruh.

##### **4.1.1 Deskripsi Umum Aplikasi SIPARUS**

Aplikasi SIPARUS, adalah aplikasi yang dibuat dengan tujuan untuk membantu bagian pengelolaan rusun di DISKOMINFO Semarang dalam menyediakan pembookingan kamar pada sebuah rusun di Kota Semarang. Aplikasi ini memiliki beberapa fitur, diantaranya:

1. Penyediaan untuk pembookingan kamar pada rusun yang dapat dilihat oleh pengunjung aplikasi(*user*) dan dapat dikelola oleh admin.
2. Pembayaran booking didahului dengan mengontak whatsapp admin yang kemudian admin memberikan pilihan metode pembayaran kepada pengunjung aplikasi(*user*) via chat whatsapp. Bukti pembayaran dibutuhkan yang nantinya akan digunakan untuk verifikasi pembayaran booking oleh admin.

##### **4.1.2 Karakteristik Pengguna**

Karakteristik pengguna aplikasi berbasis web SDM.ku dapat dilihat pada Tabel 4.1.2.1.

**Tabel 4.1.2. 1 Karakteristik Pengguna**

NO	Jenis Pengguna	Deskripsi
1	Admin	Admin merupakan pengguna yang memiliki hak akses untuk melihat semua informasi terkait rusun, kamar, pengguna, dan pembayaran booking. Admin dapat mengedit data admin, menghapus user, mengelola

		pembookingan kamar pada rusun, dan mengelola rusun seperti menambahkan, meng-update, dan menghapus data rusun ataupun kamar pada sebuah rusun.
2	User	User merupakan pengguna yang memiliki hak akses terbatas yang dimana hanya dapat melihat list rusun. fokus user adalah membooking sebuah kamar pada rusun yang ingin ditinggali sesuai inputan tanggal dari User tersebut.

#### 4.1.3 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak dari aplikasi SIPARUS yang dikembangkan dibagi menjadi dua bagian, yaitu kebutuhan web browser dan kebutuhan database server.

##### 1. Web Browser

Jenis web browser yang dapat digunakan untuk mengakses aplikasi SIPARUS adalah Microsoft Edge, Google Chrome, dan lainnya.

##### 2. Database Server

basis data diperlukan untuk menyimpan dan mengelola data pada aplikasi SIPARUS. Server basis data yang digunakan adalah MySQL yang diakses melalui PhpMyAdmin.

#### 4.1.4 Kebutuhan Fungsional

Tabel kebutuhan Fungsional tertera pada tabel 4.1.4.1.

**Tabel 4.1.4. 1 Kebutuhan Fungsional**

No	SRS-ID	Fungsionalitas	Deskripsi
1.	SRS-F-1	Melakukan <i>Login</i> (Admin & User)	Sistem memungkinkan Admin dan User untuk <i>login</i> menggunakan kredensial mereka.
2.	SRS-F-2	Melakukan <i>Logout</i> (Admin & User)	Sistem memungkinkan Admin dan User untuk <i>logout</i> .
3.	SRS-F-3	Melakukan <i>Registrasi</i>	Sistem menyediakan

		(User)	fitur untuk User baru melakukan <i>registrasi</i> , dan data akan disimpan ke dalam database.
4.	SRS-F-4	Menghapus Transaksi User (Admin)	Admin dapat menghapus data Transaksi User dari database.
5.	SRS-F-5	Menambah Data Rusun (Admin)	Admin dapat menambahkan data rusun baru ke dalam sistem.
6.	SRS-F-6	Mengubah Detail Rusun (Admin)	Admin dapat mengubah detail kamar yang ada di sistem.
7.	SRS-F-7	Melihat Detail Rusun (Admin & User)	Admin dan User dapat melihat informasi lengkap rusun tertentu.
8.	SRS-F-8	Menghapus Data Rusun (Admin)	Admin dapat menghapus data kamar yang ada di sistem.
9.	SRS-F-9	Melihat List Rusun (Admin & User)	Admin dan User dapat melihat daftar rusun yang tersedia di sistem.
10.	SRS-F-10	Membooking Kamar Rusun (User)	User dapat membooking kamar berdasarkan tanggal dan ketersediaan kamar.
11.	SRS-F-11	Melihat Daftar Transaksi (Admin & User)	Admin dan User dapat melihat daftar transaksi

12.	SRS-F-12	Memberikan Bukti Pembayaran (User)	User dapat mengunggah bukti pembayaran melalui media komunikasi WhatsApp.
13.	SRS-F-13	Mengkonfirmasi Pembayaran (Admin)	Admin memberikan konfirmasi pembayaran yang telah divalidasi kepada User.

#### 4.1.5 Kebutuhan Non Fungsional

**Tabel 4.1.5. 1 Kebutuhan Non Fungsional**

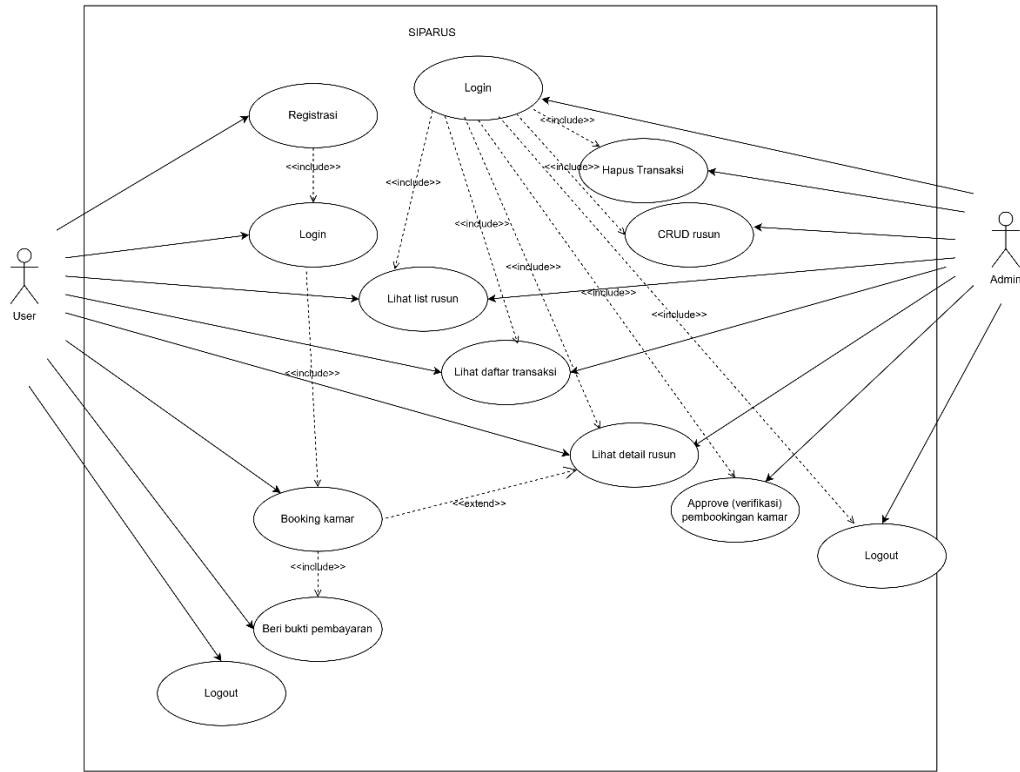
No	SRS-ID	Deskripsi
1.	SRS-N-1	Sistem dapat berjalan pada browser yang mendukung HTML versi 5 dan Javascript.
2.	SRS-N-2	Sistem harus mendukung keamanan tingkat tinggi, termasuk enkripsi password dan data sensitif pengguna, dengan integrasi HTTPS untuk komunikasi data yang aman antara klien dan server.

#### 4.2 Desain Aplikasi

Tahapan berikutnya yang dilakukan adalah mendesain aplikasi. Pada tahap ini, setelah kebutuhan atau *requirements* telah dideskripsikan, proses desain aplikasi dimulai dengan pembuatan diagram dan rancangan antarmuka. Diagram yang digunakan meliputi diagram *use case* untuk menggambarkan dan mendefinisikan *use case* yang dibutuhkan oleh aplikasi, diagram *sequence* untuk merancang alur proses aplikasi, serta diagram *class* yang digunakan untuk mendesain struktur database. Tahap

akhir dari proses ini adalah pembuatan rancangan antarmuka sebagai desain awal tampilan aplikasi.

#### 4.2.1 *Diagram Use Case*



**Gambar 4.2.1. 1 Use Case SIPARUS**

- Pada use case yang tertera pada gambar 4.2.1.1, CRUD yang dilakukan oleh admin pada rusun diantaranya, menambah rusun, mengubah detail rusun, menghapus rusun.

#### 4.2.2 *Diagram Sequence*

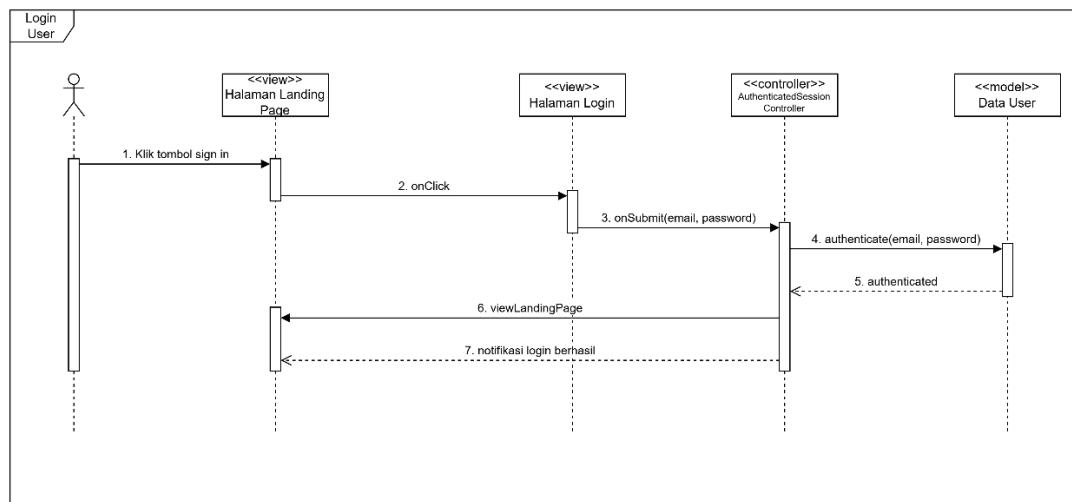
Berdasarkan diagram *use case* yang telah dibuat, tahap selanjutnya yang dapat dilakukan adalah membuat skenario untuk setiap *use case* yang ada pada diagram *use case*. Berdasarkan skenario-skenario tersebutlah diagram *sequence* dapat dibuat.

## 1. Login atau Sign In oleh User

Skenario Login atau Sign In oleh user tertera pada tabel 4.2.2.1 dan gambar sequencenya tertera pada gambar 4.2.2.1.

**Tabel 4.2.2. 1 Skenario Login atau Sign In oleh user**

Aktor	User
Nama Skenario	Melakukan <i>login</i> dengan akun user
Flow of events (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. User meneklik tombol <i>sign in</i> pada <i>landing page</i></li> <li>2. Sistem menampilkan halaman <i>login</i></li> <li>3. User memasukkan email dan password pada form di halaman <i>login</i></li> <li>4. User meneklik tombol <i>Sign In</i></li> <li>5. Sistem akan melakukan proses autentikasi email dan password yang dimasukkan.</li> <li>6. Autentikasi yang dilakukan sistem berhasil. Username dan password yang dimasukkan benar dan role akun adalah user.</li> <li>7. Sistem menampilkan <i>landing page</i></li> </ol>
Alternate Scenario	-
Exceptional Scenario	<p>5.1.1 Autentikasi yang dilakukan sistem gagal. Email atau password yang dimasukkan salah.</p> <p>5.1.2 Sistem menampilkan pesan gagal <i>login</i>.</p>



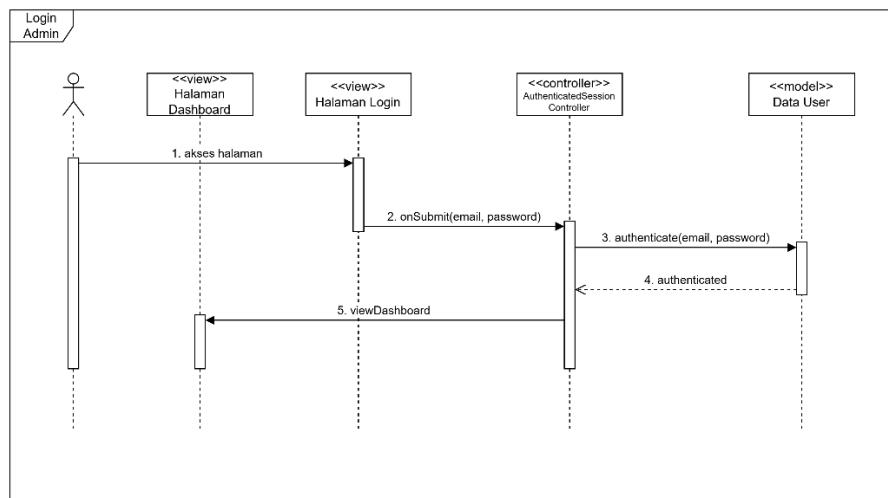
**Gambar 4.2.2. 1 Sequence Diagram Login atau Sign In oleh user**

## 2. Login atau Sign In oleh Admin

Skenario Login atau Sign In oleh admin tertera pada tabel 4.2.2.2 dan gambar sequencenya tertera pada gambar 4.2.2.2.

Tabel 4.2.2. 2 Skenario Login atau Sign In oleh admin

Aktor	Admin
Nama Skenario	Melakukan <i>login</i> dengan akun admin
Flow of events (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. User mengakses halaman <i>login</i> pada admin</li> <li>2. Sistem menampilkan halaman <i>login</i></li> <li>3. Admin memasukkan email dan password pada form di halaman <i>login</i></li> <li>4. User menekan tombol <i>Sign In</i></li> <li>5. Sistem akan melakukan proses autentikasi email dan password yang dimasukkan.</li> <li>6. Autentikasi yang dilakukan sistem berhasil. Username dan password yang dimasukkan benar dan role akun adalah admin.</li> <li>7. Sistem menampilkan <i>dashboard</i></li> </ol>
Alternate Scenario	-
Exceptional Scenario	<p>5.1.1 Autentikasi yang dilakukan sistem gagal. Email atau password yang dimasukkan salah.</p> <p>5.1.2 Sistem menampilkan pesan gagal <i>login</i>.</p>



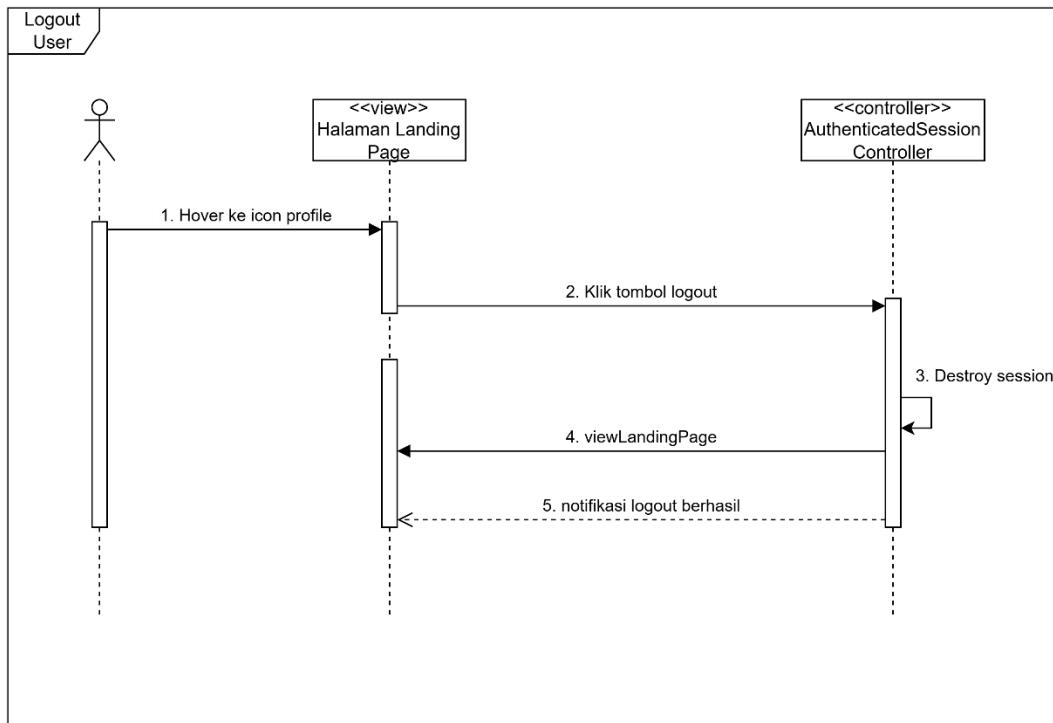
Gambar 4.2.2. 2 Sequence Diagram Login atau Sign In oleh admin

### 3. Logout oleh User

Skenario Logout oleh user tertera pada tabel 4.2.2.3 dan gambar sequencenya tertera pada gambar 4.2.2.3.

**Tabel 4.2.2. 3 Skenario Logout oleh user**

Aktor	User
Nama Skenario	Melakukan <i>logout</i> dengan akun user
<i>Flow of events</i> (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. User <i>hover</i> atau mengarahkan cursor ke tombol ikon <i>profile</i> pada <i>landing page</i></li> <li>2. User menekan tombol <i>log out</i></li> <li>3. Sistem melakukan <i>destroy session</i></li> <li>4. Sistem menampilkan <i>landing page</i></li> </ol>
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	3.1.1 Sistem menampilkan pesan gagal <i>log out</i> .



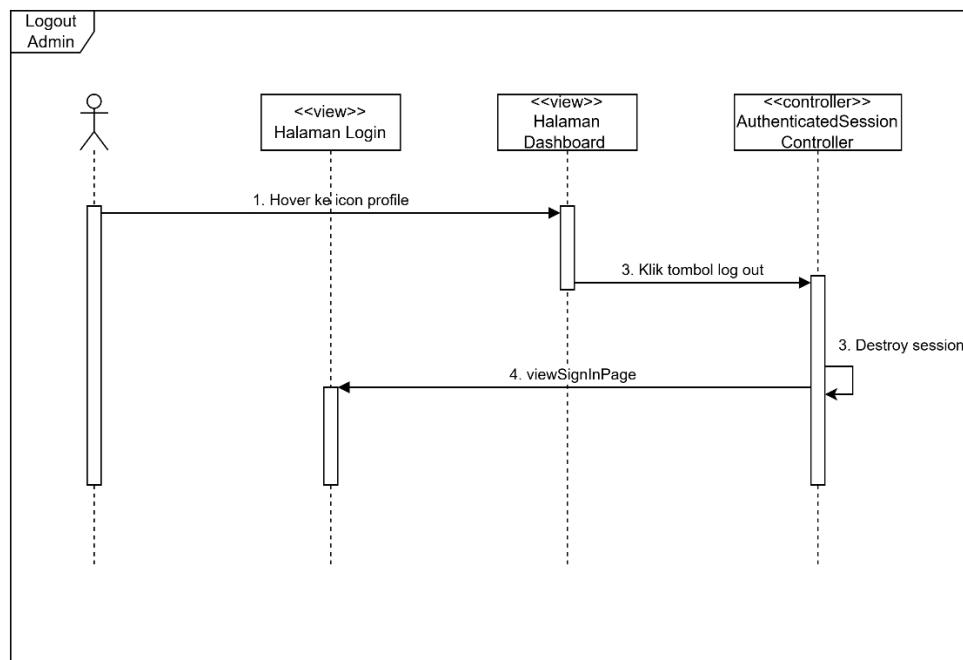
**Gambar 4.2.2. 3 Sequence Diagram Logout oleh user**

### 4. Logout oleh Admin

Skenario Logout admin tertera pada tabel 4.2.2.4 dan gambar sequencenya tertera pada gambar 4.2.2.4.

**Tabel 4.2.2. 4 Skenario *Logout* oleh admin**

Aktor	Admin
Nama Skenario	Melakukan <i>logout</i> dengan akun admin
<i>Flow of events</i> (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. User <i>hover</i> atau mengarahkan cursor ke tombol ikon <i>profile</i> pada <i>dashboard</i></li> <li>2. User menekan tombol <i>log out</i></li> <li>3. Sistem melakukan <i>destroy session</i></li> <li>4. Sistem menampilkan <i>landing page</i></li> </ol>
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	3.1.1 Sistem menampilkan pesan gagal <i>log out</i> .



**Gambar 4.2.2. 4 Sequence Diagram *Logout* oleh admin**

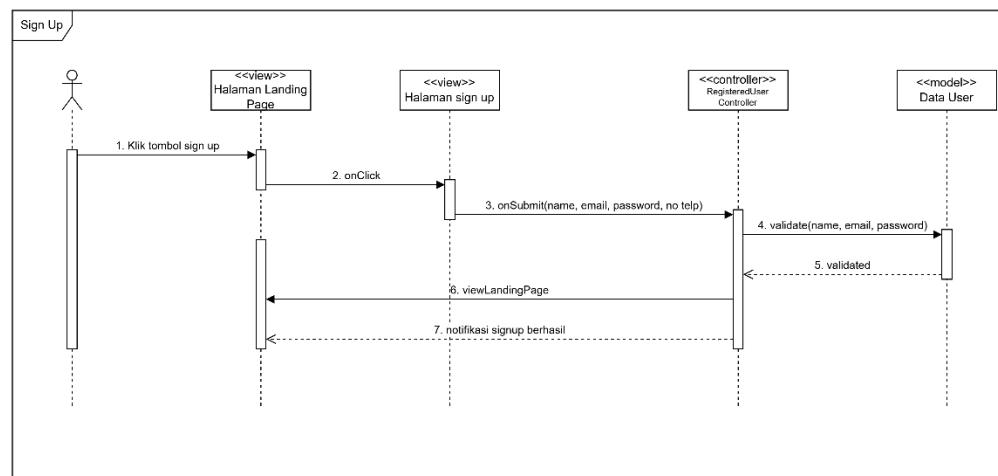
## 5. Register atau Sign Up User

Skenario Register atau Sign Up oleh user tertera pada tabel 4.2.2.5 dan gambar sequencenya tertera pada gambar 4.2.2.5.

**Tabel 4.2.2. 5 Skenario *Register* atau *Sign Up* oleh user**

Aktor	User
Nama Skenario	Melakukan registrasi untuk akun user
<i>Flow of events</i> (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. User mengeklik tombol <i>sign up</i> pada</li> </ol>

	<p><i>landing page</i></p> <ol style="list-style-type: none"> <li>2. Sistem menampilkan halaman registrasi</li> <li>3. User memasukkan email, password, no telp. dan username pada form di halaman registrasi</li> <li>4. User menekan tombol <i>Sign Up</i></li> <li>5. Sistem akan melakukan proses validasi email dan password yang dimasukkan.</li> <li>6. Validasi yang dilakukan sistem berhasil. Username dan password yang dimasukkan benar. Akun telah dibuat dan role akun adalah user.</li> <li>8. Sistem menampilkan <i>landing page</i></li> </ol>
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	<p>5.1.1 Validasi yang dilakukan sistem gagal. email atau password yang dimasukkan salah atau tidak memenuhi ketentuan..</p> <p>5.1.2 Sistem menampilkan pesan gagal <i>sign up</i>.</p>



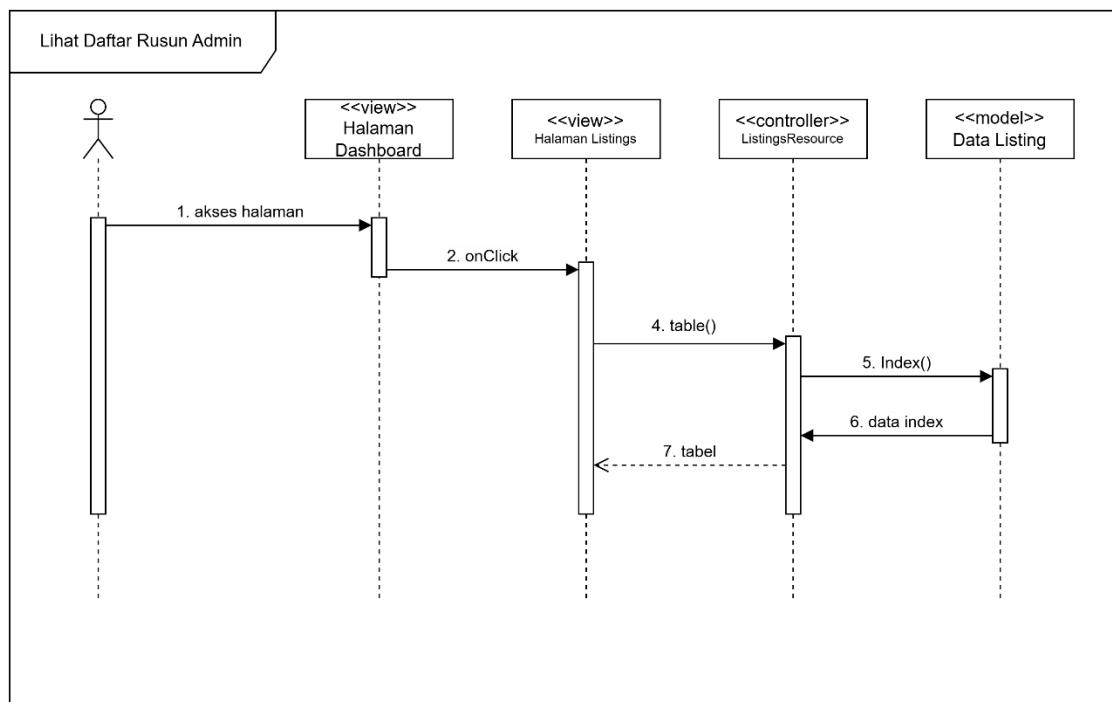
Gambar 4.2.2. 5 Sequence Diagram register user

## 6. Lihat List Rusun oleh Admin

Skenario lihat list rusun oleh admin tertera pada tabel 4.2.2.6 dan gambar sequencenya tertera pada gambar 4.2.2.6.

**Tabel 4.2.2. 6 Skenario melihat *List rusun* untuk akun admin**

Aktor	Admin
Nama Skenario	Melihat <i>List rusun</i> untuk akun admin
<i>Flow of events</i> (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. Admin mengakses halaman <i>dashboard</i></li> <li>2. Sistem menampilkan halaman <i>dashboard</i></li> <li>3. Admin menekan menu <i>listings</i></li> <li>4. Sistem akan menampilkan daftar rusun</li> </ol>
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	4.1.1 Sistem menampilkan pesan gagal menampilkan daftar rusun



**Gambar 4.2.2. 6 Sequence Diagram Lihat *List Rusun* oleh admin**

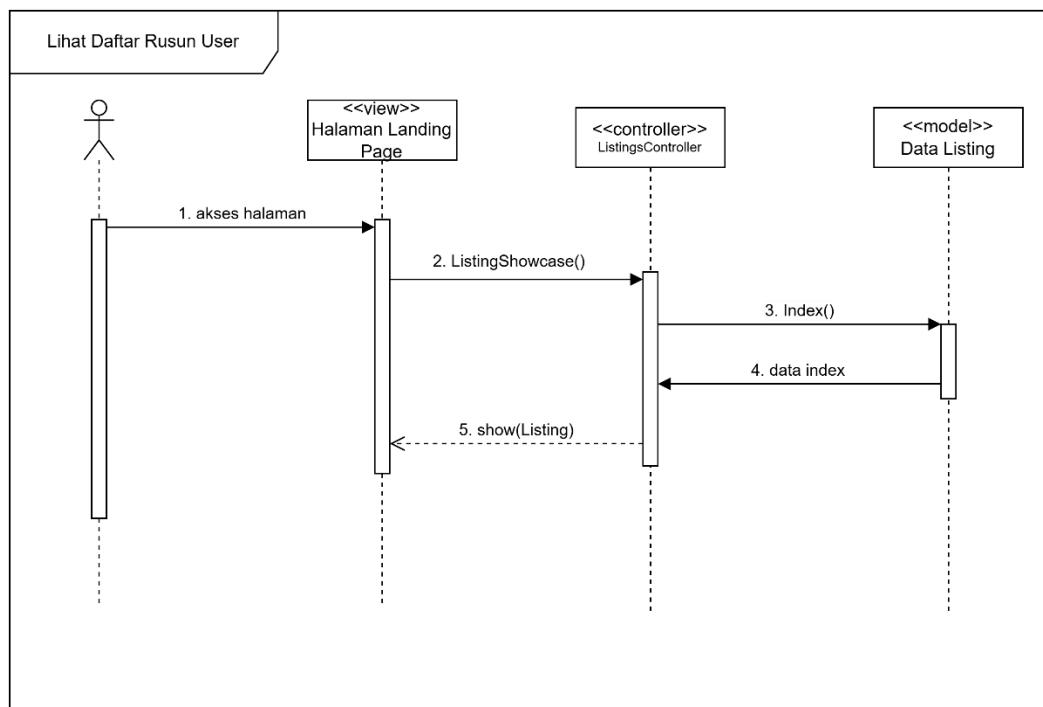
## 7. Lihat *List Rusun* oleh User

Skenario lihat list rusun oleh user tertera pada tabel 4.2.2.7 dan gambar sequencenya tertera pada gambar 4.2.2.7.

**Tabel 4.2.2. 7 Skenario melihat *List rusun* untuk akun user**

Aktor	User
-------	------

Nama Skenario	Melihat <i>List</i> rusun untuk akun user
<i>Flow of events</i> (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. User mengakses halaman <i>landing page</i></li> <li>2. Sistem menampilkan daftar rusun pada <i>landing page</i></li> </ol>
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	2.1.1 Sistem menampilkan pesan gagal menampilkan daftar rusun



Gambar 4.2.2. 7 Sequence Diagram Lihat List Rusun oleh User

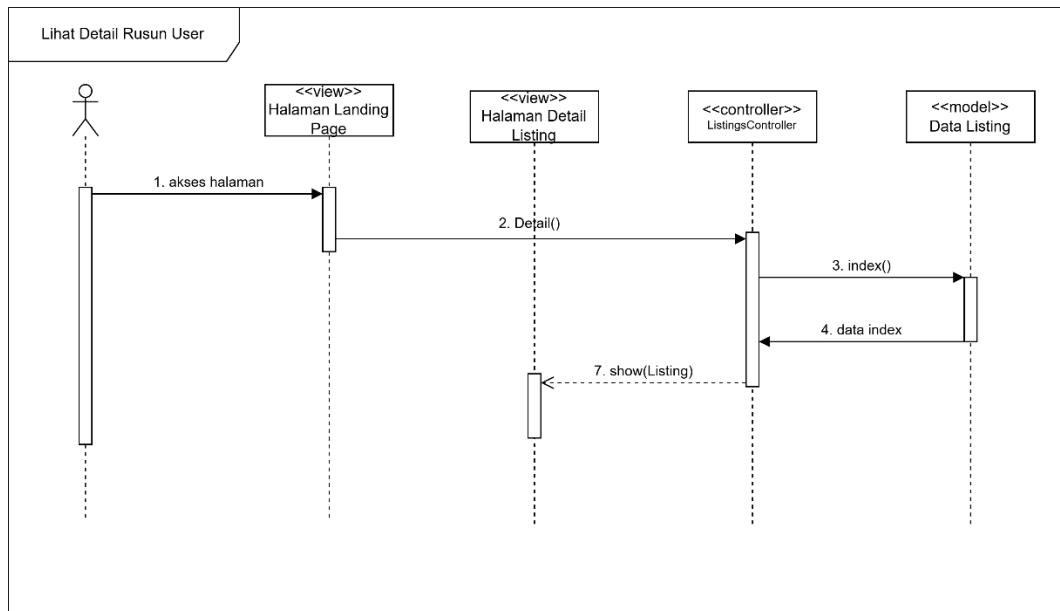
## 8. Lihat Detail Rusun oleh User

Skenario lihat detail rusun oleh user tertera pada tabel 4.2.2.8 dan gambar sequencenya tertera pada gambar 4.2.2.8.

Tabel 4.2.2. 8 Skenario melihat detail rusun untuk akun user

Aktor	User
Nama Skenario	Melihat detail rusun untuk akun user
<i>Flow of events</i> (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. User mengakses halaman <i>landing page</i></li> <li>2. User menekan salah satu rusun pada daftar rusun yang berada pada <i>landing page</i></li> </ol>

	3. Sistem akan menampilkan halaman detail rusun
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	3.1.1 Sistem menampilkan pesan gagal menampilkan detail rusun



Gambar 4.2.2. 8 Sequence Diagram Lihat Detail Rusun oleh user

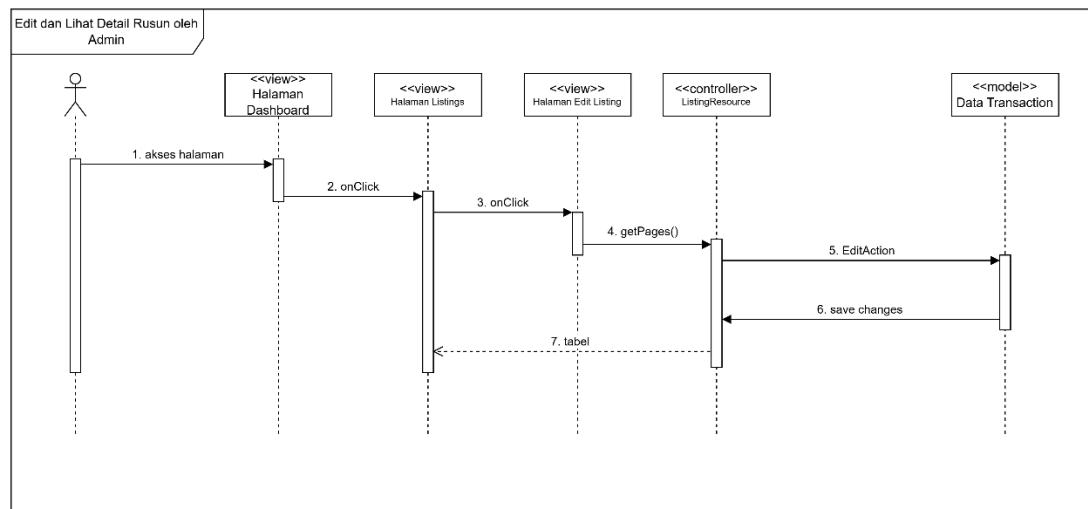
## 9. Lihat Detail dan Ubah Detail Rusun oleh Admin

Skenario lihat detail dan ubah detail rusun oleh admin tertera pada tabel 4.2.2.9 dan gambar sequencenya tertera pada gambar 4.2.2.9.

Tabel 4.2.2. 9 Skenario melihat detail rusun dan edit rusun untuk akun admin

Aktor	Admin
Nama Skenario	Melihat detail rusun dan edit rusun untuk akun admin
Flow of events (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. Admin mengakses halaman <i>dashboard</i></li> <li>2. Admin menekan menu <i>listings</i></li> <li>3. Admin menampilkan halaman <i>listings</i></li> <li>4. Sistem akan menampilkan daftar rusun</li> <li>5. Admin menekan edit pada salah satu rusun di dalam daftar rusun</li> <li>6. Sistem akan menampilkan halaman edit <i>listing</i></li> </ol>

	<p>7. Admin mengedit detail rusun      8. Admin menekan tombol <i>save changes</i>      9. Sistem berhasil menyimpan perubahan      10. Sistem akan menampilkan detail rusun yang sama pada halaman <i>listings</i> dengan perubahan yang dilakukan</p>
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	9.1.1 Sistem menampilkan pesan gagal menyimpan perubahan.



**Gambar 4.2.2. 9 Sequence Diagram Lihat Detail dan Ubah Detail Rusun oleh admin**

## 10. Lihat Daftar Transaksi oleh User

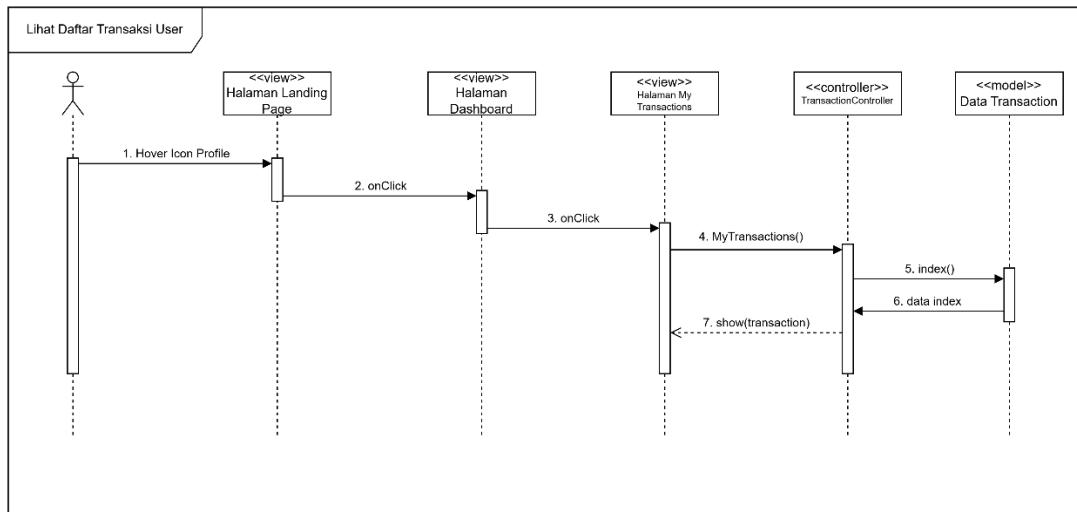
Skenario lihat daftar transaksi oleh user tertera pada tabel 4.2.2.10 dan gambar sequencenya tertera pada gambar 4.2.2.10.

**Tabel 4.2.2. 10 Skenario melihat daftar transaksi untuk akun user**

Aktor	User
Nama Skenario	Melihat daftar transaksi untuk akun user
<i>Flow of events</i> (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. User mengakses halaman <i>landing page</i></li> <li>2. User menekan menu <i>dashboard</i></li> <li>3. Sistem akan menampilkan halaman <i>dashboard</i></li> <li>4. User menekan menu <i>my transaction</i></li> <li>5. Sistem akan menampilkan daftar transaksi yang dilakukan user</li> </ol>
<i>Alternate Scenario</i>	-

### Exceptional Scenario

5.1.1 Sistem menampilkan pesan gagal menampilkan daftar transaksi



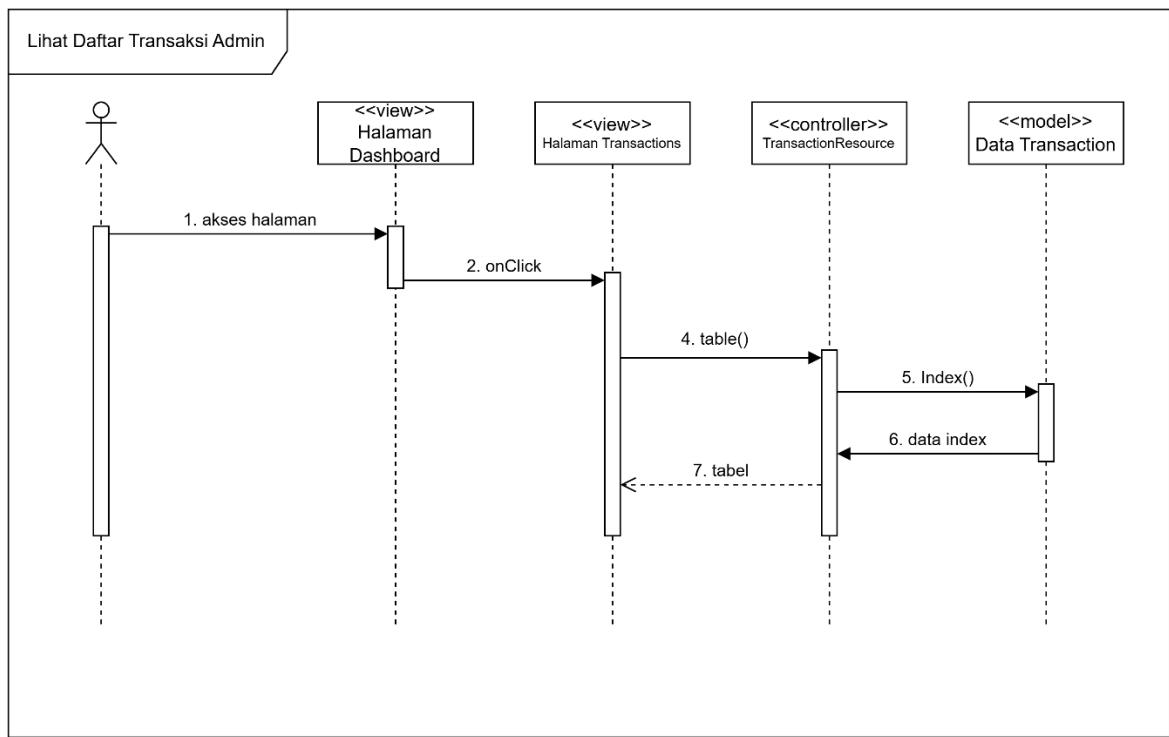
Gambar 4.2.2. 10 Sequence Diagram Lihat Daftar Transaksi oleh user

## 11. Lihat Daftar Transaksi oleh Admin

Skenario lihat daftar transaksi oleh admin tertera pada tabel 4.2.2.11 dan gambar sequencenya tertera pada gambar 4.2.2.11.

Tabel 4.2.2. 11 Skenario melihat daftar transaksi untuk akun admin

Aktor	Admin
Nama Skenario	Melihat daftar transaksi untuk akun admin
Flow of events (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. Admin mengakses halaman <i>dashboard</i></li> <li>2. Admin menekan menu <i>transactions</i></li> <li>3. Sistem akan menampilkan halaman <i>transaction</i></li> <li>4. Sistem akan menampilkan daftar transaksi</li> </ol>
Alternate Scenario	-
Exceptional Scenario	4.1.1 Sistem menampilkan pesan gagal menampilkan daftar transaksi



**Gambar 4.2.2. 11 Sequence Diagram Lihat Daftar Transaksi oleh admin**

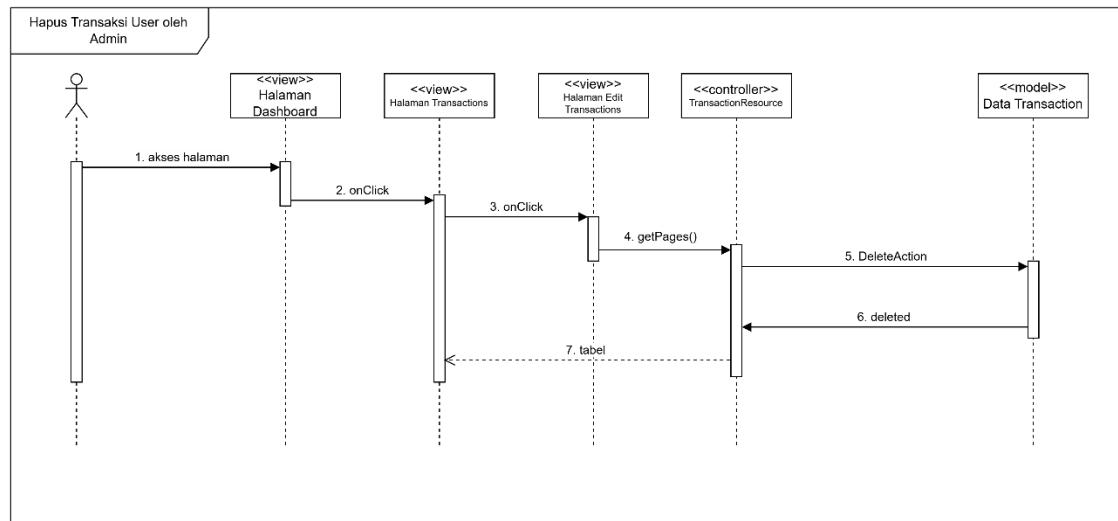
## 12. Hapus Transaksi

Skenario hapus transaksi oleh admin tertera pada tabel 4.2.2.12 dan gambar sequencenya tertera pada gambar 4.2.2.12.

**Tabel 4.2.2. 12 Skenario menghapus transaksi user untuk akun admin**

Aktor	Admin
Nama Skenario	Menghapus transaksi user untuk akun admin
Flow of events (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. Admin mengakses halaman <i>dashboard</i></li> <li>2. Admin menekan menu <i>transactions</i></li> <li>3. Sistem akan menampilkan halaman <i>transaction</i></li> <li>4. Sistem akan menampilkan daftar transaksi</li> <li>5. Admin memilih atau menekan salah satu transaksi di dalam daftar transaksi</li> <li>6. Sistem akan menampilkan halaman edit transaksi</li> <li>7. Admin menekan tombol <i>delete</i></li> <li>8. Sistem menampilkan <i>pop up</i> konfirmasi</li> <li>9. Admin menekan tombol <i>confirm</i></li> <li>10. Sistem berhasil menyimpan perubahan</li> <li>11. Sistem tidak akan menampilkan</li> </ol>

	transaksi yang sudah dihapus pada halaman <i>transactions</i>
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	10.1.1 Sistem menampilkan pesan gagal menghapus transaksi



Gambar 4.2.2. 12 Sequence Diagram Hapus Transaksi oleh admin

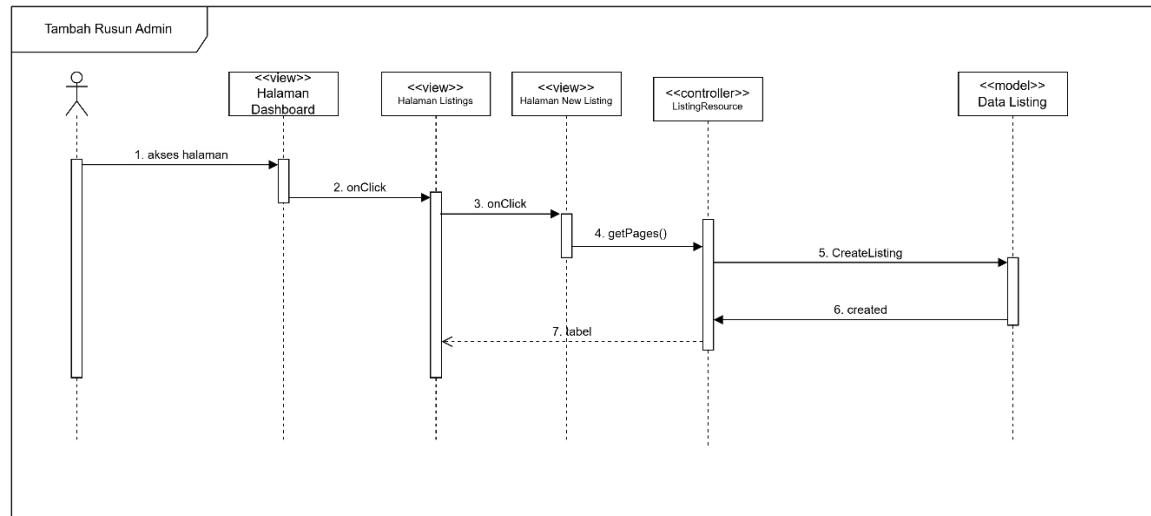
### 13. Tambah Rusun

Skenario tambah rusun oleh admin tertera pada tabel 4.2.2.13 dan gambar sequencencnya tertera pada gambar 4.2.2.13.

Tabel 4.2.2. 13 Skenario menambah rusun untuk akun admin

Aktor	Admin
Nama Skenario	Menambah rusun untuk akun admin
Flow of events (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. Admin mengakses halaman <i>dashboard</i></li> <li>2. Sistem menampilkan halaman <i>dashboard</i></li> <li>3. Admin menekan menu <i>listings</i></li> <li>4. Sistem akan menampilkan daftar rusun</li> <li>5. Admin menekan tombol <i>new listing</i></li> <li>6. Sistem akan menampilkan halaman <i>new listing</i></li> <li>7. Admin mengisi <i>form</i> sesuai ketentuan</li> <li>8. Admin menekan tombol <i>create new listing</i></li> <li>9. Sistem berhasil menambahkan rusun</li> </ol>

	baru 10. Sistem akan menampilkan rusun baru pada halaman <i>listings</i>
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	9.1.1 Sistem menampilkan pesan gagal menambah <i>new listing</i>



Gambar 4.2.2. 13 Sequence Diagram Tambah Rusun oleh admin

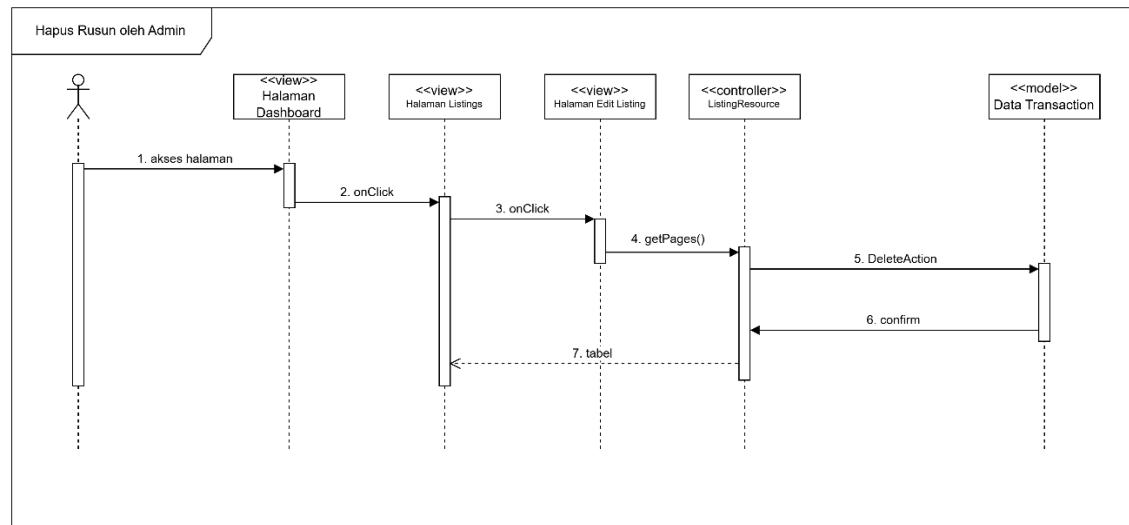
#### 14. Hapus Rusun

Skenario hapus rusun oleh admin tertera pada tabel 4.2.2.14 dan gambar sequencenya tertera pada gambar 4.2.2.14.

Tabel 4.2.2. 14 Skenario menghapus rusun untuk akun admin

Aktor	Admin
Nama Skenario	menghapus rusun untuk akun admin
Flow of events (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. Admin mengakses halaman <i>dashboard</i></li> <li>2. Sistem menampilkan halaman <i>dashboard</i></li> <li>3. Admin menekan menu <i>listings</i></li> <li>4. Sistem akan menampilkan daftar rusun</li> <li>5. Admin menekan edit pada salah satu rusun di dalam daftar rusun</li> <li>6. Sistem akan menampilkan halaman edit <i>listing</i></li> <li>7. Admin menekan tombol <i>delete</i></li> <li>8. Sistem menampilkan pop up</li> </ol>

	<p>konfirmasi</p> <p>9. Admin menekan tombol <i>confirm</i></p> <p>10. Sistem berhasil menghapus rusun</p> <p>11. Sistem tidak akan menampilkan rusun yang sudah dihapus pada halaman <i>listings</i></p>
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	10.1.1 Sistem menampilkan pesan gagal menambah <i>new listing</i>



Gambar 4.2.2. 14 Sequence Diagram Hapus Rusun oleh admin

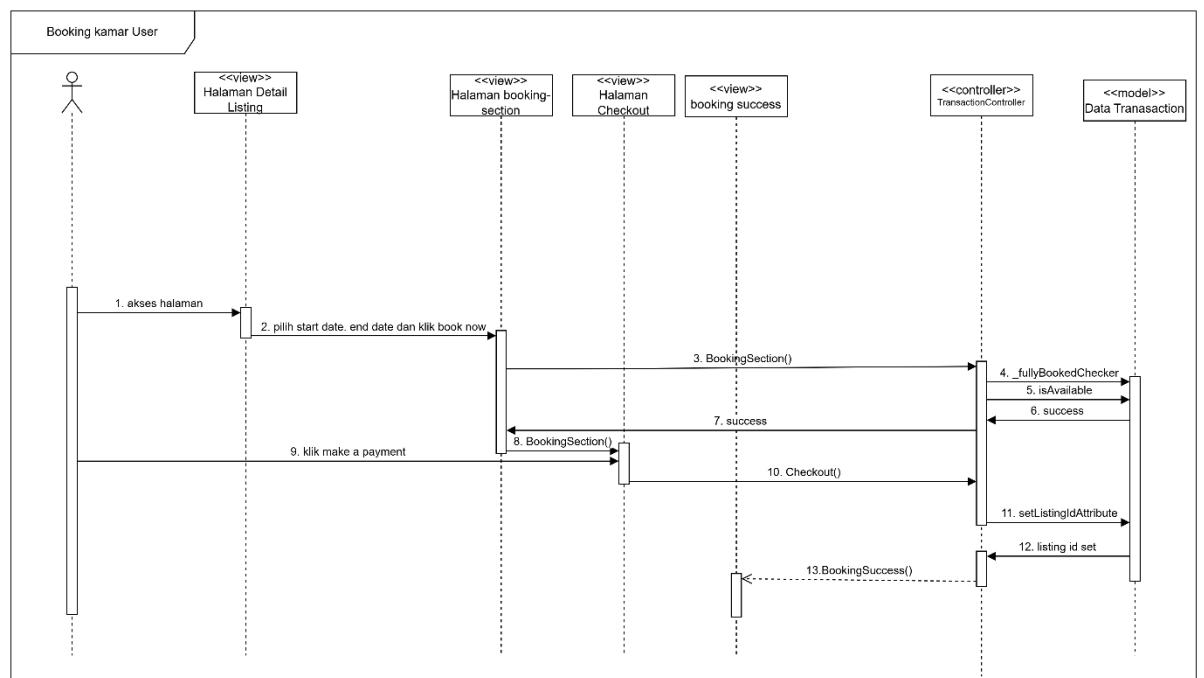
## 15. Booking Kamar

Skenario booking kamar oleh user tertera pada tabel 4.2.2.15 dan gambar sequencenya tertera pada gambar 4.2.2.15.

Tabel 4.2.2. 15 Skenario membooking kamar untuk akun user

Aktor	User
Nama Skenario	membooking kamar untuk akun user
Flow of events (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. User mengakses halaman detail <i>listing</i></li> <li>2. User memasukkan inputan start date dan end date pada elemen booking section di halaman detail <i>listing</i> dan menekan tombol <i>book now</i></li> <li>3. Sistem memvalidasi apakah ada ketersediaan kamar di rusun pada database</li> <li>4. Sistem mengirimkan respon sukses</li> <li>5. Sistem otomatis <i>redirect</i> ke halaman</li> </ol>

	<p><i>checkout</i></p> <ol style="list-style-type: none"> <li>6. User menekan tombol <i>make a payment</i></li> <li>7. Sistem memasukkan transaksi baru ke database</li> <li>8. Sistem menampilkan halaman <i>booking success</i></li> </ol>
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	<ol style="list-style-type: none"> <li>4.1.1 Sistem mengirimkan respon failed</li> <li>4.1.2 Sistem menampilkan pesan rusun penuh</li> <li>7.1.1 Sistem menampilkan pesan gagal memproses transaksi</li> </ol>



Gambar 4.2.2. 15 . Sequence Diagram Booking Kamar

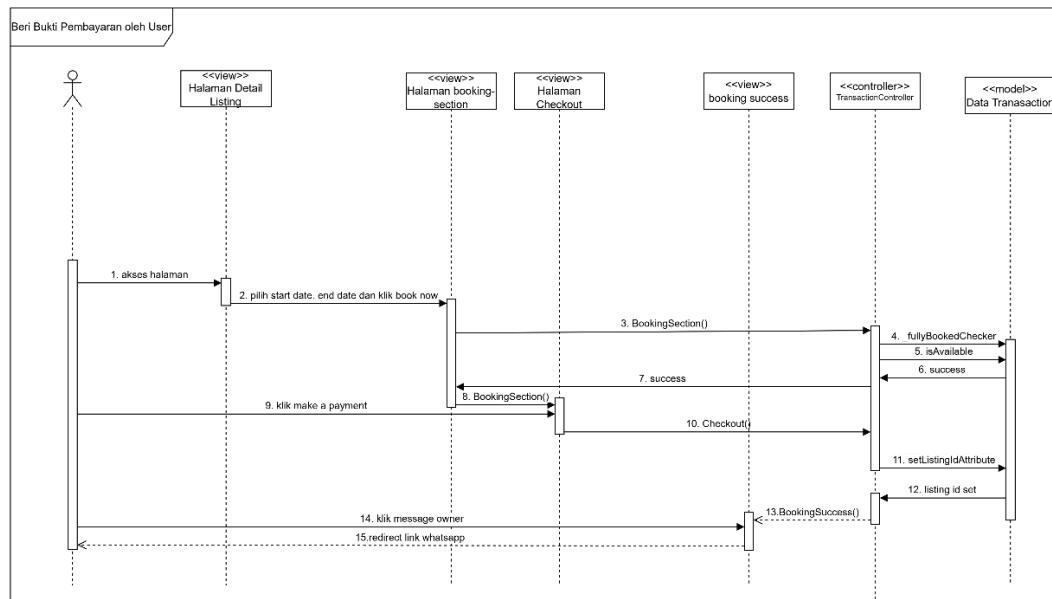
## 16. Beri Bukti Pembayaran

Skenario beri bukti pembayaran oleh user tertera pada tabel 4.2.2.16 dan gambar sequencenya tertera pada gambar 4.2.2.16.

Tabel 4.2.2. 16 Skenario beri bukti pembayaran untuk akun user

Aktor	User
Nama Skenario	membooking kamar untuk akun user
<i>Flow of events</i> (Urutan kejadian)	<ol style="list-style-type: none"> <li>9. User mengakses halaman detail <i>listing</i></li> <li>10. User memasukkan inputan start date dan end date pada elemen booking</li> </ol>

	<p>section di halaman detail <i>listing</i> dan menekan tombol <i>book now</i></p> <p>11. Sistem memvalidasi apakah ada ketersediaan kamar di rusun pada database</p> <p>12. Sistem mengirimkan respon sukses</p> <p>13. Sistem otomatis <i>redirect</i> ke halaman <i>checkout</i></p> <p>14. user menekan tombol <i>make a payment</i></p> <p>15. Sistem memasukkan transaksi baru ke database</p> <p>16. Sistem menampilkan halaman <i>booking success</i></p> <p>17. User menekan tombol <i>message owner</i></p> <p>18. Sistem me-<i>redirect</i> ke whatsapp pemilik rusun</p>
<i>Alternate Scenario</i>	-
<i>Exceptional Scenario</i>	<p>4.1.1 Sistem mengirimkan respon failed</p> <p>4.1.2 Sistem menampilkan pesan rusun penuh</p> <p>7.1.1 Sistem menampilkan pesan gagal memproses transaksi</p> <p>18.1.1 Sistem gagal redirect ke whatsapp</p>



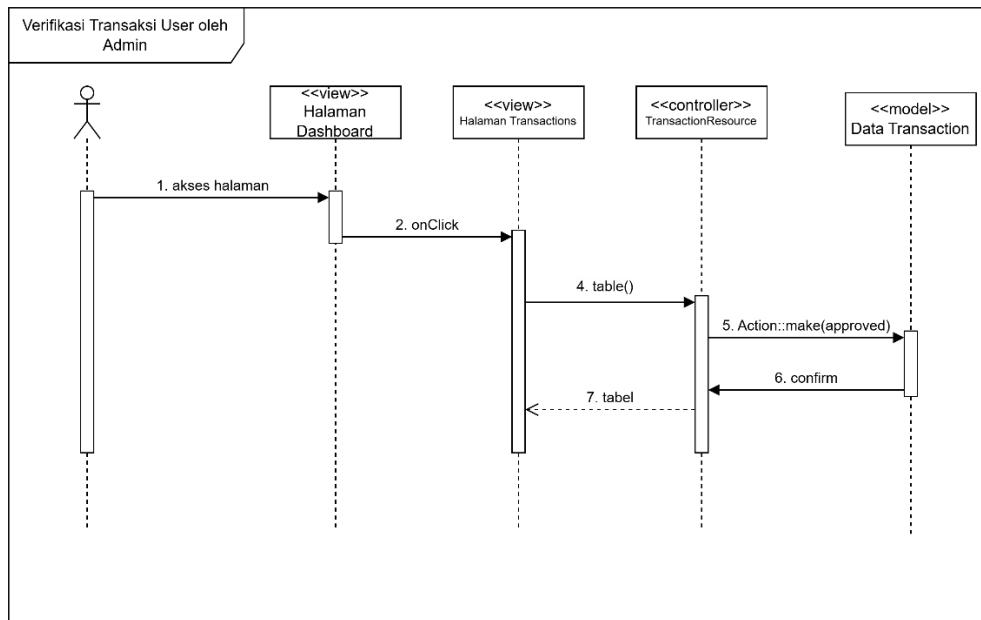
Gambar 4.2.2. 16 Sequence Diagram Beri Bukti Pembayaran

## 17. Verifikasi Pembookingan Kamar

Skenario verifikasi pembookingan kamar oleh admin tertera pada tabel 4.2.2.17 dan gambar sequencenya tertera pada gambar 4.2.2.17.

**Tabel 4.2.2. 17 Skenario verifikasi pembookingan kamar oleh admin**

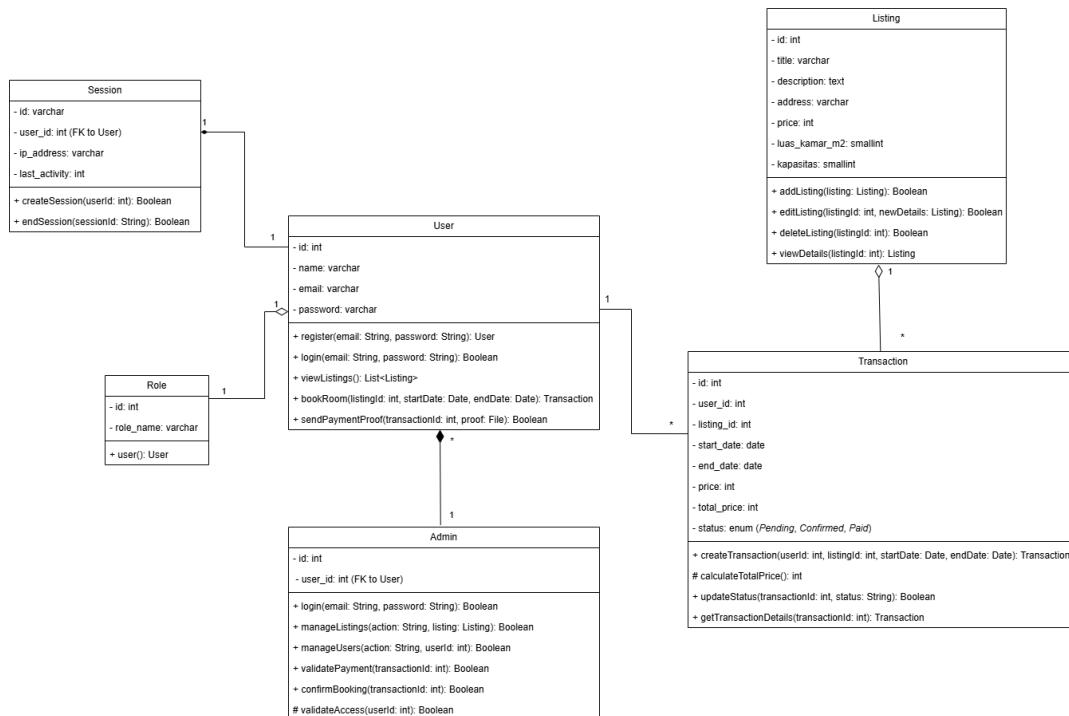
Aktor	Admin
Nama Skenario	Verifikasi pembookingan kamar oleh admin
Flow of events (Urutan kejadian)	<ol style="list-style-type: none"> <li>1. Admin mengakses halaman <i>dashboard</i></li> <li>2. Admin menekan menu <i>transactions</i></li> <li>3. Sistem akan menampilkan halaman <i>transaction</i></li> <li>4. Sistem akan menampilkan daftar transaksi</li> <li>5. Admin menekan tombol approve</li> <li>6. Sistem menampilkan pop up konfirmasi</li> <li>7. Admin menekan tombol confirm</li> <li>8. Sistem berhasil menyimpan konfirmasi</li> <li>9. Sistem akan menampilkan status transaksi dari waiting menjadi approved</li> </ol>
Alternate Scenario	-
Exceptional Scenario	8.1.1 Sistem menampilkan pesan gagal approve



**Gambar 4.2.2. 17 Sequence Diagram Verifikasi Pembookingan Kamar oleh admin**

### 4.2.3 Diagram Class

Class diagram pada gambar 4.2.3.1 menggambarkan sistem yang terdiri dari entitas utama seperti User, Admin, Transaction, Listing, Role, dan Session. Entitas User menjadi pusat sistem dengan kemampuan mengelola sesi melalui Session, memesan kamar melalui Transaction, dan memiliki peran yang ditentukan oleh Role. Entitas Admin, sebagai turunan dari User, memiliki hak khusus untuk mengelola pengguna dan listing. Relasi antara Transaction dan Listing menunjukkan satu transaksi terkait dengan satu listing, sementara satu listing dapat memiliki banyak transaksi.



Gambar 4.2.3. 1 Diagram Class

Simbol dalam class diagram dalam Gambar 4.2.3.1 memiliki arti sebagai berikut:

- Pada Session, variabel seperti id, user\_id, ip\_address, dan last\_activity bersifat private (-) untuk menjaga keamanan data sesi pengguna. Dengan metode public (+) seperti createSession dan endSession, sistem dapat mengelola sesi pengguna secara aman.

2. User memiliki variabel private (-) seperti id, name, email, dan password untuk melindungi informasi sensitif pengguna. Metode seperti register, login, viewListings, dan bookRoom bersifat public (+) agar dapat diakses oleh sistem lain, seperti *User Interface* atau API.
3. Role menentukan hak akses pengguna melalui variabel private (-) seperti id dan role\_name, sehingga hanya dapat diakses secara internal dalam class.
4. Listing memiliki variabel private (-) seperti id, title, description, price, dan lainnya untuk menjaga integritas data properti. Metode seperti addListing, editListing, dan deleteListing bersifat public (+) agar admin dapat mengelola data properti.
5. Transaction bertanggung jawab atas pemesanan properti, dengan variabel seperti id, user\_id, listing\_id, start\_date, end\_date, dan status yang semuanya bersifat *private* (-) untuk mencegah manipulasi langsung. Metode seperti createTransaction dan updateStatus bersifat public (+) untuk memfasilitasi proses transaksi.
6. Admin, sebagai turunan dari User, memiliki variabel seperti id dan user\_id yang bersifat *private* (-), sedangkan metode seperti manageListings dan validatePayment bersifat *public* (+) untuk memungkinkan pengelolaan sistem. *Protected* (#) digunakan dalam beberapa metode khusus, seperti validasi akses (validateAccess), yang hanya diperlukan oleh admin dan tidak boleh diakses oleh entitas lain.

Simbol ini memastikan bahwa setiap atribut dan metode hanya dapat diakses sesuai dengan tingkat akses yang dibutuhkan, menjaga keamanan dan keteraturan sistem sesuai kebutuhan pengembangan aplikasi. Dengan struktur ini, sistem menjadi lebih aman, modular, dan mudah dikembangkan.

#### 4.2.4 Perancangan Antarmuka

##### 4.2.4.1 Antarmuka User

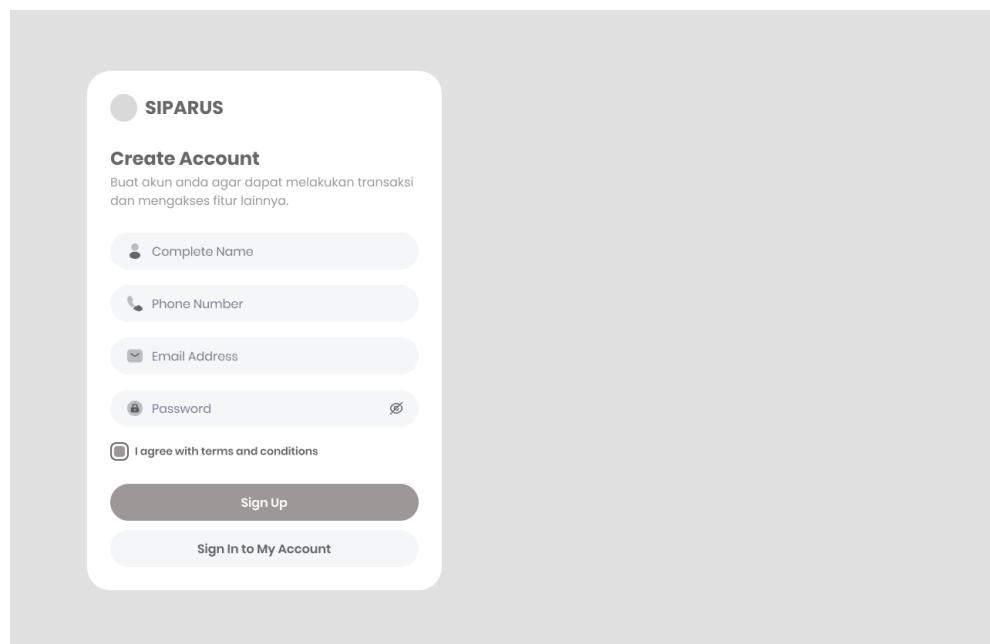
1. *Wireframe* antarmuka *Sign Up*

Gambar 4.2.4.1.1 menunjukkan halaman *Sign Up* User yang akan memuat *form* isian, diantranya nama lengkap, nomor telepon, email, dan password, serta sebuah *checkbox* untuk *terms and condition*. Tombol yang dimuat adalah *Sign*

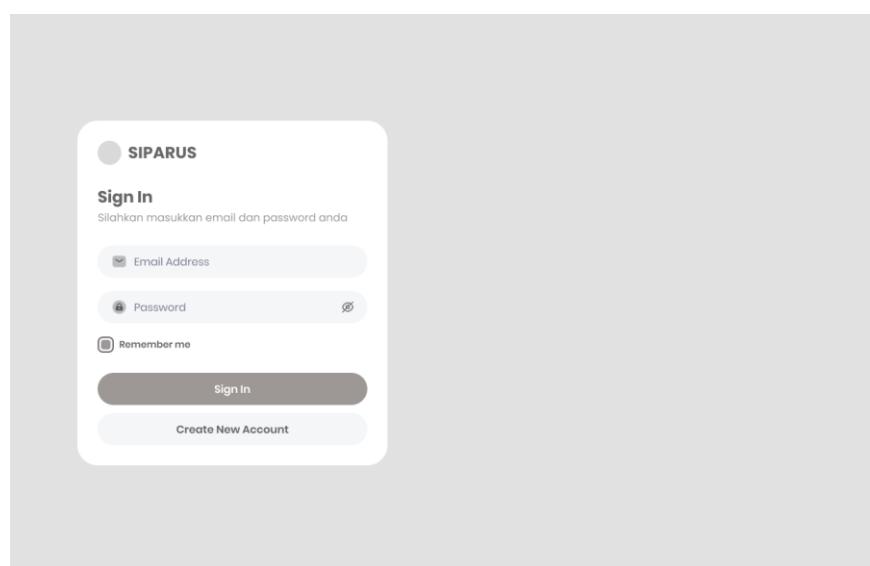
*Up* yang digunakan untuk menginisiasi proses registrasi dan *Sign In to My Account* untuk menuju halaman *Sign In* jika sudah memiliki sebuah akun.

## 2. Wireframe antarmuka *Sign In*

Gambar 4.2.4.1.2 menunjukkan halaman *Sign In* User yang akan memuat *form* isian, diantranya email dan password, serta sebuah *checkbox* untuk *Remember me*. Tombol yang dimuat adalah *Sign In* yang digunakan untuk menginisiasi proses *login* dan *Create New Account* untuk menuju halaman *Sign Up* jika belum memiliki sebuah akun.



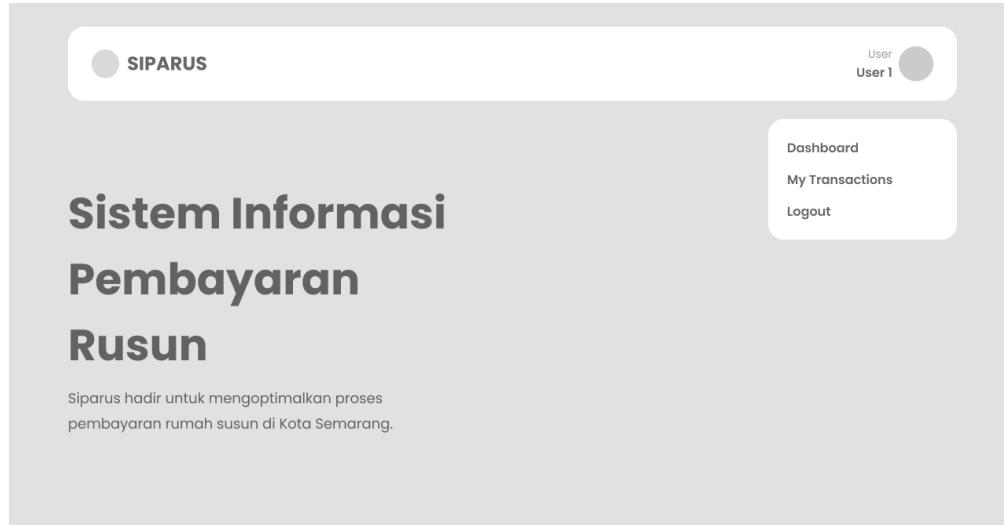
Gambar 4.2.4.1. 1 Wireframe antarmuka *Sign Up*



Gambar 4.2.4.1. 2 Wireframe antarmuka *Sign In*

### 3. Wireframe antarmuka *Logged In*

Gambar 4.2.4.1.3 menunjukkan halaman *Logged In* User yang akan memuat sebuah *bar* yang berisikan judul aplikasi dan nama, serta ikon *Profile User*. Pada *Profile User* akan memuat menu pilihan *Dashboard*, *My Transactions*, dan *Logout*.



Gambar 4.2.4.1. 3 Wireframe antarmuka *Logged In*

### 4. Wireframe antarmuka *Landing Page*

Gambar 4.2.4.1.4 menunjukkan halaman Landing Page User yang sama dengan halaman Logged In, perbedaanya terdapat pada pilihan tombol Sign In dan Sign Up. Halaman juga akan memuat List Rusun, Data Rusun, dan deskripsi fasilitas pada rusun.

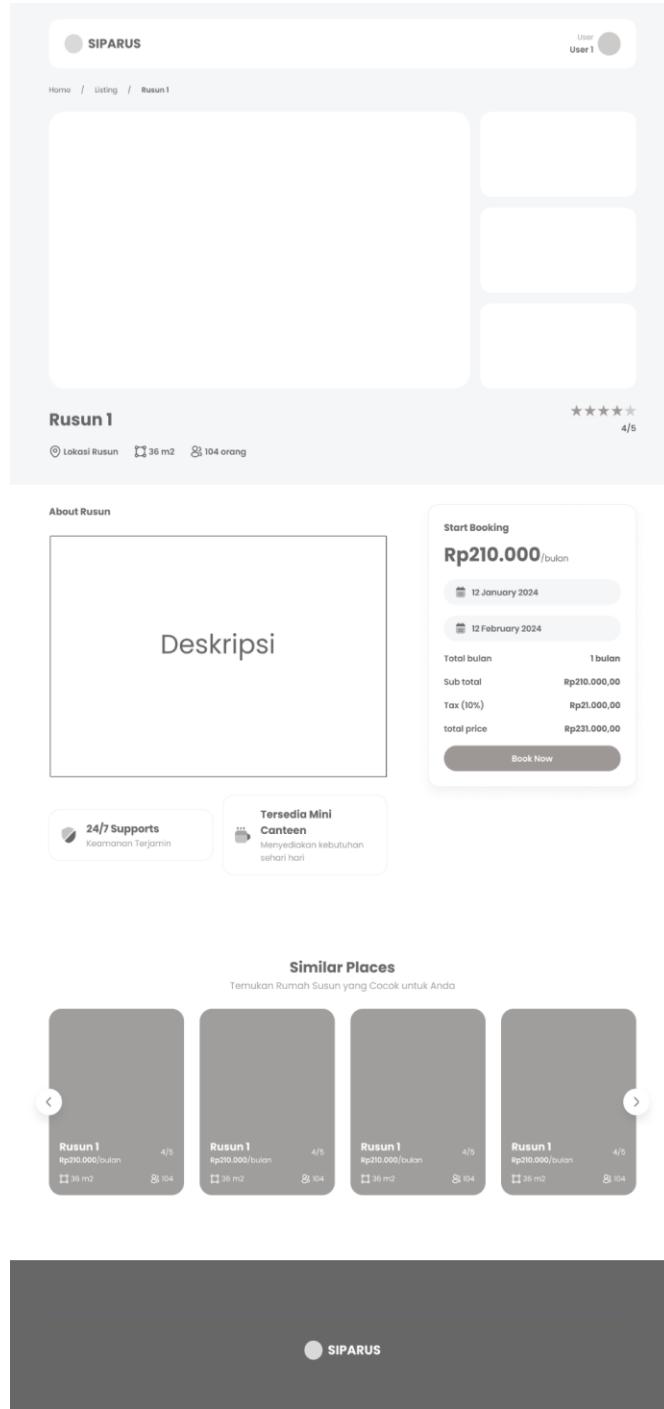


Gambar 4.2.4.1. 4 Wireframe antarmuka Landing Page

##### 5. Wireframe antarmuka Detail Rusun

Gambar 4.2.4.1.5 menunjukkan halaman Detail rusun yang akan memuat gambar dari rusun, deskripsi detail dari rusun, fasilitas dari rusun tersebut, serta list rusun untuk memilih rusun lain. Halaman juga akan memuat card yang

berisikan pilihan *start date* dan *end date* untuk memesan sebuah kamar pada rusun.

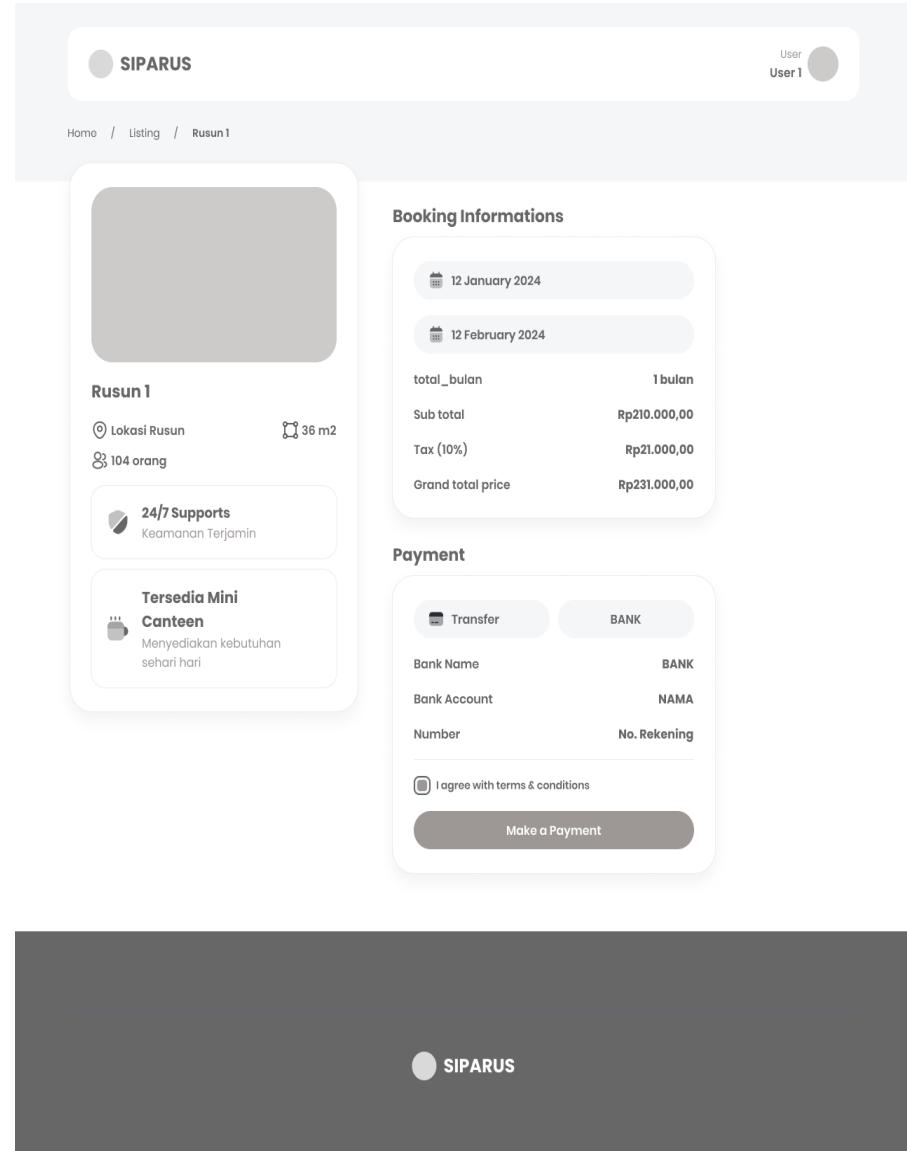


**Gambar 4.2.4.1. 5 Wireframe antarmuka Detail Rusun**

#### 6. Wireframe antarmuka *Checkout*

Gambar 4.2.4.1.6 menunjukkan halaman Halaman *Checkout* yang akan memuat detail pembayaran untuk pembookingan kamar, serta metode pembayaran yang

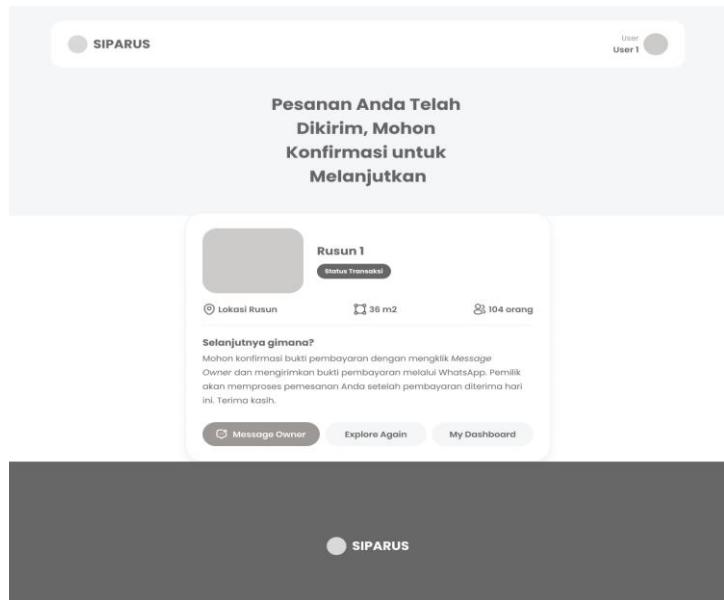
tersedia yaitu transfer bank dan sebuah tombol *make a payment* untuk menginisiasi proses checkout.



Gambar 4.2.4.1. 6 Wireframe antarmuka *Checkout*

#### 7. Wireframe antarmuka Booking Sukses

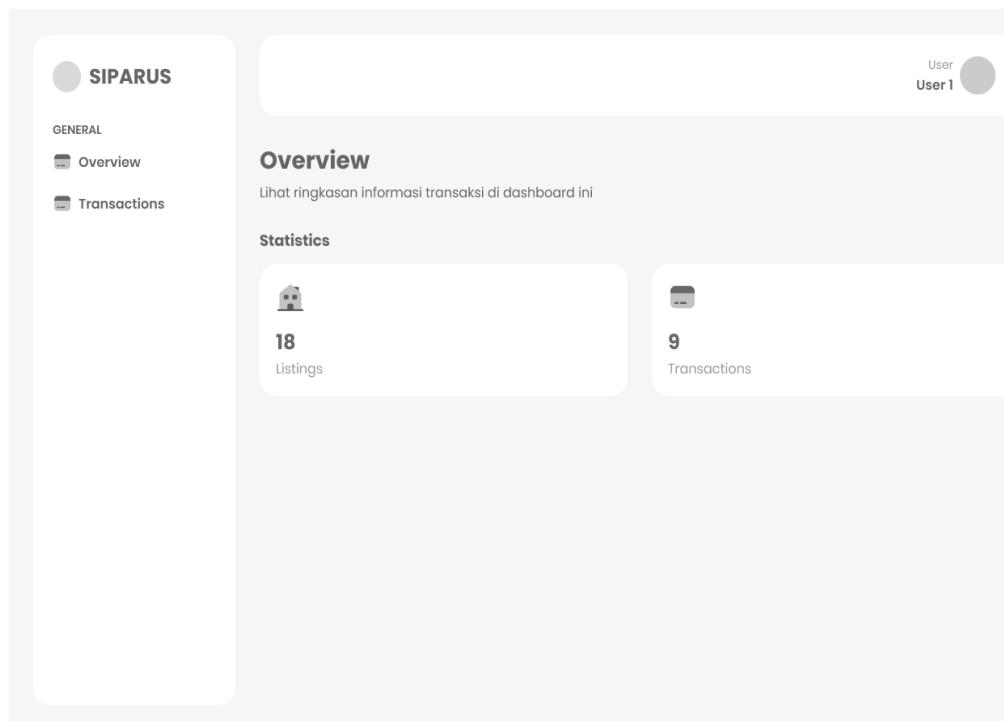
Gambar 4.2.4.1.7 menunjukkan halaman Booking Sukses yang akan memuat status sebuah transaksi serta panduan untuk langkah selanjutnya dalam proses pembookingan. Tombol yang dimuat adalah *Message Owner*, *Explore Again*, dan *My Dahboard*.



**Gambar 4.2.4.1. 7 Wireframe antarmuka Booking Sukses**

#### 8. Wireframe antarmuka *Dashboard Overview*

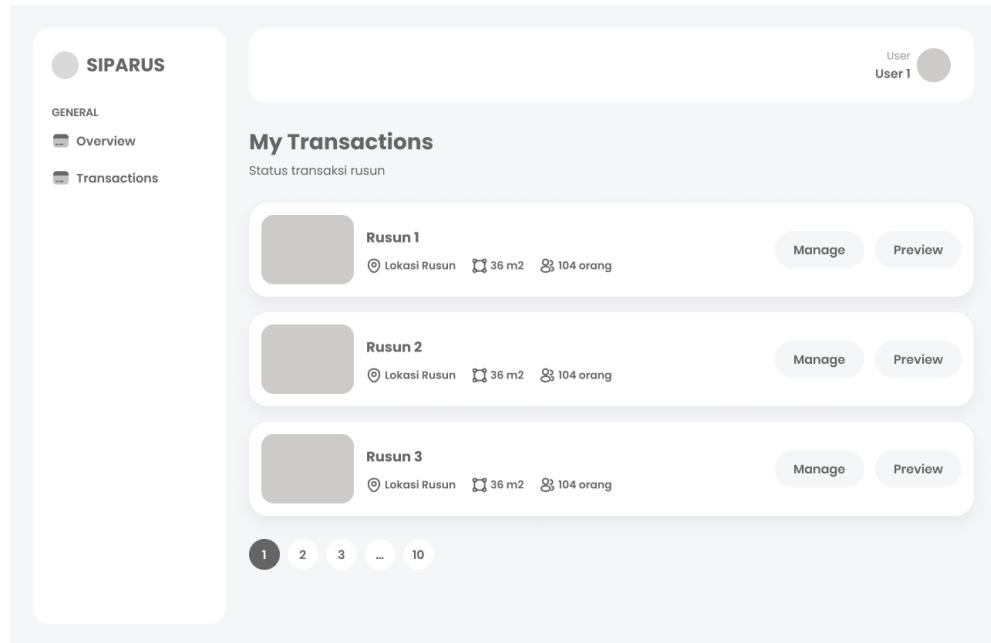
Gambar 4.2.4.1.8 menunjukkan halaman *Dashboard* yang akan memuat 2 menu yaitu *Overview* dan *Transactions*. *Overview* berisikan jumlah banyaknya rusun dan transaksi.



**Gambar 4.2.4.1. 8 Wireframe antarmuka *Dashboard Overview***

#### 9. Wireframe antarmuka *Transactions*

Gambar 4.2.4.1.8 menunjukkan halaman *Transactions* yang akan memuat daftar transaksi dari user serta tombol *Manage* dan *Preview*.

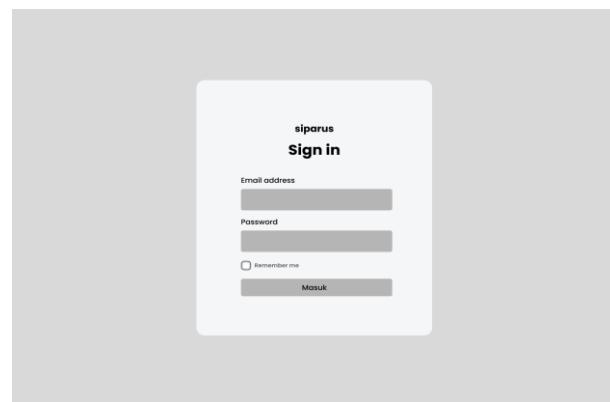


Gambar 4.2.4.1. 9 Wireframe antarmuka *Transactions*

#### 4.2.4.2 Antarmuka Admin

##### 1. Wireframe antarmuka *Sign In*

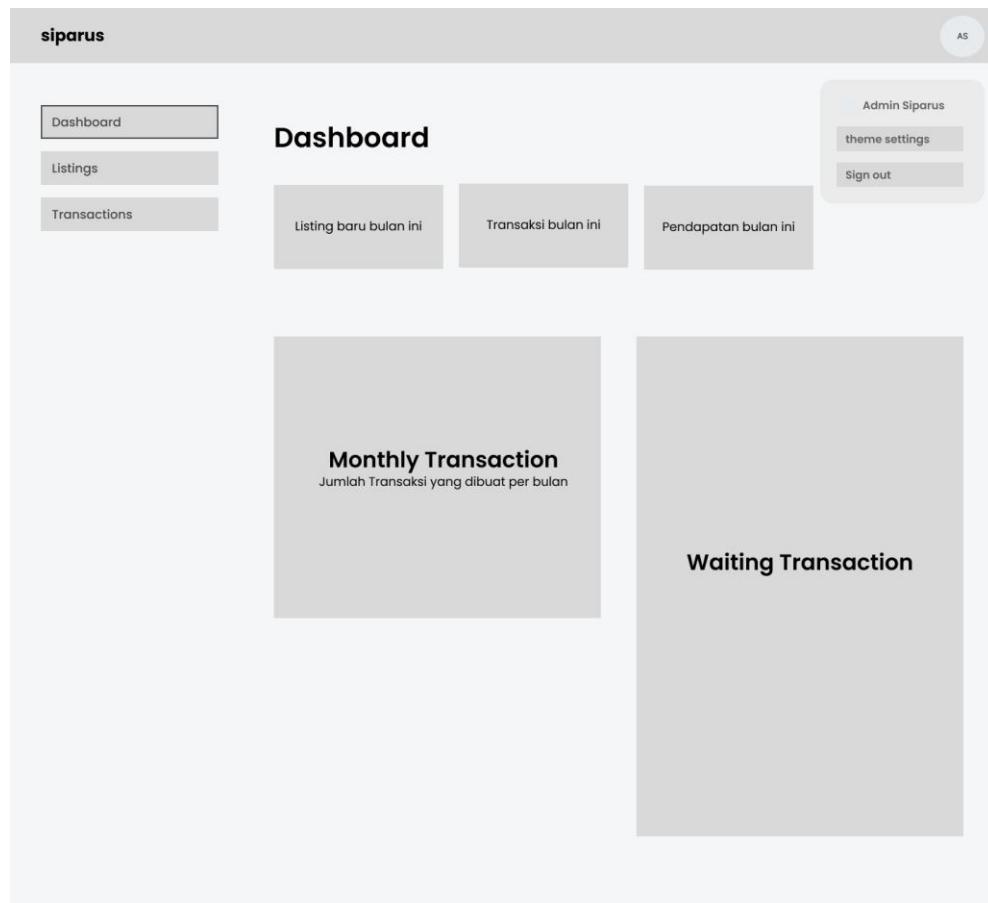
Gambar 4.2.4.2.1 menunjukkan halaman *Sign In* yang akan memuat form isian, yaitu email dan password. Tombol yang dimuat adalah tombol Masuk untuk menginisiasi proses *login* admin.



Gambar 4.2.4.2. 1 Wireframe antarmuka *Sign In*

## 2. Wireframe antarmuka *Dashboard*

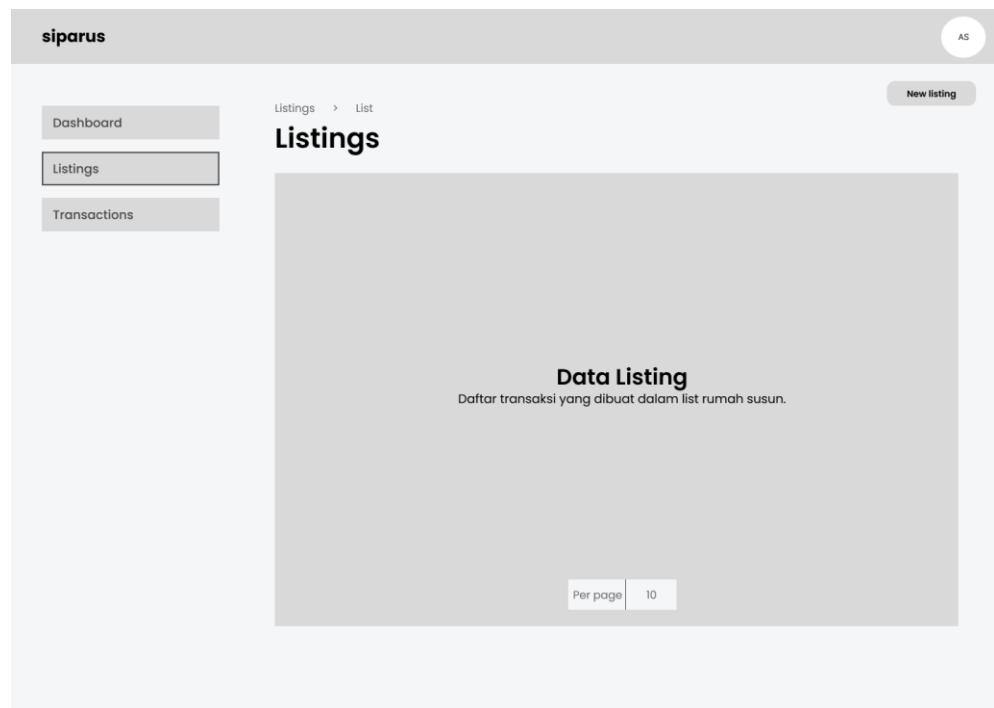
Gambar 4.2.4.2.2 menunjukkan halaman *Dashboard* yang akan memuat menu *Dashboard*, *Listings*, dan *Transactions*. Halaman ini juga memuat menu pada ikon *profile* akun, yaitu *theme settings* dan *sign out*. *Dashboard* sendiri memuat detail perihal transaksi pada aplikasi, diantaranya grafik transaksi, jumlah pendapatan, *listing* baru, dan daftar transaksi yang belum *approved* atau masih dalam status *waiting*.



Gambar 4.2.4.2. 2 Wireframe antarmuka *Dashboard*

## 3. Wireframe antarmuka *Listings*

Gambar 4.2.4.2.3 menunjukkan halaman *Listings* memuat daftar tabel rusun serta tombol *new listing*.



Gambar 4.2.4.2. 3 Wireframe antarmuka *Listings*

#### 4. Wireframe antarmuka *Create Listings*

Gambar 4.2.4.2.3 menunjukkan halaman *Create Listing* akan memuat beberapa isian *form* untuk membuat sebuah listing baru atau rusun baru. Tombol yang dimuat adalah *Create*, *Create & Create Another*, dan *Cancel*.

The wireframe shows a top navigation bar with the brand name 'siparus'. Below it is a sidebar with three buttons: 'Dashboard', 'Listings' (which is highlighted), and 'Transactions'. The main content area has a breadcrumb path 'Listings > Create' and a title 'Create Listings'. The form contains fields for 'Title' (with a 'slug' field to its right), 'Description' (a large text area), 'Address' (with a 'Luas kamar m2' field to its right), 'Kapasitas' (with a 'Price' field to its right), and two checkboxes: 'Full support available' and 'Mini canteen available'. There is also a 'Attachments' field. At the bottom are three buttons: 'Create', 'Create & create another', and 'Cancel'.

Gambar 4.2.4.2. 4 Wireframe antarmuka *Create Listings*

## 5. Wireframe antarmuka *Edit Listings*

Gambar 4.2.4.2.5 menunjukkan halaman *Edit listing* memiliki isian *form* yang sama dengan halaman *Create Listings*. Tombol yang dimuat adalah *Delete*, *Save changes*, dan *Cancel*.

The wireframe shows a sidebar with 'Dashboard', 'Listings' (selected), and 'Transactions'. The main area has a title 'Edit Listings'. It contains fields for 'Title' and 'slug', a large 'Description' text area, 'Address' and 'Luas kamar m2' fields, 'Kapasitas' and 'Price' fields, two checkboxes for 'Full support available' and 'Mini canteen available', and an 'Attachments' section. At the bottom are 'Save Changes' and 'Cancel' buttons.

Gambar 4.2.4.2. 5 Wireframe antarmuka *Edit Listings*

## 6. Wireframe antarmuka *Transactions*

Gambar 4.2.4.2.6 menunjukkan halaman *Transactions* akan memuat daftar tabel yang berisikan transaksi dari berbagai user.

The wireframe shows a sidebar with 'Dashboard', 'Listings', and 'Transactions' (selected). The main area has a title 'Transactions' and a table header 'Data Transactions' with the note 'Daftar transaksi yang perlu disetujui'. At the bottom are 'Per page' and '10' buttons.

Gambar 4.2.4.2. 6 Wireframe antarmuka *Transactions*

## BAB V

### HASIL DAN PEMBAHASAN

#### 5.1 Implementasi

Tahap implementasi berisi hasil analisis dan rancangan yang telah dibuat dengan menggunakan bahasa pemrograman.

##### 5.1.1 Implementasi Sistem

Spesifikasi perangkat lunak dan perangkat keras yang digunakan dalam pengembangan aplikasi SIPARUS adalah sebagai berikut:

1. CPU : Asus 11th Gen Intel(R)  
Core(TM) i7-11370H @ 3.30GHz 3.30 GHz
2. Sistem operasi : Windows 11 Home Single Language
3. Bahasa pemrograman : PHP, Javascript, HTML, CSS
4. Framework : Laravel 11, Next.JS 14
5. Database Management System : MySQL
6. Alat bantu pemrograman : Visual Studio Code
7. Web server : Google Chrome

##### 5.1.2 Implementasi Database

Rancangan database untuk aplikasi SIPARUS diimplementasikan dengan menggunakan MySQL yang diakses melalui PhpMyAdmin. Gambar 5.2.1.1 hingga 5.1.2.5 menunjukkan hasil implementasi tabel *users*, *transactions*, *sessions*, *migrations*, dan *listings* secara berturut-turut.

###### 1. Tabel *users*

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	<b>id</b>	bigint(20)		UNSIGNED	Tidak	Tidak ada		AUTO_INCREMENT	Ubah  Hapus  Lainnya
2	<b>name</b>	varchar(255)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah  Hapus  Lainnya
3	<b>email</b>	varchar(255)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah  Hapus  Lainnya
4	<b>email_verified_at</b>	timestamp			Ya	NULL			Ubah  Hapus  Lainnya
5	<b>role</b>	enum('admin','user')	utf8mb4_general_ci		Tidak	user			Ubah  Hapus  Lainnya
6	<b>password</b>	varchar(255)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah  Hapus  Lainnya
7	<b>remember_token</b>	varchar(100)	utf8mb4_general_ci		Ya	NULL			Ubah  Hapus  Lainnya
8	<b>created_at</b>	timestamp			Ya	NULL			Ubah  Hapus  Lainnya
9	<b>updated_at</b>	timestamp			Ya	NULL			Ubah  Hapus  Lainnya

Gambar 5.1.2. 1 Tabel *users*

## 2. Tabel *transactions*

#	Nama	Jenis	Penyortiran	Atribut	Tak Ter nilai	Bawaan	Komentar	Ekstra	Tindakan
1	<code>id</code> 🛡	bigint(20)		UNSIGNED	Tidak	<i>Tidak ada</i>		AUTO_INCREMENT	📝 Ubah 🗑️ Hapus Lainnya
2	<code>user_id</code> 🚀	bigint(20)		UNSIGNED	Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya
3	<code>listing_id</code> 🚀	bigint(20)		UNSIGNED	Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya
4	<code>start_date</code>	date			Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya
5	<code>end_date</code>	date			Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya
6	<code>price</code>	int(10)			UNSIGNED	Tidak	0		📝 Ubah 🗑️ Hapus Lainnya
7	<code>total_bulan</code>	int(10)			UNSIGNED	Tidak	0		📝 Ubah 🗑️ Hapus Lainnya
8	<code>fee</code>	int(10)			UNSIGNED	Tidak	0		📝 Ubah 🗑️ Hapus Lainnya
9	<code>total_price</code>	int(10)			UNSIGNED	Tidak	0		📝 Ubah 🗑️ Hapus Lainnya
10	<code>status</code>	enum('waiting', 'approved', 'canceled')	utf8mb4_general_ci		Tidak	<i>waiting</i>			📝 Ubah 🗑️ Hapus Lainnya
11	<code>created_at</code>	timestamp			Ya	<i>NULL</i>			📝 Ubah 🗑️ Hapus Lainnya
12	<code>updated_at</code>	timestamp			Ya	<i>NULL</i>			📝 Ubah 🗑️ Hapus Lainnya

Gambar 5.1.2. 2 Tabel *transaction*

## 3. Tabel *sessions*

#	Nama	Jenis	Penyortiran	Atribut	Tak Ter nilai	Bawaan	Komentar	Ekstra	Tindakan
1	<code>id</code> 🛡	varchar(255)	utf8mb4_general_ci		Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya
2	<code>user_id</code> 🚶	bigint(20)		UNSIGNED	Ya	<i>NULL</i>			📝 Ubah 🗑️ Hapus Lainnya
3	<code>ip_address</code>	varchar(45)	utf8mb4_general_ci		Ya	<i>NULL</i>			📝 Ubah 🗑️ Hapus Lainnya
4	<code>user_agent</code>	text	utf8mb4_general_ci		Ya	<i>NULL</i>			📝 Ubah 🗑️ Hapus Lainnya
5	<code>payload</code>	longtext	utf8mb4_general_ci		Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya
6	<code>last_activity</code> 🚶	int(11)			Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya

Gambar 5.1.2. 3 Tabel *sessions*

## 4. Tabel *migrations*

#	Nama	Jenis	Penyortiran	Atribut	Tak Ter nilai	Bawaan	Komentar	Ekstra	Tindakan
1	<code>id</code> 🛡	int(10)		UNSIGNED	Tidak	<i>Tidak ada</i>		AUTO_INCREMENT	📝 Ubah 🗑️ Hapus Lainnya
2	<code>migration</code>	varchar(255)	utf8mb4_general_ci		Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya
3	<code>batch</code>	int(11)			Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya

Gambar 5.1.2. 4 Tabel *migrations*

## 5. Tabel *listings*

#	Nama	Jenis	Penyortiran	Atribut	Tak Ter nilai	Bawaan	Komentar	Ekstra	Tindakan
1	<code>id</code> 🛡	bigint(20)		UNSIGNED	Tidak	<i>Tidak ada</i>		AUTO_INCREMENT	📝 Ubah 🗑️ Hapus Lainnya
2	<code>title</code> 🚶	varchar(255)	utf8mb4_general_ci		Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya
3	<code>slug</code> 🚶	varchar(255)	utf8mb4_general_ci		Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya
4	<code>description</code>	text	utf8mb4_general_ci		Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya
5	<code>address</code>	varchar(255)	utf8mb4_general_ci		Tidak	<i>Tidak ada</i>			📝 Ubah 🗑️ Hapus Lainnya
6	<code>luas_kamar_m2</code>	smallint(5)		UNSIGNED	Tidak	0			📝 Ubah 🗑️ Hapus Lainnya
7	<code>kapasitas</code>	smallint(5)		UNSIGNED	Tidak	0			📝 Ubah 🗑️ Hapus Lainnya
8	<code>price</code>	int(10)			UNSIGNED	Tidak	0		📝 Ubah 🗑️ Hapus Lainnya
9	<code>attachments</code>	longtext	utf8mb4_general_ci		Ya	<i>NULL</i>			📝 Ubah 🗑️ Hapus Lainnya
10	<code>full_support_available</code>	smallint(5)		UNSIGNED	Tidak	0			📝 Ubah 🗑️ Hapus Lainnya
11	<code>mini_canteen_available</code>	smallint(5)		UNSIGNED	Tidak	0			📝 Ubah 🗑️ Hapus Lainnya
12	<code>created_at</code>	timestamp			Ya	<i>NULL</i>			📝 Ubah 🗑️ Hapus Lainnya
13	<code>updated_at</code>	timestamp			Ya	<i>NULL</i>			📝 Ubah 🗑️ Hapus Lainnya
14	<code>deleted_at</code>	timestamp			Ya	<i>NULL</i>			📝 Ubah 🗑️ Hapus Lainnya

Gambar 5.1.2. 5 Tabel *listings*

### 5.1.3 Implementasi Antarmuka

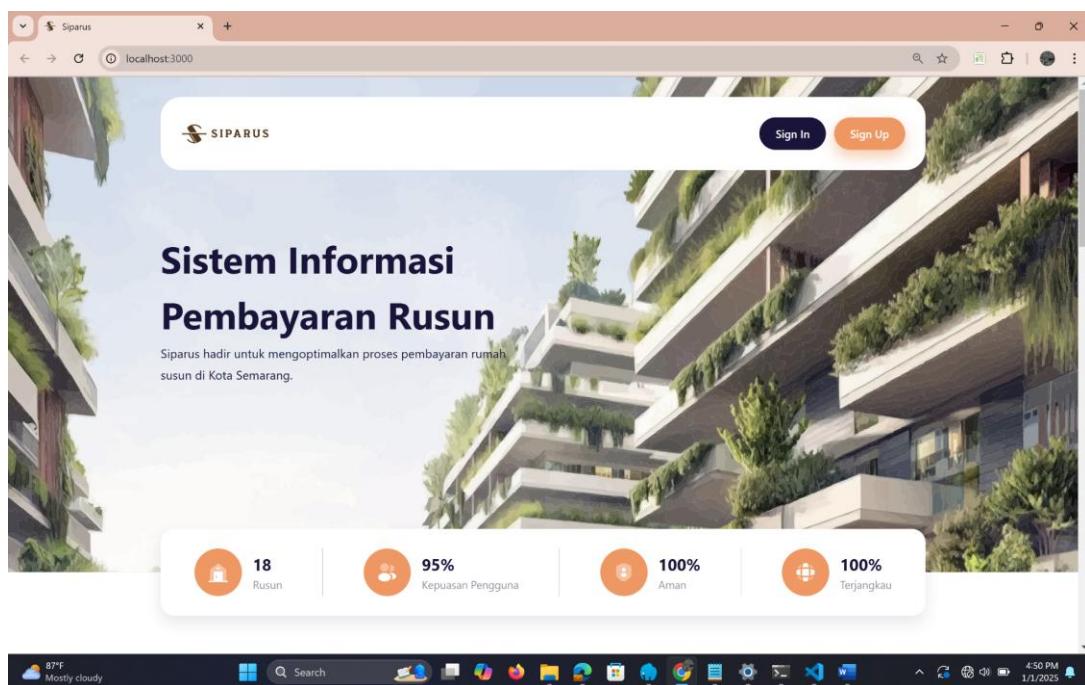
Aplikasi SIPARUS dirancang dalam bentuk diagram sequence dan diimplementasikan menggunakan framework Laravel dan Next.js, serta didukung oleh HTML, CSS, dan JavaScript. Berikut ini adalah hasil implementasi untuk setiap halaman.

### 5.1.3.1 Antarmuka User

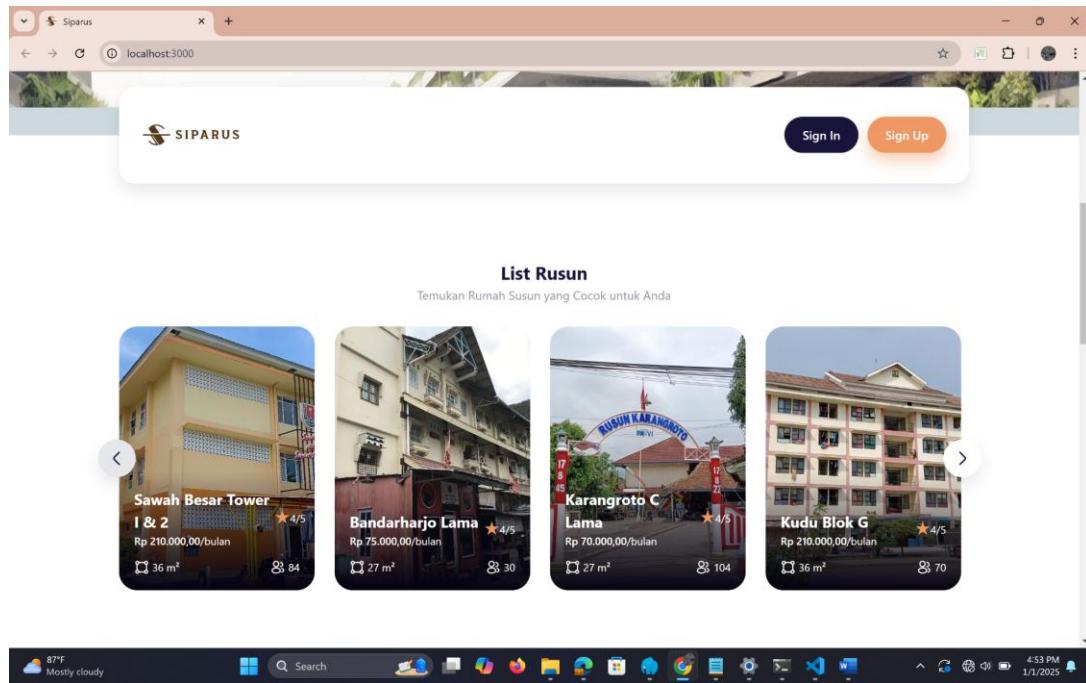
#### 1. Halaman *Landing Page*

Gambar 5.1.3.1.1.1 hingga 5.1.3.1.1.4 menjelaskan *Landing Page* pada user. Halaman *Landing Page* SIPARUS menonjolkan fitur utama berupa List Rusun, yang dirancang untuk mempermudah pengguna menemukan unit rumah susun yang sesuai dengan kebutuhan mereka. Setiap kartu rusun menampilkan informasi penting seperti harga bulanan, luas unit, dan jumlah penghuni. Pengguna juga dapat melihat rating untuk membantu memilih unit terbaik.

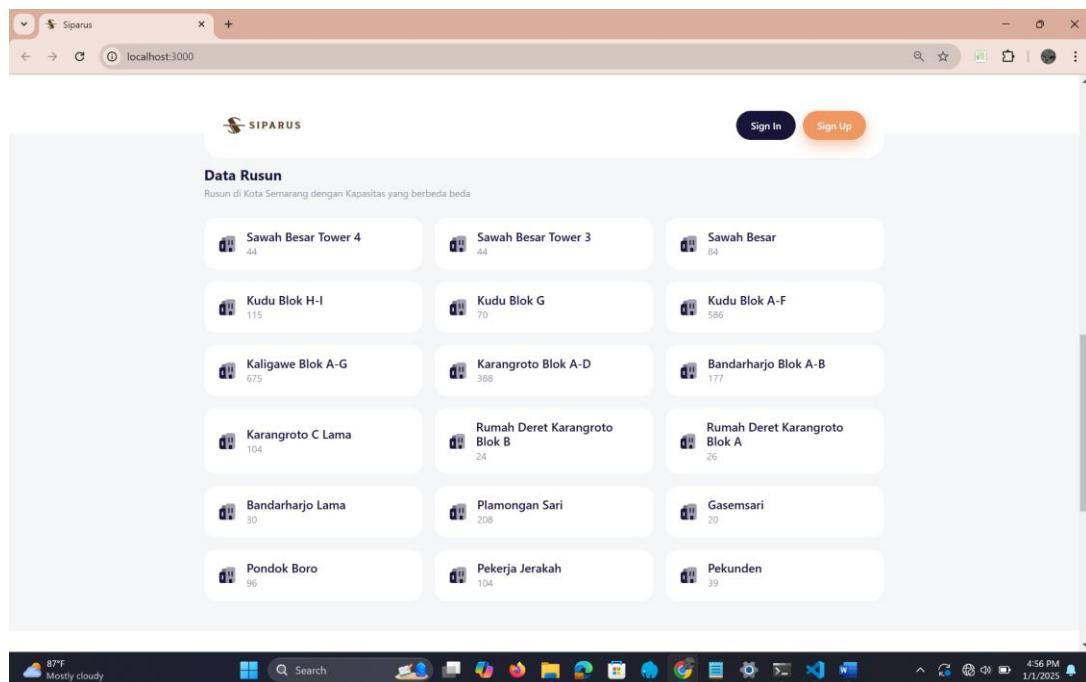
Di bagian atas halaman, terdapat header dengan logo SIPARUS dan tombol navigasi Sign In serta Sign Up. Judul utama "Sistem Informasi Pembayaran Rusun" memperkenalkan tujuan platform, didukung oleh statistik jumlah rusun, kapasitas, keamanan, dan keterjangkauan.



Gambar 5.1.3.1.1. 1 Halaman Landing Page 1



Gambar 5.1.3.1.1. 2 Halaman *Landing Page 2*



Gambar 5.1.3.1.1. 3 Halaman *Landing Page 3*



#### Fasilitas Rumah Susun yang Menjamin Kenyamanan dan Keamanan

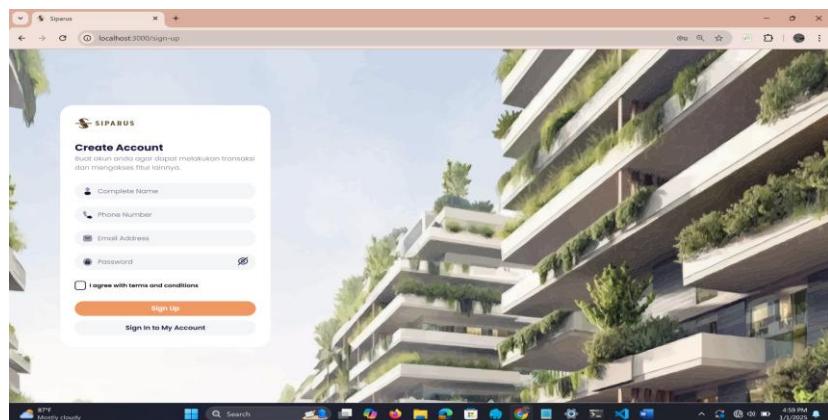
- ✓ Keamanan 24 jam dengan sistem pengawasan CCTV
- ✓ Fasilitas parkir yang luas dan teratur
- ✓ Akses transportasi umum yang mudah dijangkau
- ✓ Kamar yang dirancang dengan efisien dan fungsional
- ✓ Lingkungan yang bersih dan nyaman untuk keluarga



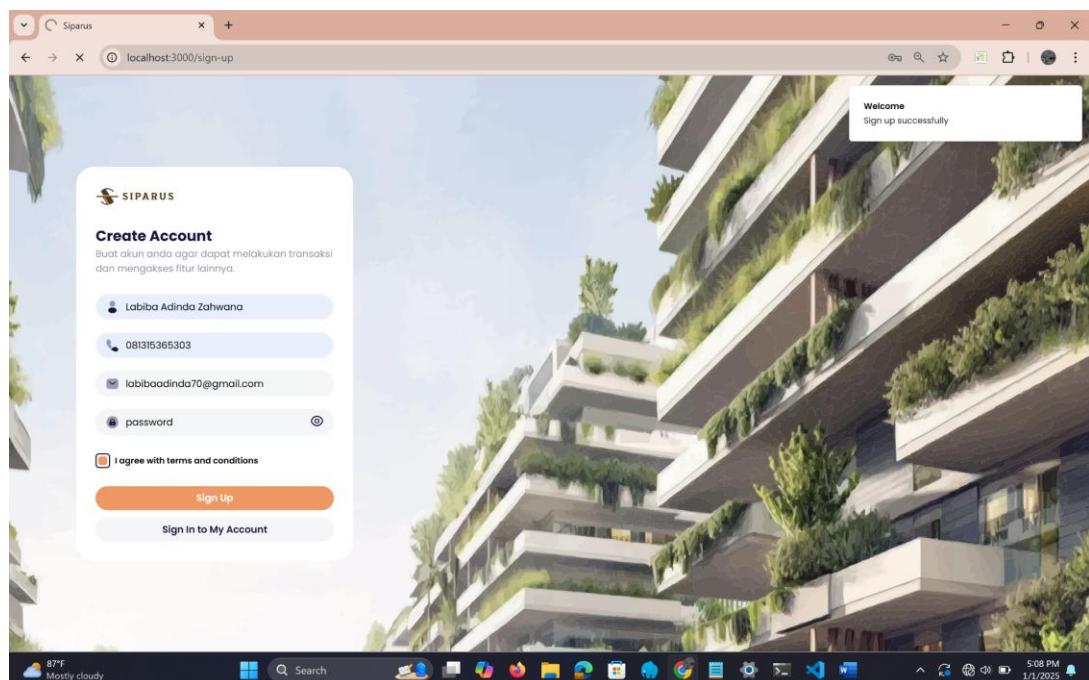
Gambar 5.1.3.1.1. 4 Halaman *Landing Page* 4

## 2. Halaman *Sign Up*

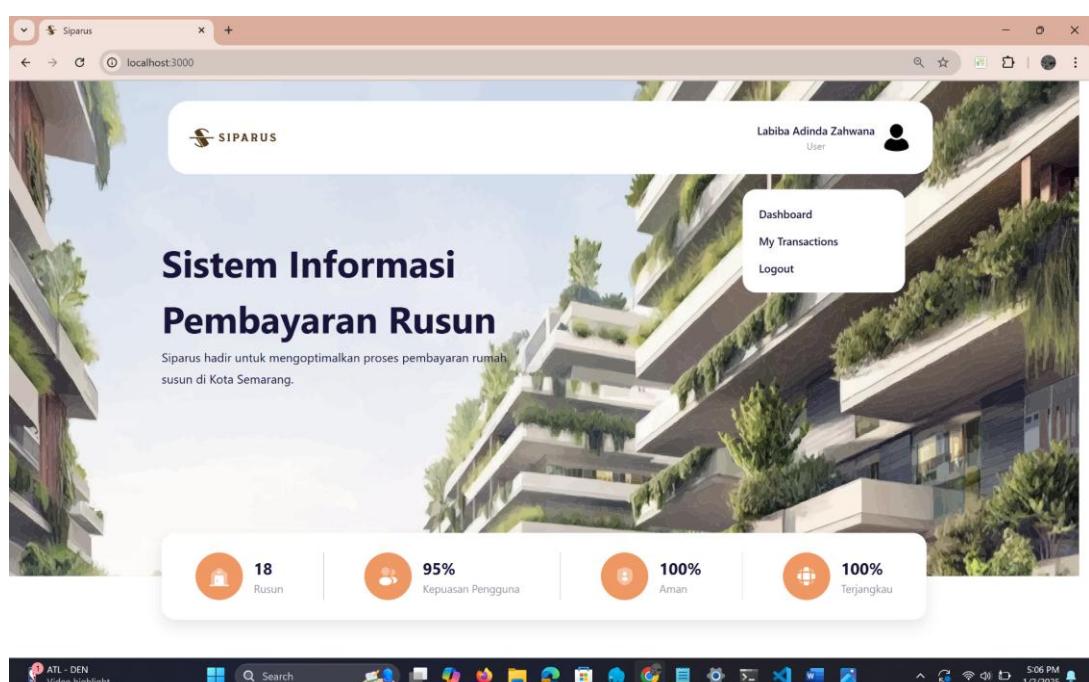
Gambar 5.1.3.1.2.1 hingga 5.1.3.1.2.4 menjelaskan *Sign Up* pada User. Halaman *Sign Up* pada aplikasi SIPARUS memiliki *form input* untuk membuat akun yaitu *Complete Name, Phone Number, Email Address, dan Password*. Pengguna juga diwajibkan mencentang checkbox *I agree with terms and conditions* sebelum menekan tombol *Sign Up* untuk mendaftarkan akun. Selain itu, terdapat tombol *Sign In to My Account* yang memungkinkan pengguna untuk beralih ke Landing Page dalam kondisi sudah terhubung dengan akun yang dibuat.



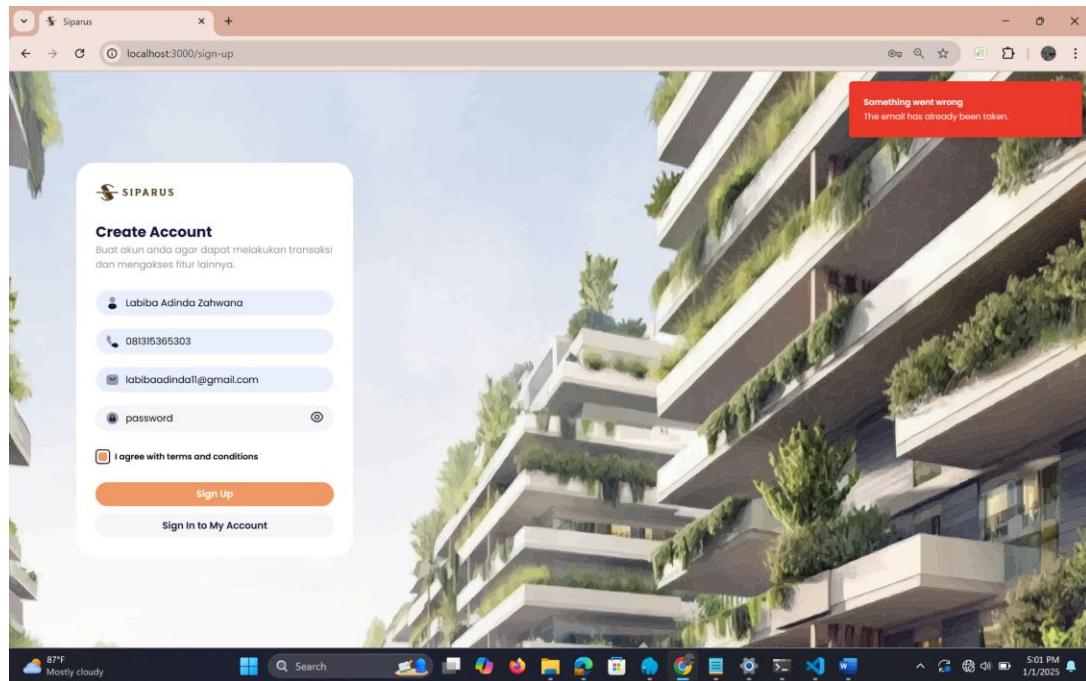
Gambar 5.1.3.1.2. 1 Halaman *Sign Up*



Gambar 5.1.3.1.2. 2 Halaman *Sign Up* berhasil



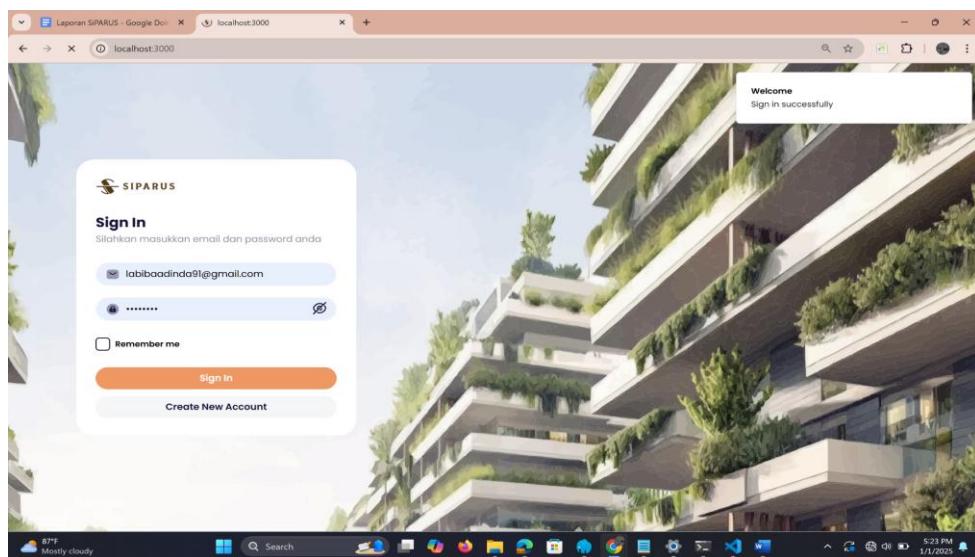
Gambar 5.1.3.1.2. 3 Halaman *auto direct* dan *auto masuk akun* setelah berhasil *Sign Up*



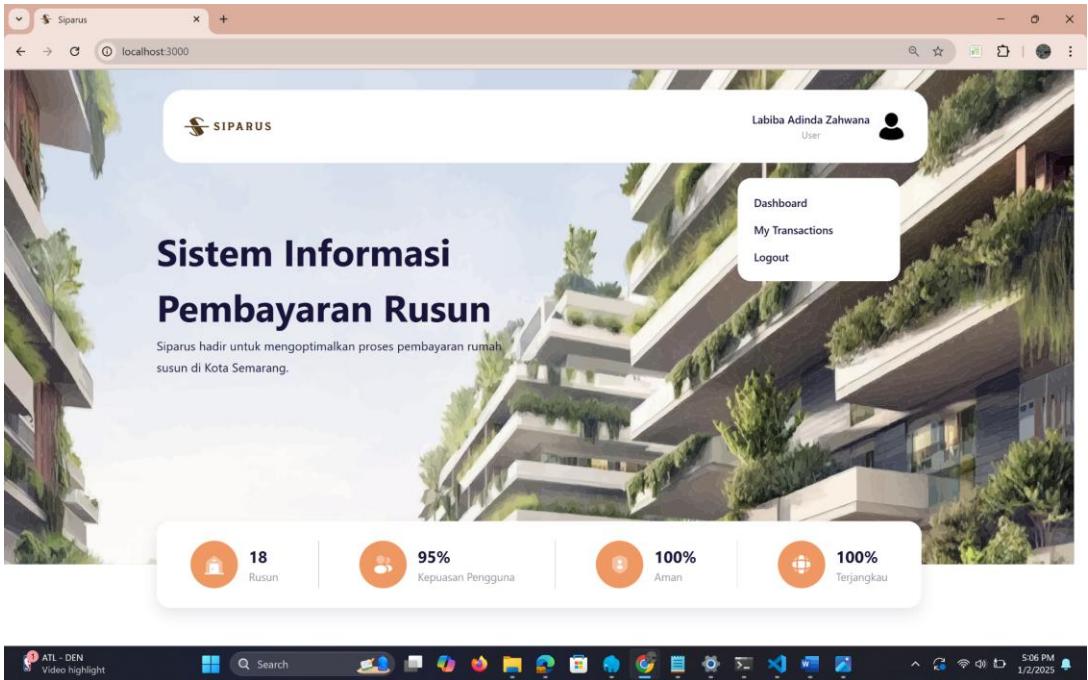
Gambar 5.1.3.1.2. 4 Halaman Notifikasi Gagal Buat Akun menggunakan Email yang sama

### 3. Halaman *Sign In*

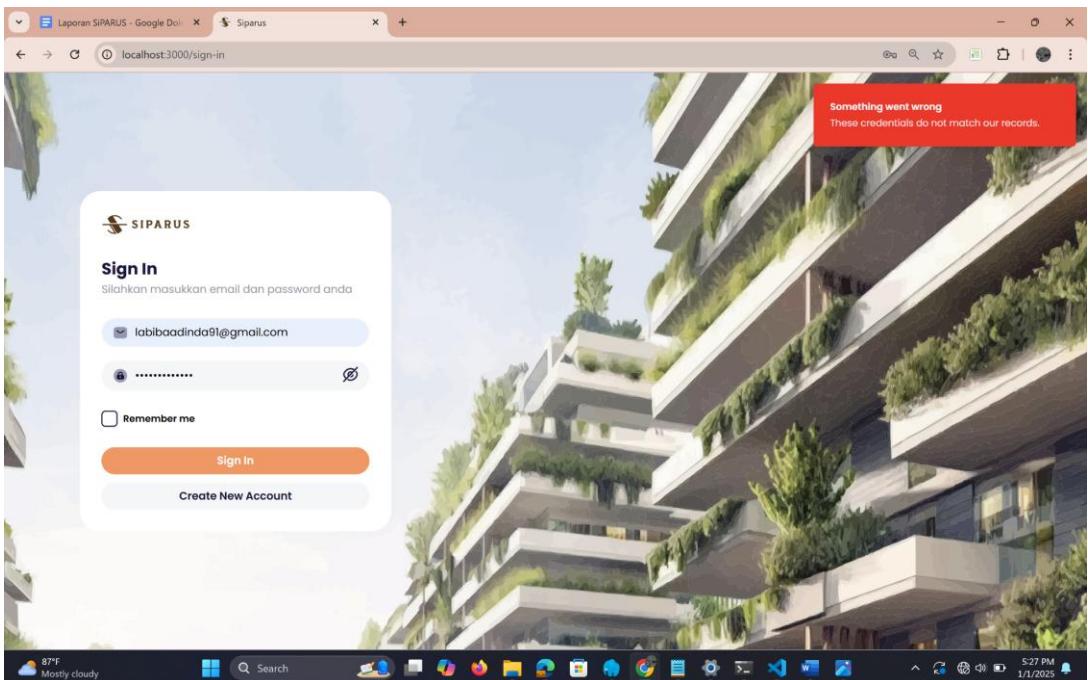
Gambar 5.1.3.1.3.1 hingga 5.1.3.1.3.3 menjelaskan *Sign In* pada User. Halaman *Sign In* pada aplikasi SIPARUS dirancang untuk memudahkan pengguna dalam mengakses akun mereka. Pada halaman ini, pengguna dapat memasukkan email dan kata sandi untuk masuk. Terdapat juga opsi "*Remember Me*" yang memungkinkan pengguna tetap masuk tanpa harus mengisi ulang kredensial setiap kali mengakses aplikasi.



Gambar 5.1.3.1.3. 1 Halaman Sign In berhasil



Gambar 5.1.3.1.3. 2 Halaman *Signed In*

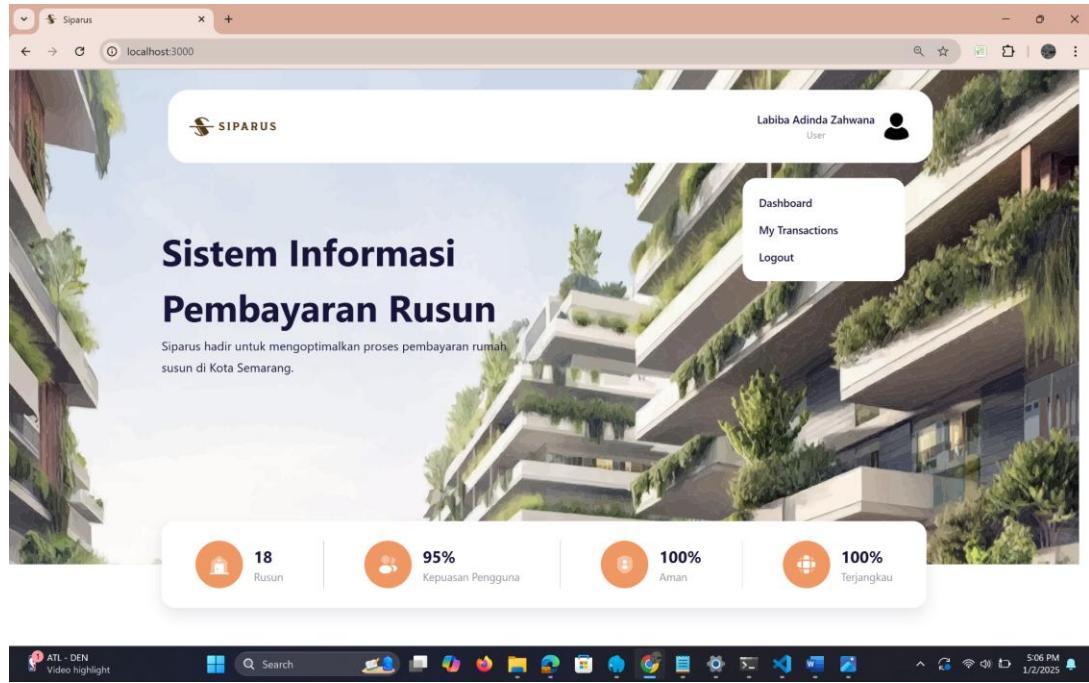


Gambar 5.1.3.1.3. 3 Halaman *Sign In* gagal

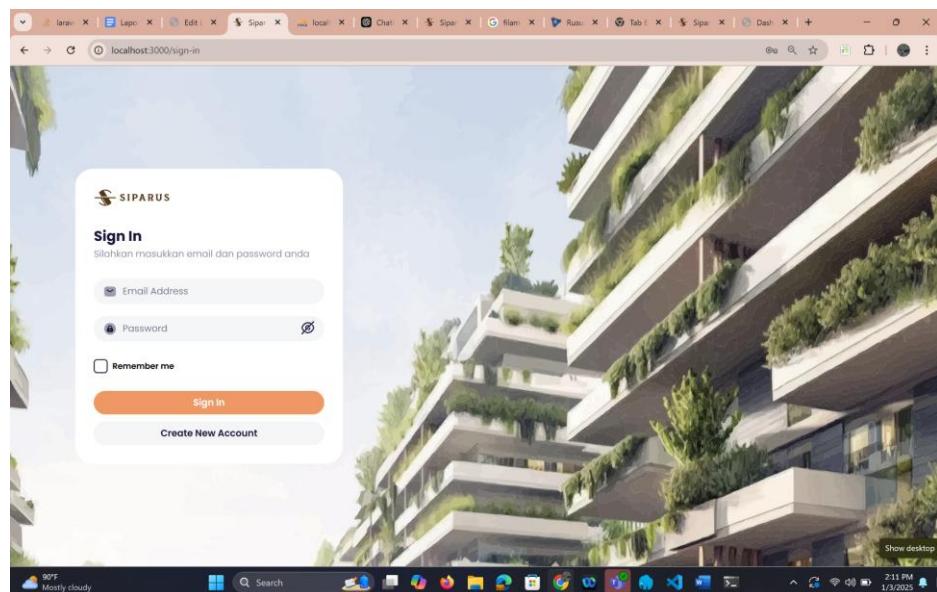
#### 4. Halaman *Logout*

Gambar 5.1.3.1.4.1 hingga 5.1.3.1.4.2 menjelaskan *Logout* pada User. Pada tampilan halaman SIPARUS yang menampilkan pengguna telah masuk (logged in), terdapat menu dropdown yang terletak di bagian kanan atas *navbar*.

Menu ini menyediakan tiga opsi yaitu *Dashboard*, *My Transactions*, dan *Logout* untuk memudahkan navigasi pengguna. Opsi *Logout* yang memungkinkan pengguna untuk keluar dari akun mereka. Fitur logout ini dirancang untuk memberikan keamanan tambahan dengan memastikan sesi pengguna dapat diakhiri dengan mudah.



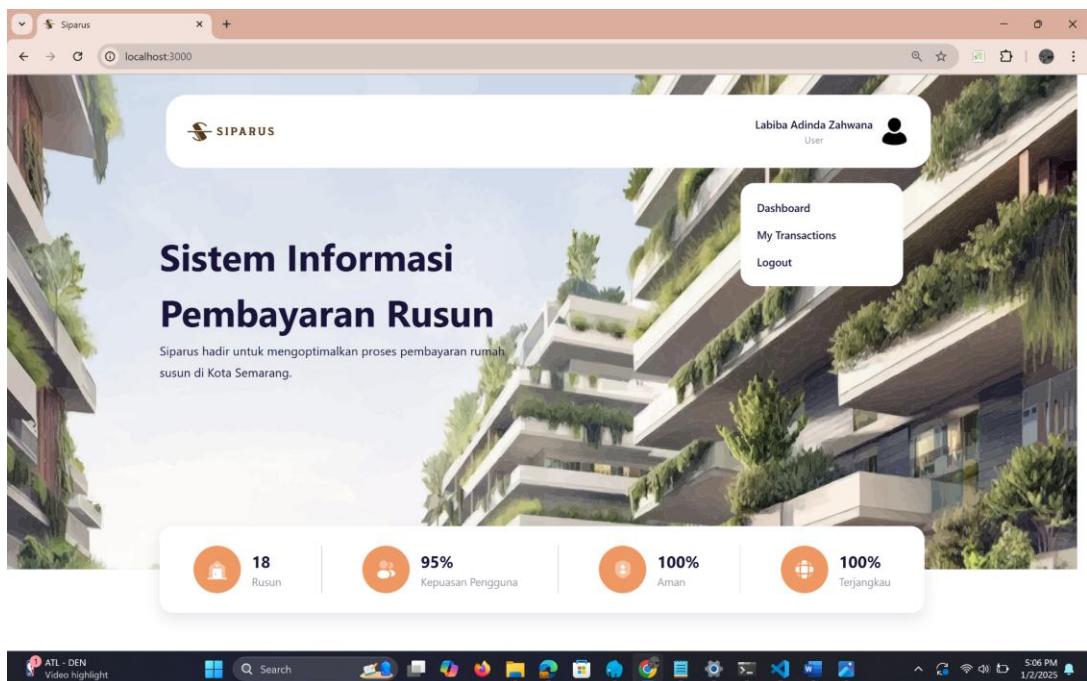
Gambar 5.1.3.1.4. 1 Halaman Navbar untuk *Logout*



Gambar 5.1.3.1.4. 2 Halaman direct ke *Login Page*

## 5. Halaman *Navbar Logged In*

Gambar 5.1.3.1.5.1 menjelaskan *Navbar Logged In* pada User. Pada tampilan halaman SIPARUS yang menampilkan pengguna telah masuk (logged in), terdapat menu dropdown yang terletak di bagian kanan atas *navbar*. Menu ini menyediakan tiga opsi utama untuk memudahkan navigasi pengguna. Opsi pertama adalah *Dashboard*, yang mengarahkan pengguna ke halaman ringkasan user. Opsi kedua adalah *My Transactions*, yang memungkinkan pengguna untuk melihat riwayat transaksi pembayaran atau detail pembayaran yang telah dilakukan. Terakhir, terdapat opsi *Logout* yang memungkinkan pengguna untuk keluar dari akun mereka. Fitur logout ini dirancang untuk memberikan keamanan tambahan dengan memastikan sesi pengguna dapat diakhiri dengan mudah, sehingga pengguna dapat meninggalkan aplikasi dengan tenang tanpa khawatir terhadap keamanan data mereka.

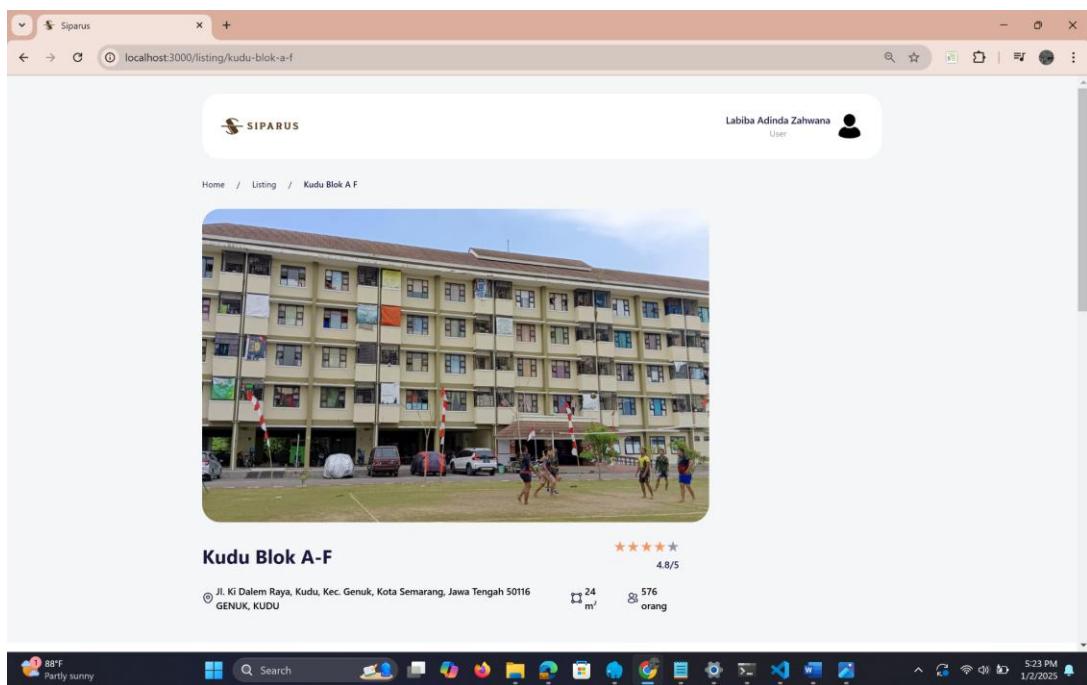


Gambar 5.1.3.1.5. 1 Halaman *Logged In*

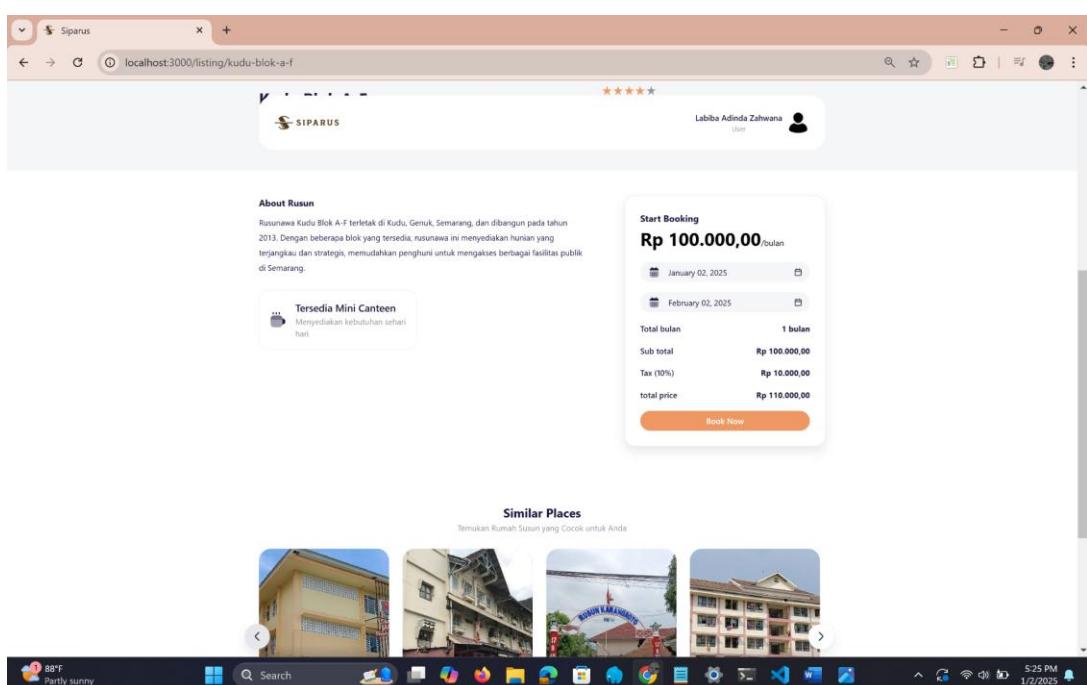
## 6. Halaman Detail Rusun

Gambar 5.1.3.1.6.1 hingga 5.1.3.1.6.3 menjelaskan halaman Detail Rusun pada User. Halaman detail rusun pada aplikasi SIPARUS menyediakan informasi lengkap mengenai unit rumah susun yang dipilih oleh pengguna. Pada bagian atas, breadcrumb ditampilkan untuk membantu pengguna mengetahui posisi mereka dalam aplikasi dan navigasi ke halaman sebelumnya. Halaman ini memuat informasi utama seperti lokasi, luas area, kapasitas penghuni, dan rating. Di bagian deskripsi, pengguna dapat melihat detail tambahan mengenai fasilitas yang disediakan oleh rusun.

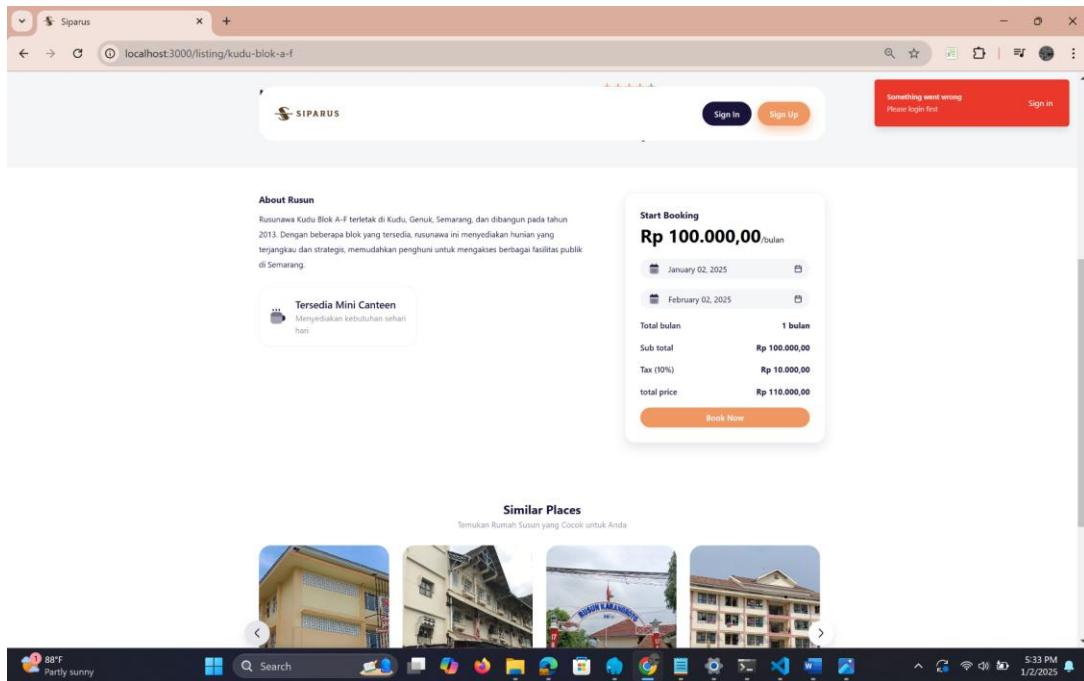
Panel di sisi kanan halaman dirancang untuk memudahkan pengguna dalam memesan unit. Panel ini mencakup informasi harga, durasi sewa, serta total biaya yang perlu dibayarkan. Tombol "*Book Now*" tersedia untuk mengarahkan pengguna ke proses pemesanan. Bagian bawah halaman menampilkan rekomendasi unit rumah susun lainnya yang sesuai dengan preferensi pengguna.



Gambar 5.1.3.1.6. 1 Detail Rusun



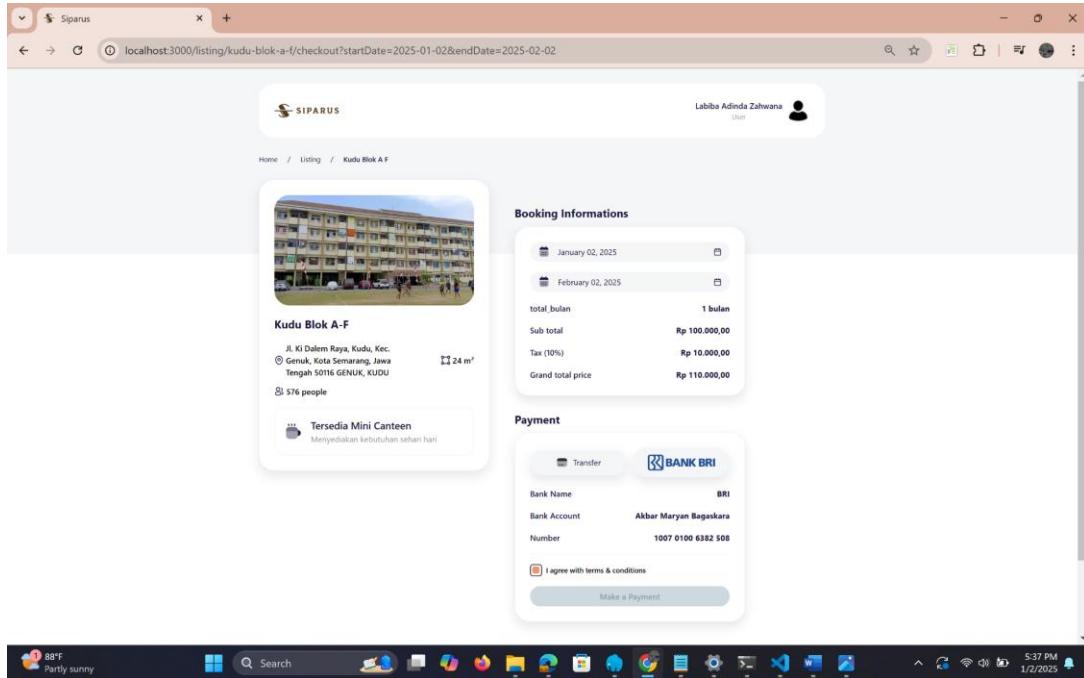
Gambar 5.1.3.1.6. 2 Detail Rusun set booking



Gambar 5.1.3.1.6. 3 Halaman booking gagal karena belum sign in

## 7. Halaman *Checkout* Pembayaran

Gambar 5.1.3.1.7.1 menjelaskan halaman *Checkout* pada user. Halaman *Checkout* pada aplikasi SIPARUS dirancang untuk memfasilitasi proses pembayaran pengguna setelah mereka memilih unit rumah susun. Pada bagian kanan, terdapat *Booking Information* yang merinci periode sewa, subtotal, pajak, dan total harga akhir yang harus dibayar. Selain itu, terdapat bagian *Payment* yang memberikan detail metode pembayaran, seperti nama bank, nomor rekening, dan nama pemilik akun bank. Pengguna juga diwajibkan untuk menyetujui syarat dan ketentuan sebelum dapat menyelesaikan pembayaran melalui tombol *Make a Payment*.

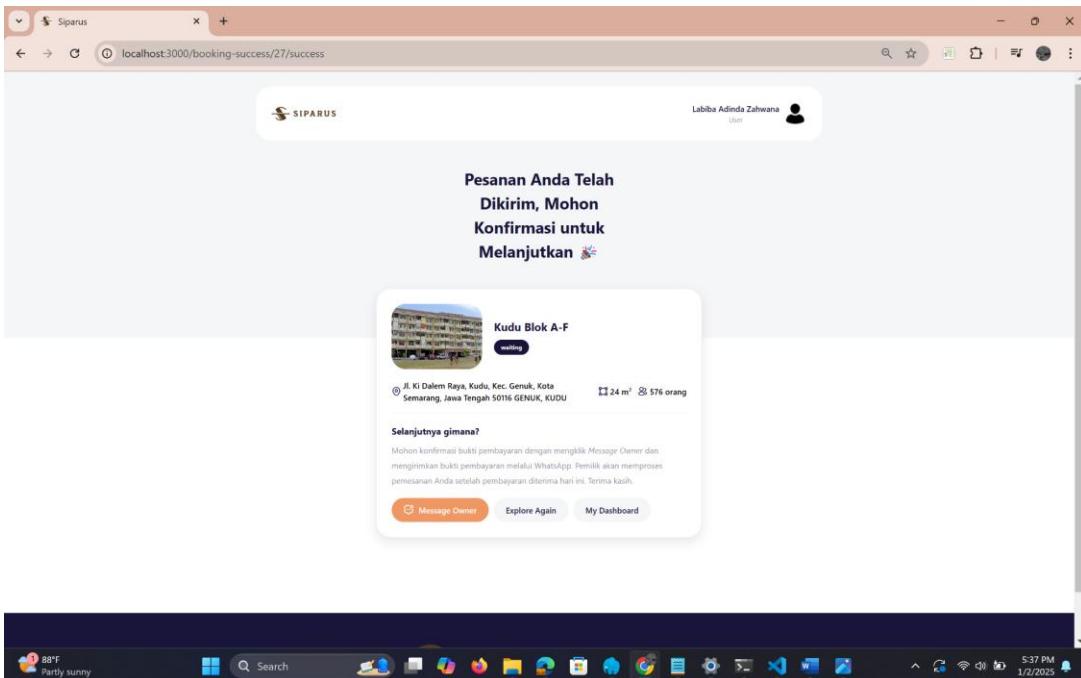


Gambar 5.1.3.1.7. 1 Halaman *Checkout*

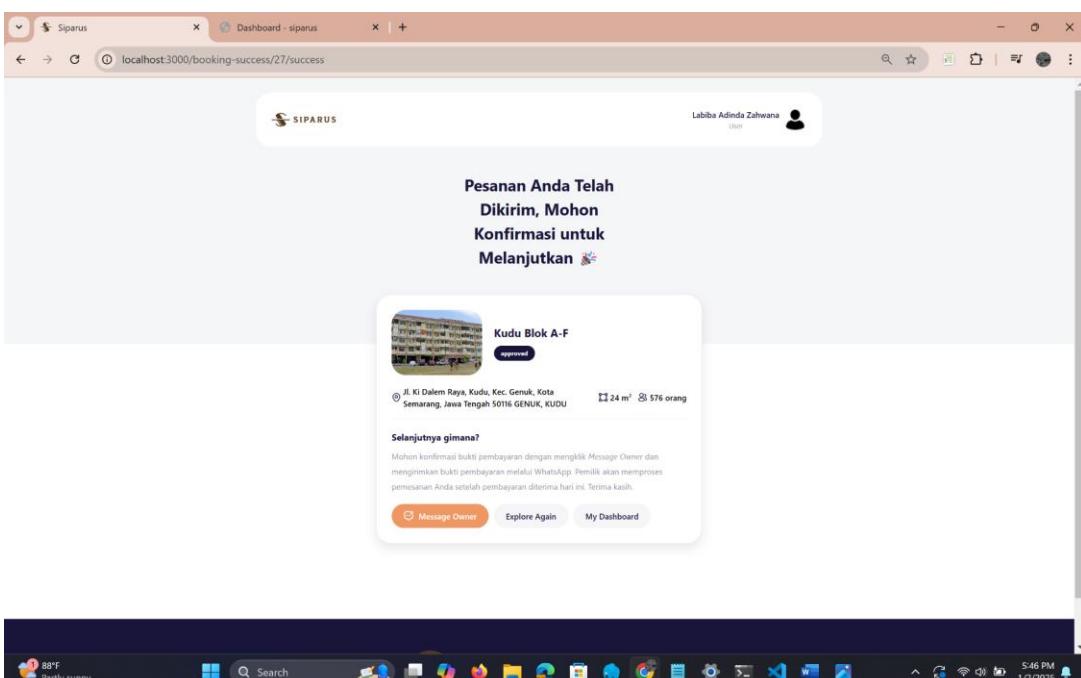
## 8. Halaman Booking Sukses

Gambar 5.1.3.1.8.1 hingga 5.1.3.1.8.2 menjelaskan halaman Booking sukses pada user. Halaman Booking Sukses menampilkan pesan konfirmasi bahwa pemesanan unit rumah susun telah berhasil dilakukan. Pada tahap ini, status transaksi awal berada dalam kondisi "waiting" yang berarti pemesanan masih menunggu konfirmasi lebih lanjut dari pihak pemilik atau pengelola rumah susun.

Pengguna diberikan instruksi untuk mengirimkan bukti pembayaran melalui fitur *Message Owner*. Proses ini dilakukan untuk memastikan bahwa pembayaran telah diterima, sehingga pemesanan dapat diproses lebih lanjut. Selain itu, halaman ini menyediakan opsi untuk kembali menjelajahi unit lain dengan tombol *Explore Again* atau menuju ke *Dashboard* untuk melihat riwayat transaksi.



Gambar 5.1.3.1.8. 1 Halaman Booking Sukses dan menunggu konfirmasi dari admin

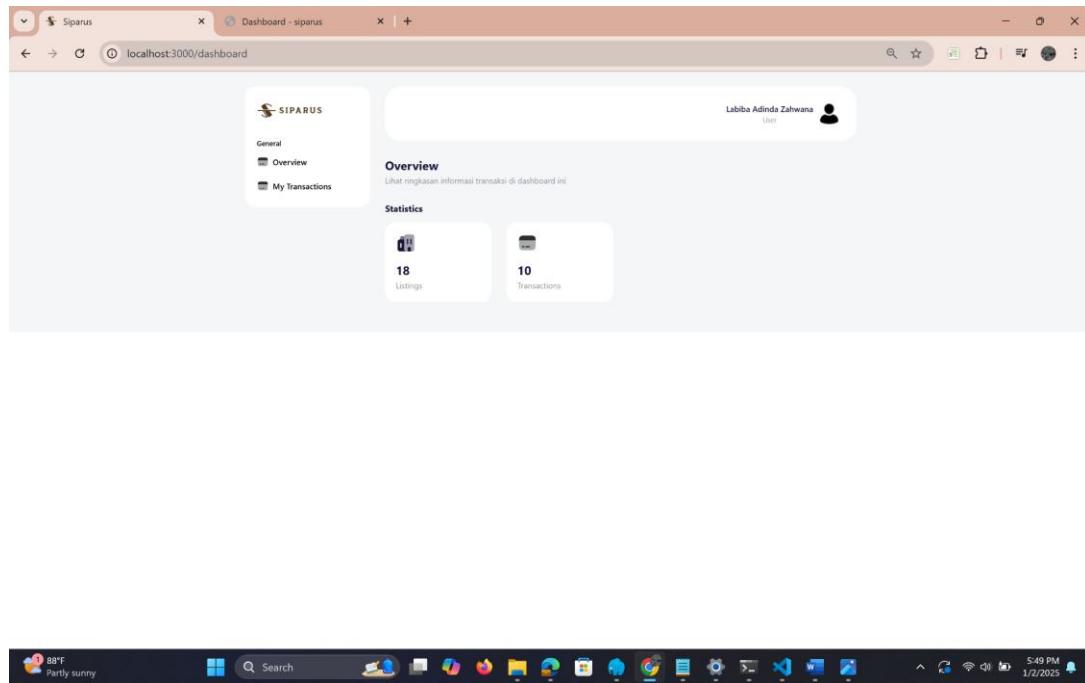


Gambar 5.1.3.1.8. 2 Halaman booking sudah disetujui (approved)

## 9. Halaman *Dashboard Overview*

Gambar 5.1.3.1.9.1 menjelaskan halaman *Dashboard Overview* pada user.

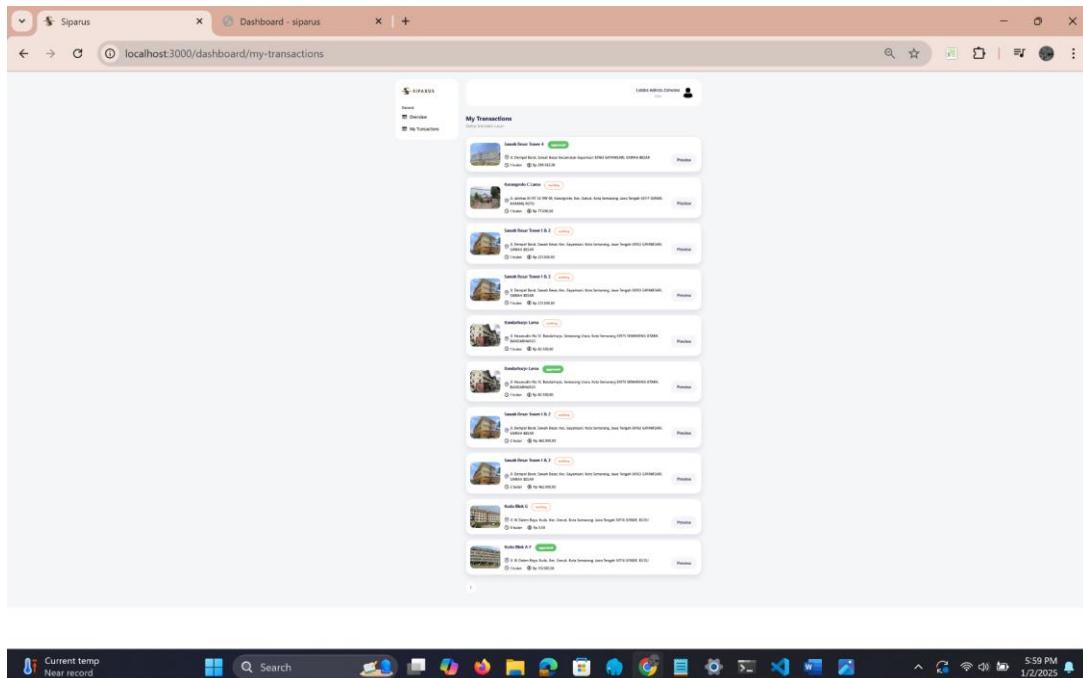
Halaman *Dashboard Overview* dirancang untuk memberikan gambaran ringkas terkait aktivitas pengguna di platform SIPARUS. Pada bagian ini, pengguna dapat melihat statistik utama, seperti jumlah listing rumah susun yang tersedia dan total transaksi yang telah dilakukan.



Gambar 5.1.3.1.9. 1 Halaman *Overview*

## 10. Halaman *Dashboard Transactions*

Gambar 5.1.3.1.10.1 menjelaskan halaman *Dashboard Transactions* pada user. Halaman *My Transactions* pada dashboard SIPARUS memberikan daftar transaksi rumah susun yang dimiliki pengguna. Setiap transaksi menampilkan informasi penting seperti daftar transaksi yang pernah dilakukan dan status transaksi tersebut.

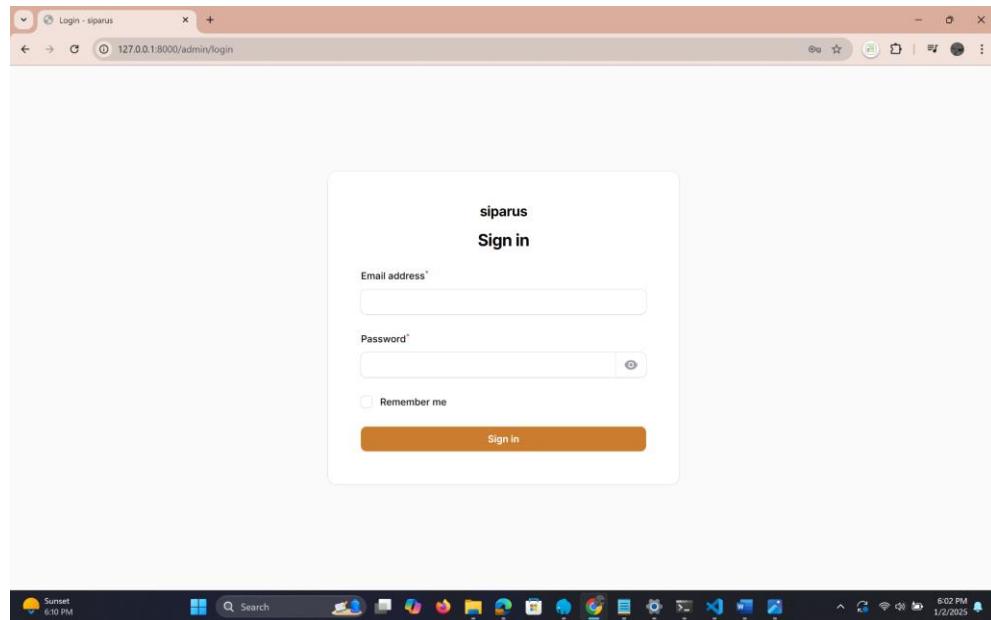


Gambar 5.1.3.1.10. 1 Halaman *dashboard* transaksi yang telah dilakukan

### 5.1.3.2 Antarmuka Admin

#### 1. Halaman *Sign In*

Gambar 5.1.3.2.1.1 menjelaskan halaman *Sign In* pada admin. Halaman *Sign In* dirancang untuk memudahkan admin dalam mengakses akun mereka. Pada halaman ini dapat memasukkan email dan kata sandi untuk masuk. Terdapat juga opsi "*Remember Me*" yang memungkinkan pengguna tetap masuk tanpa harus mengisi ulang kredensial setiap kali mengakses aplikasi.

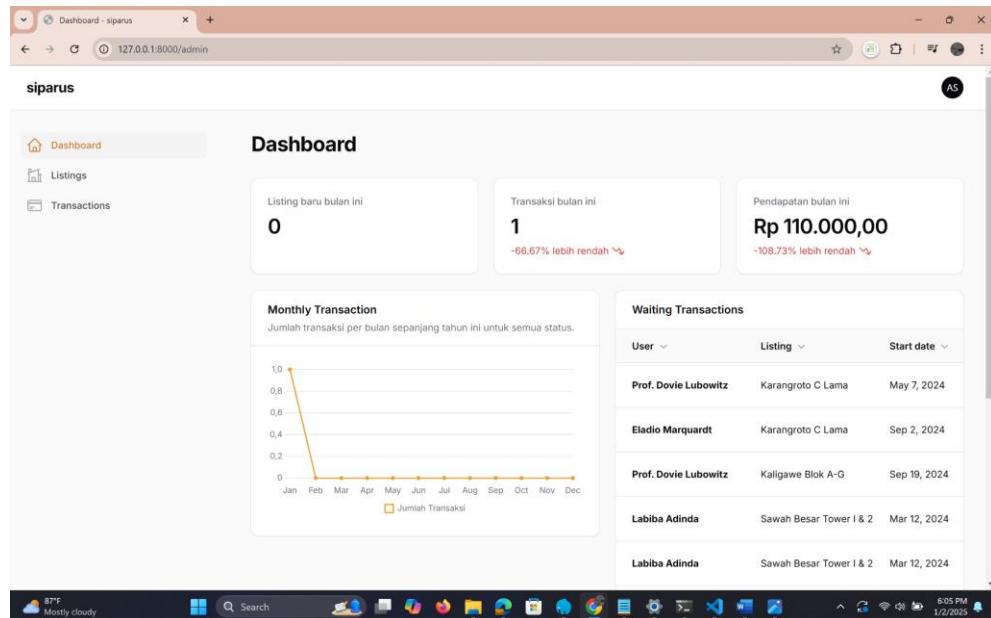


Gambar 5.1.3.2.1. 1 Halaman *Sign In* admin

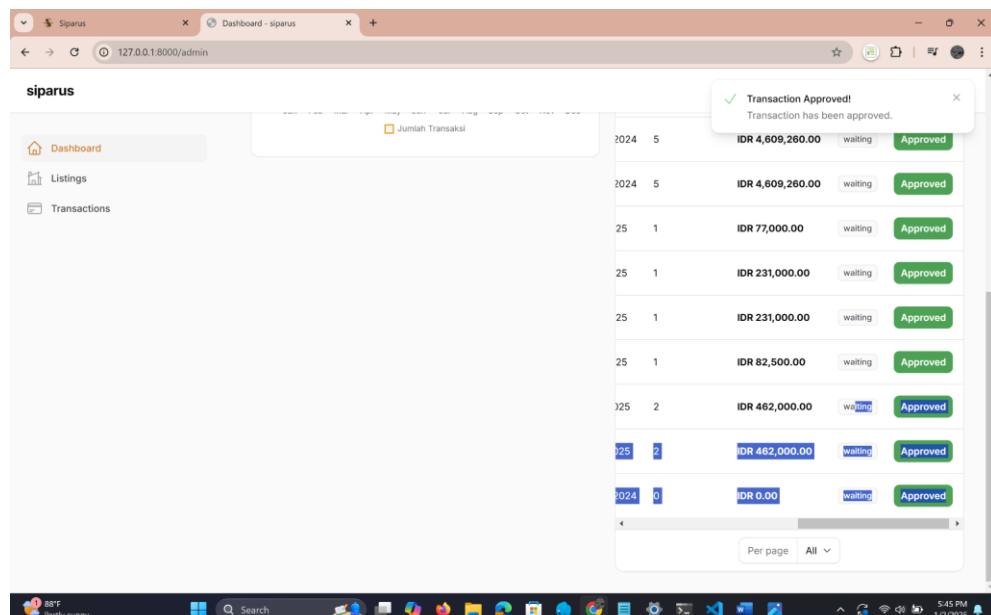
## 2. Halaman *Landing Page*

Gambar 5.1.3.2.2.1 hingga 5.1.3.2.2.2 menjelaskan halaman *Landing Page* pada admin. Halaman *Landing Page Admin* pada sistem SIPARUS memberikan gambaran umum mengenai aktivitas dan performa pengelolaan rumah susun. Di bagian atas, terdapat informasi statistik penting seperti jumlah listing baru, total transaksi bulan ini, dan pendapatan bulan ini, disertai dengan perubahan persentase dibandingkan bulan sebelumnya.

Terdapat grafik yang menunjukkan jumlah transaksi bulanan sepanjang tahun, memberikan visualisasi tren performa. Di sisi kanan, daftar transaksi yang masih menunggu proses (*waiting transactions*). Halaman ini membantu admin untuk memantau aktivitas sistem secara efisien dan mengambil tindakan berdasarkan data yang disajikan.



Gambar 5.1.3.2.2. 1 Halaman *Landing Page* admin

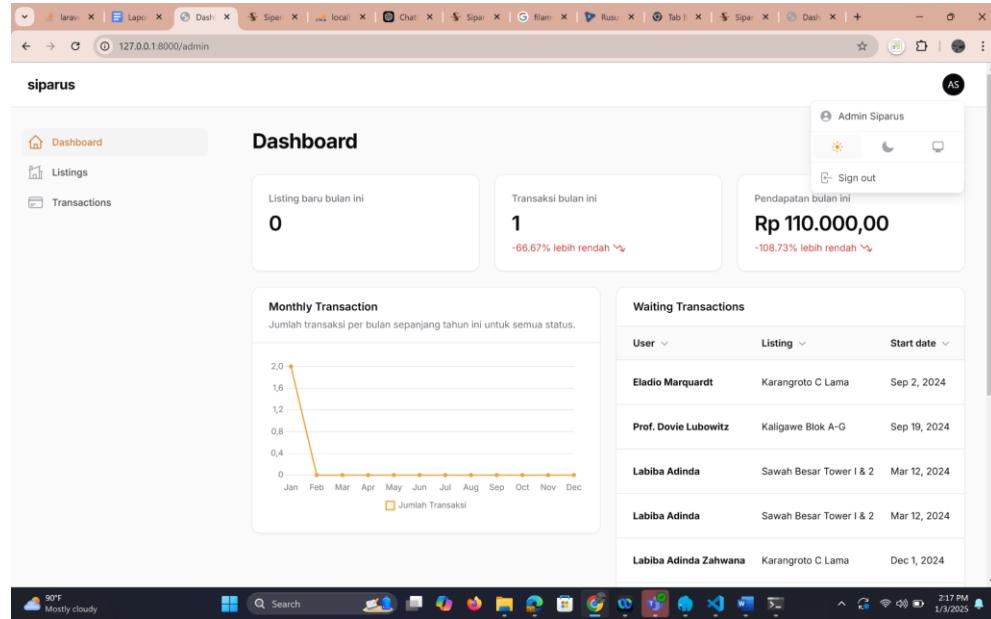


Gambar 5.1.3.2.2. 2 Halaman *Landing Page* berhasil *approved*

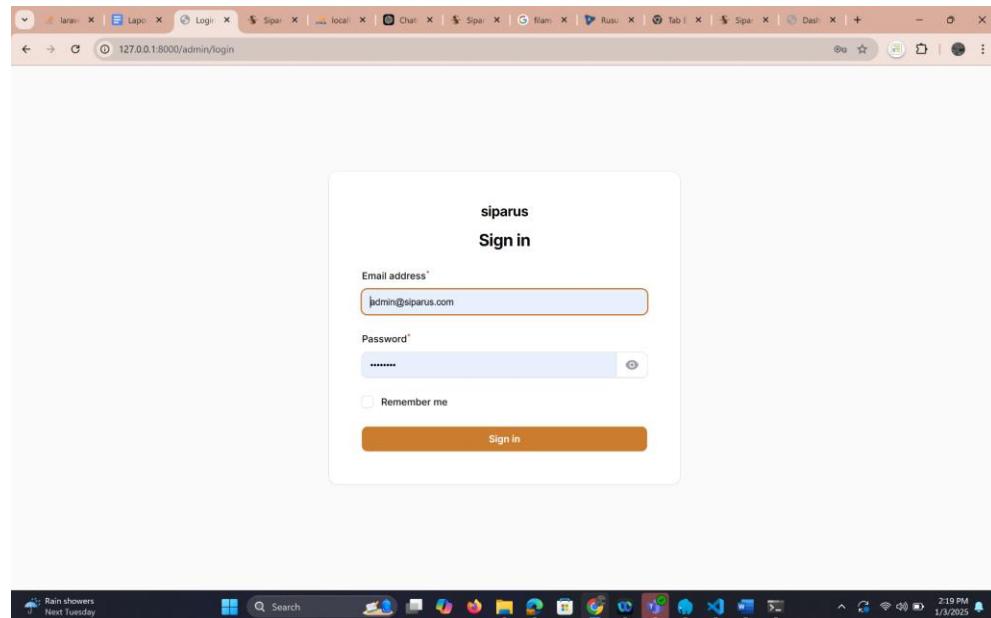
### 3. Halaman *Navbar Logout*

Gambar 5.1.3.2.3.1 hingga 5.1.3.2.3.2 menjelaskan *Navbar Logout* pada admin. *Navbar* ini mencakup opsi *Theme Settings*, yang memungkinkan admin untuk menyesuaikan tampilan aplikasi dengan tema terang atau gelap, memberikan fleksibilitas dan kenyamanan visual sesuai preferensi pengguna.

Selain itu, terdapat opsi *Sign Out* yang digunakan untuk keluar dari akun secara aman, memastikan bahwa sesi pengguna ditutup dengan benar.



Gambar 5.1.3.2.3. 1 Halaman *Navbar Logout*



Gambar 5.1.3.2.3. 2 Halaman *Redirect Sign In page*

#### 4. Halaman *Listings*

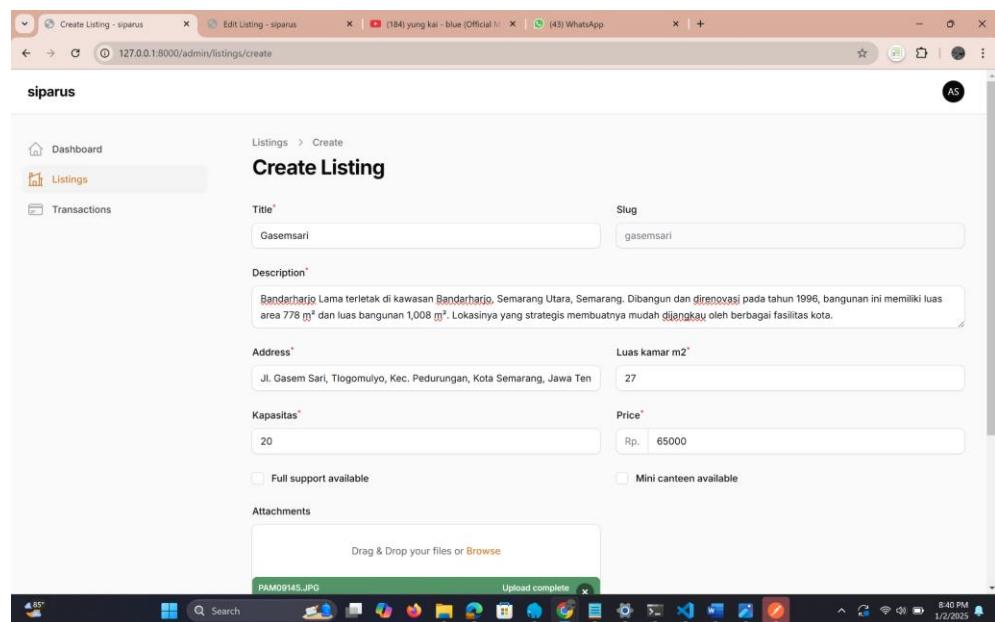
Gambar 5.1.3.2.4.1 hingga 5.1.3.2.4.8 menjelaskan halaman *Listings* pada admin. Halaman *Listings* pada panel admin SIPARUS dirancang untuk

mengelola daftar rumah susun yang tersedia dalam sistem. Admin dapat melihat informasi penting seperti judul rusun, luas kamar, kapasitas, harga, dan ketersediaan fasilitas mini canteen. Informasi ini disusun dalam format tabel yang terorganisir sehingga memudahkan pengelolaan.

Pada halaman ini, admin memiliki dua fitur utama yaitu *Edit* dan *Delete* untuk setiap listing. Tombol *Edit* memungkinkan admin memperbarui informasi rumah susun, seperti harga, kapasitas, atau fasilitas yang tersedia. Sementara itu, tombol *Delete* digunakan untuk menghapus listing rusun yang sudah tidak relevan atau tersedia. Selain itu, terdapat tombol *New Listing* yang memungkinkan admin menambahkan listing baru dengan informasi lengkap terkait rumah susun, memberikan fleksibilitas dalam mengelola data yang terus diperbarui.

Title	Luas kamar m2	Kapasitas	Price	Mini canteen available	
Sawah Besar Tower 4	36	44	IDR 250,000.00	1	
Sawah Besar Tower 3	36	44	IDR 250,000.00	1	
Sawah Besar Tower I & 2	36	84	IDR 210,000.00	1	
Kudu Blok H-I	36	115	IDR 200,000.00	1	
Kudu Blok G	36	70	IDR 210,000.00	1	
Kudu Blok A-F	24	576	IDR 100,000.00	1	
Kaligawe Blok A-G	21	675	IDR 100,000.00	0	
Karangroto Blok A-D	24	384	IDR 90,000.00	0	

Gambar 5.1.3.2.4. 1 Halaman *Listings*



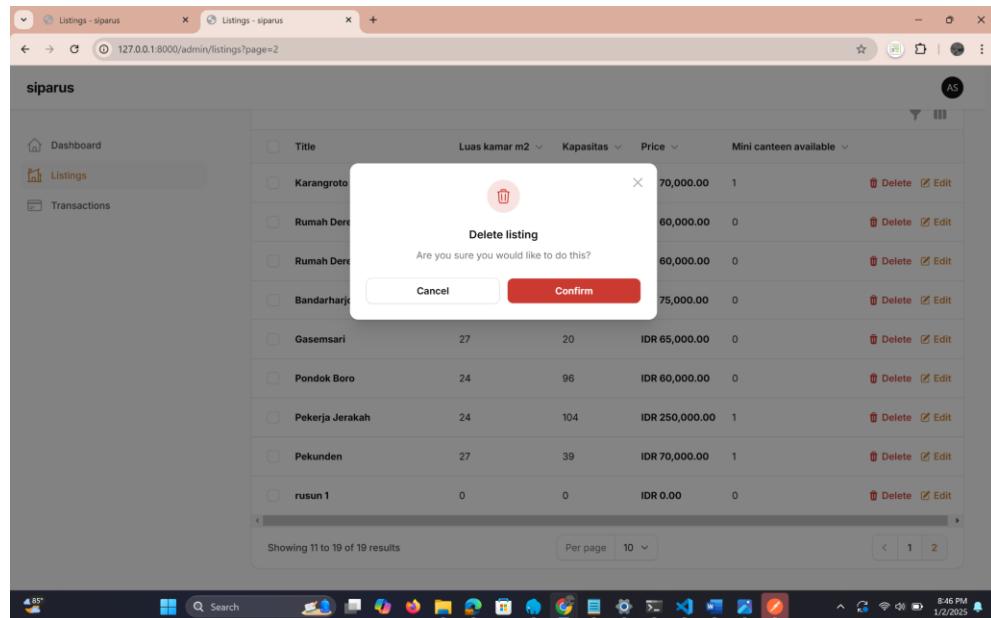
Gambar 5.1.3.2.4. 2 Halaman tambah *listings* rusun



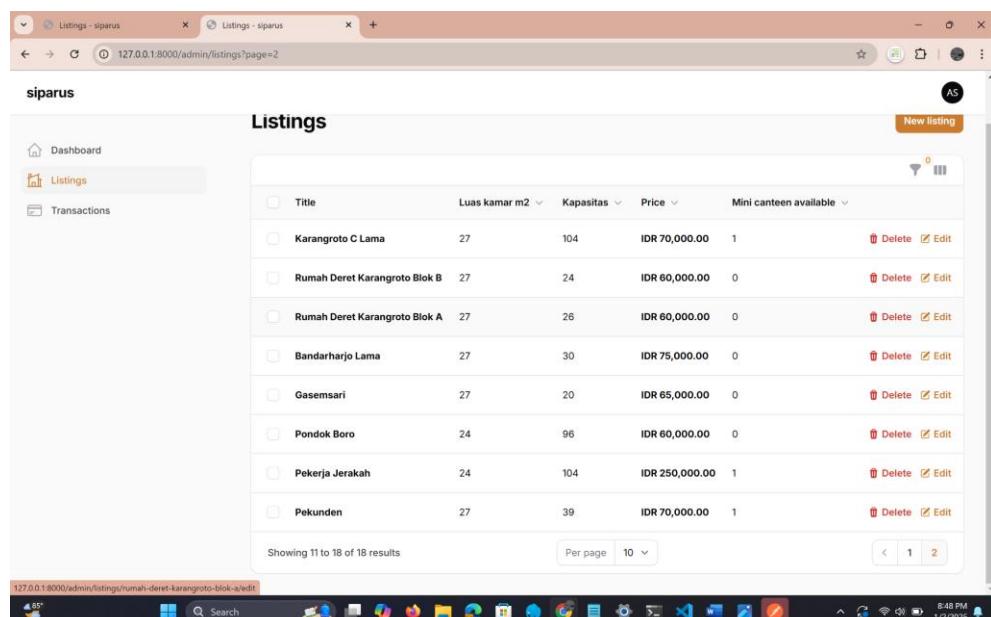
Gambar 5.1.3.2.4. 3 Halaman *redirect* ke *edit listing page* dengan *pop up Created*

Listings					
<input type="checkbox"/> Title	Luas kamar m2	Kapasitas	Price	Mini canteen available	
<input type="checkbox"/> Karangroto C Lama	27	104	IDR 70,000.00	1	<input type="button" value="Delete"/> <input type="checkbox"/> Edit
<input type="checkbox"/> Rumah Deret Karangroto Blok B	27	24	IDR 60,000.00	0	<input type="button" value="Delete"/> <input type="checkbox"/> Edit
<input type="checkbox"/> Rumah Deret Karangroto Blok A	27	26	IDR 60,000.00	0	<input type="button" value="Delete"/> <input type="checkbox"/> Edit
<input type="checkbox"/> Bandarharjo Lama	27	30	IDR 75,000.00	0	<input type="button" value="Delete"/> <input type="checkbox"/> Edit
<input type="checkbox"/> Gasemsari	27	20	IDR 65,000.00	0	<input type="button" value="Delete"/> <input type="checkbox"/> Edit
<input type="checkbox"/> Pondok Boro	24	96	IDR 60,000.00	0	<input type="button" value="Delete"/> <input type="checkbox"/> Edit
<input type="checkbox"/> Pekerja Jerakah	24	104	IDR 250,000.00	1	<input type="button" value="Delete"/> <input type="checkbox"/> Edit
<input type="checkbox"/> Pekunden	27	39	IDR 70,000.00	1	<input type="button" value="Delete"/> <input type="checkbox"/> Edit

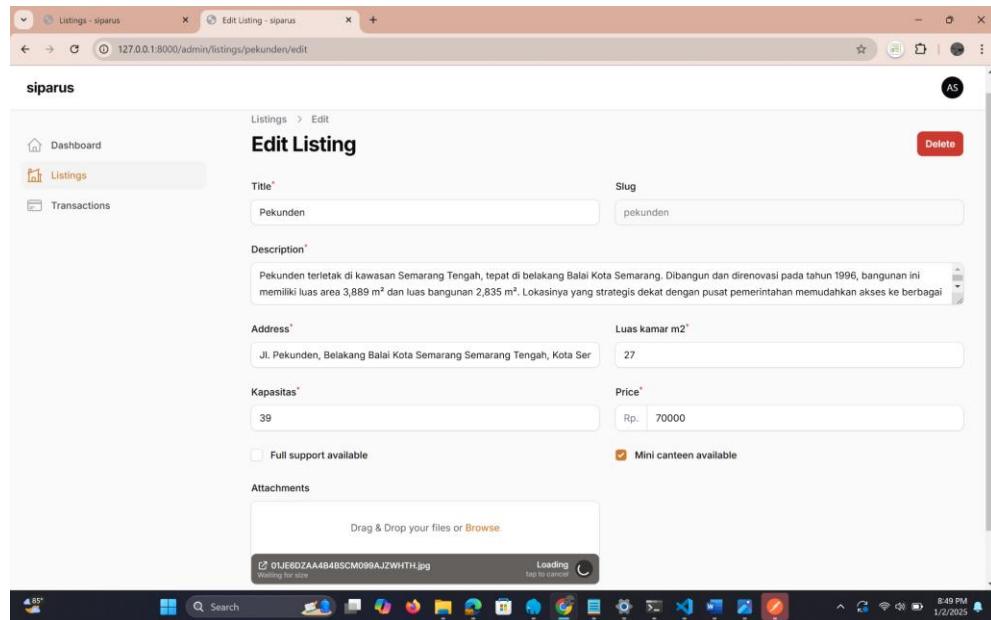
Gambar 5.1.3.2.4. 4 Halaman daftar *listings* dengan *listings* yang sudah ditambahkan



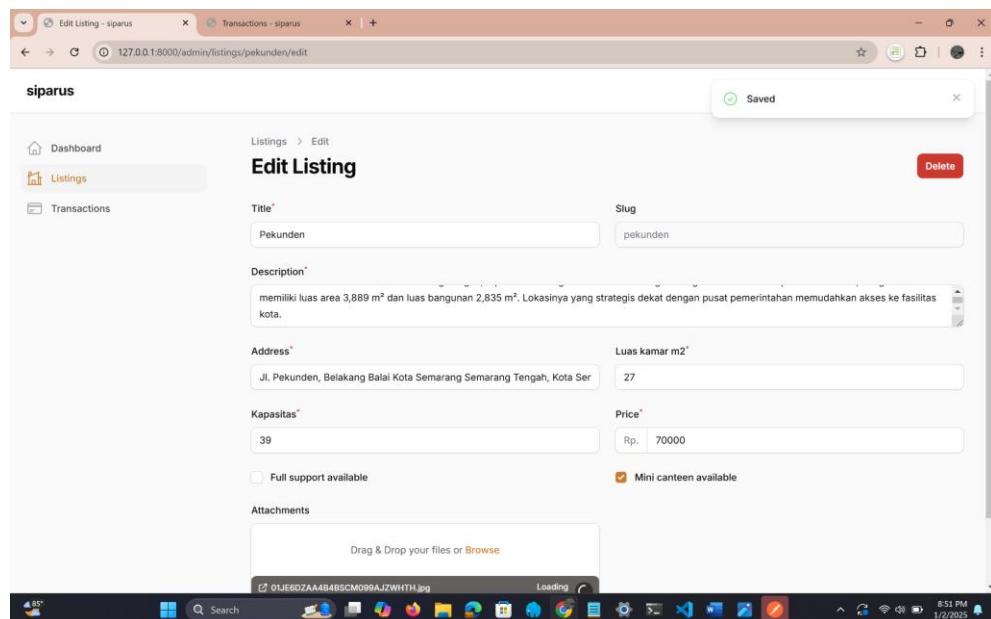
Gambar 5.1.3.2.4. 5 Halaman *listings delete listing*



Gambar 5.1.3.2.4. 6 Halaman *listings berhasil deleted*



Gambar 5.1.3.2.4. 7 Halaman *edit listing*



Gambar 5.1.3.2.4. 8 Halaman berhasil *edit listing*

## 5. Halaman Transaksi

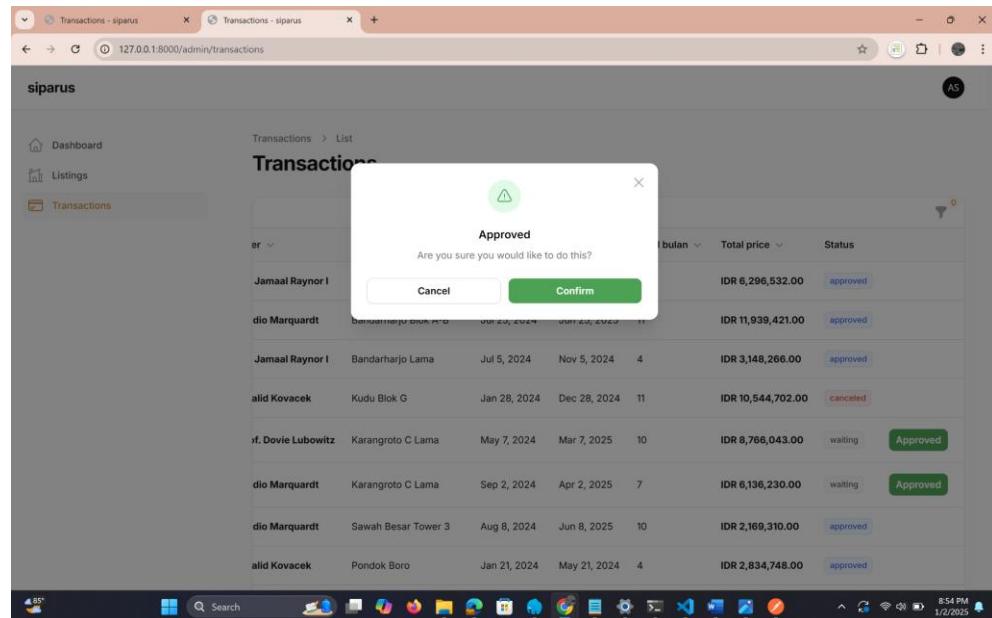
Gambar 5.1.3.2.5.1 hingga 5.1.3.2.5.5 menjelaskan halaman Transaksi pada admin. Halaman *Transactions* pada panel admin SIPARUS dirancang untuk memberikan pengelolaan yang efisien terhadap data transaksi yang terjadi dalam sistem. Halaman ini menampilkan daftar lengkap transaksi dengan informasi penting, seperti nama pengguna, daftar unit yang disewa, tanggal

mulai dan selesai penyewaan, jumlah bulan penyewaan, total harga, dan status transaksi.

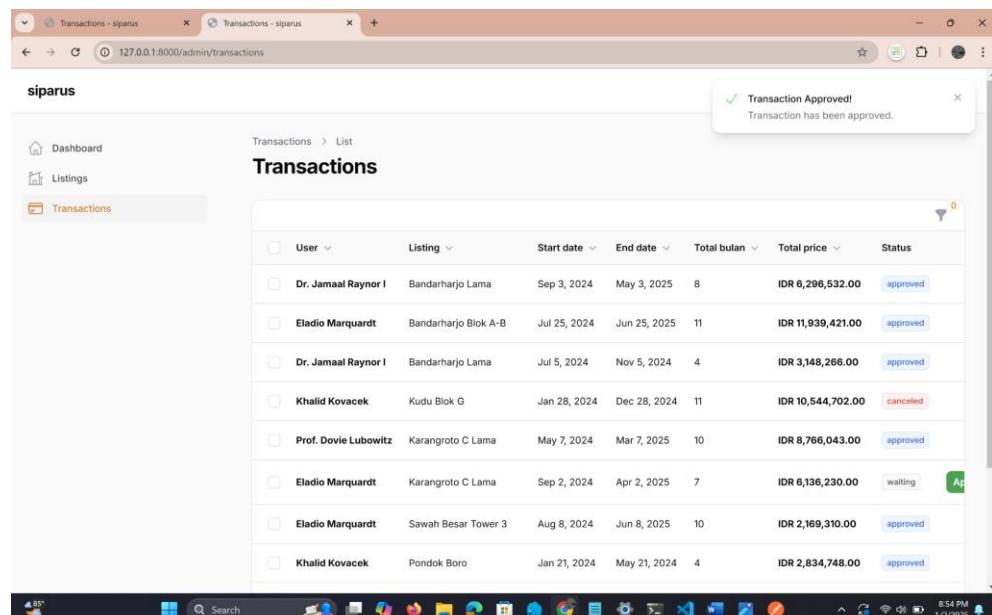
Admin dapat melihat status setiap transaksi, yang dikelompokkan ke dalam kategori seperti "*approved*," "*waiting*," atau "*canceled*." Fitur ini memudahkan admin untuk memantau dan mengelola proses transaksi. Selain itu, notifikasi pop-up seperti "*Transaction Approved!*" muncul sebagai konfirmasi tindakan yang berhasil dilakukan oleh admin, seperti menyetujui transaksi. Desain yang terorganisir memungkinkan admin untuk fokus pada pengambilan keputusan yang cepat dan akurat dalam pengelolaan transaksi penyewaan.

User	Listing	Start date	End date	Total bulan	Total price	Status
Dr. Jamaal Raynor I	Bandarharjo Lama	Sep 3, 2024	May 3, 2025	8	IDR 6,296,532.00	approved
Eladio Marquardt	Bandarharjo Blok A-B	Jul 25, 2024	Jun 25, 2025	11	IDR 11,939,421.00	approved
Dr. Jamaal Raynor I	Bandarharjo Lama	Jul 5, 2024	Nov 5, 2024	4	IDR 3,148,266.00	approved
Khalid Kovacek	Kudu Blok G	Jan 28, 2024	Dec 28, 2024	11	IDR 10,544,702.00	canceled
Prof. Dovie Lubowitz	Karangroto C Lama	May 7, 2024	Mar 7, 2025	10	IDR 8,766,043.00	waiting
Eladio Marquardt	Karangroto C Lama	Sep 2, 2024	Apr 2, 2025	7	IDR 6,136,230.00	waiting
Eladio Marquardt	Sawah Besar Tower 3	Aug 8, 2024	Jun 8, 2025	10	IDR 2,169,310.00	approved
Khalid Kovacek	Pondok Boro	Jan 21, 2024	May 21, 2024	4	IDR 2,834,748.00	approved

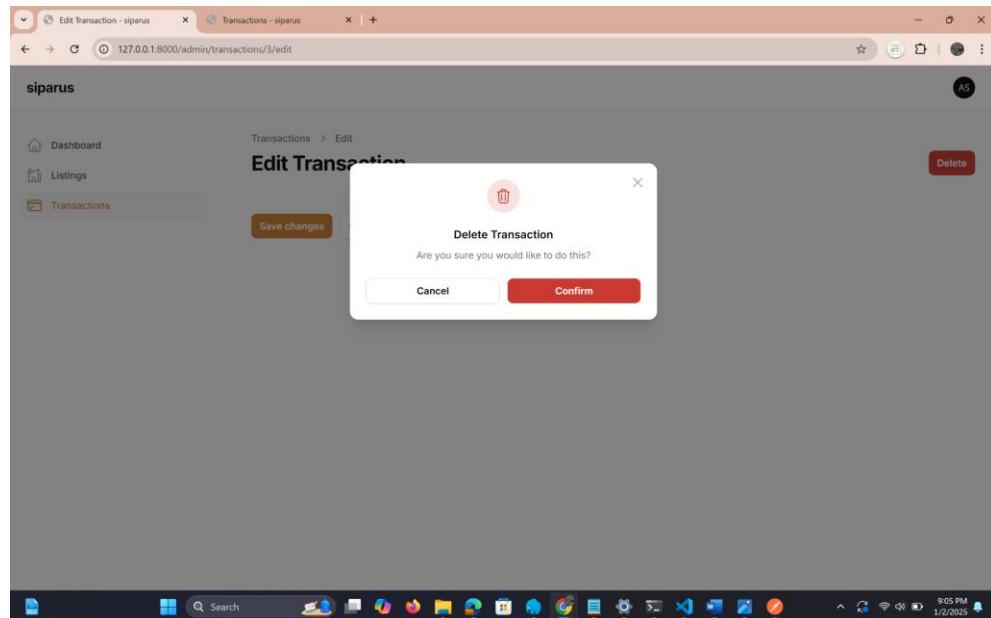
Gambar 5.1.3.2.5. 1 Halaman informasi transaksi



Gambar 5.1.3.2.5. 2 Halaman pop up approved oleh admin



Gambar 5.1.3.2.5. 3 Halaman pop up transaksi berhasil approved



Gambar 5.1.3.2.5. 4 Halaman *edit transactions* untuk menghapus transaksi

User	Listing	Start date	End date	Total bulan	Total price	Status
Dr. Jamaal Raynor I	Bandarharjo Lama	Jul 5, 2024	Nov 5, 2024	4	IDR 3,148,266.00	approved
Khalid Kovacek	Kudu Blok G	Jan 28, 2024	Dec 28, 2024	11	IDR 10,544,702.00	cancelled
Prof. Dovie Lubowitz	Karangroto C Lama	May 7, 2024	Mar 7, 2025	10	IDR 8,766,043.00	approved
Eladio Marquardt	Karangroto C Lama	Sep 2, 2024	Apr 2, 2025	7	IDR 6,136,230.00	waiting
Eladio Marquardt	Sawah Besar Tower 3	Aug 8, 2024	Jun 8, 2025	10	IDR 2,169,310.00	approved
Khalid Kovacek	Pondok Boro	Jan 21, 2024	May 21, 2024	4	IDR 2,834,748.00	approved
Prof. Dovie Lubowitz	Kaligawe Blok A-G	Sep 19, 2024	Aug 19, 2025	11	IDR 11,547,405.00	waiting
Labiba Adinda	Sawah Besar Tower I & 2	Mar 12, 2024	Aug 23, 2024	5	IDR 4,609,260.00	waiting

Gambar 5.1.3.2.5. 5 Halaman berhasil *deleted transactions*

## 5.2 Pengujian

Pada tahap ini, sistem yang telah dikembangkan akan siap digunakan sebagai perangkat lunak operasional. Namun, sebelum digunakan, sistem akan menjalani serangkaian pengujian untuk memastikan bahwa fungsionalitasnya sesuai dengan kebutuhan yang telah ditentukan. Pengujian dilakukan dengan persiapan prosedural

yang matang dan perencanaan yang terstruktur agar proses pengujian berjalan efisien dan efektif.

### 5.2.1 Persiapan Prosedural

Persiapan prosedural bertujuan untuk mengoptimalkan proses pengujian agar dapat mengidentifikasi kesalahan atau bug pada sistem yang telah dibangun, serta memungkinkan perbaikan secara cepat dan tepat. Langkah-langkah prosedur pengujian yang disesuaikan dengan rencana pengujian adalah sebagai berikut:

1. Menyiapkan perangkat lunak dan perangkat keras yang diperlukan untuk pengujian, termasuk alat bantu seperti Postman untuk pengujian API.
2. Menetapkan tujuan pengujian dan skenario pengujian, yang mencakup pengujian *black box* untuk fungsi seperti autentikasi, daftar rusun, dan transaksi, serta *white box* untuk jalur program dan validasi logika sistem.
3. Mendefinisikan kriteria keberhasilan pengujian, yaitu apakah sistem dapat memproses data input dengan benar, memberikan respons sesuai skenario, dan memenuhi kebutuhan fungsional.
4. Membuat kasus uji berdasarkan kebutuhan sistem, seperti uji validasi kredensial pengguna, perhitungan transaksi, dan akses halaman detail rusun, serta mendokumentasikan hasilnya.
5. Mengidentifikasi kesalahan atau bug dalam perangkat lunak, baik pada proses *frontend* maupun *backend*, melalui pengujian skenario yang telah direncanakan.
6. Memperbaiki kesalahan atau bug yang ditemukan selama pengujian, baik dari hasil uji manual maupun melalui Postman.
7. Menentukan penerimaan hasil pengujian, apakah sistem memenuhi kriteria yang ditentukan untuk dinyatakan siap digunakan.
8. Melakukan evaluasi hasil pengujian untuk memastikan semua fungsi telah berjalan sesuai dengan kebutuhan yang ditetapkan, serta mencatat area yang memerlukan peningkatan.

## **5.2.2 Perencanaan Pengujian**

### **5.2.2.1 Black Box**

Pada metode *black box*, fokus pengujian adalah memastikan bahwa sistem berfungsi sesuai dengan kebutuhan tanpa memeriksa kode sumbernya. Pengujian dilakukan terhadap fitur autentikasi, daftar rusun (*listing*), dan transaksi. Untuk autentikasi, direncanakan pengujian terhadap validitas kredensial pada proses login, registrasi, dan logout menggunakan data masukan valid dan tidak valid melalui Postman. Fitur daftar rusun akan diuji untuk memastikan informasi seperti luas, kapasitas, dan harga ditampilkan dengan benar, serta halaman detail rusun dapat diakses. Proses ini juga mencakup pengujian *endpoint* daftar rusun, termasuk menambahkan, mengedit, dan menghapus data. Pada fitur transaksi, pengujian melibatkan validasi data input, perhitungan total harga dan pajak, serta status transaksi (waiting, approved, canceled) melalui pengujian API. Rencana pengujian dapat dilihat pada Lampiran 1.

### **5.2.2.2 White Box**

Pada metode *white box*, pengujian dilakukan dengan memeriksa logika program, validasi form, dan jalur eksekusi kode. Pengujian API menggunakan Postman dimanfaatkan untuk mengamati bagaimana *backend* menangani setiap permintaan, termasuk validasi input, proses otentikasi, dan respons dari setiap *endpoint*.

## **5.2.3 Pelaksanaan Pengujian**

Hasil dari pengujian aplikasi ini dicatat secara rinci, mencakup rangkuman hasil pengujian, catatan pengamatan selama proses pengujian, serta rekomendasi untuk perbaikan atau penyempurnaan sistem. Berdasarkan hasil pengujian menggunakan metode *black box* dan *white box*, termasuk pengujian dengan Postman untuk validasi API, semua pengujian berhasil dilakukan sesuai dengan rencana pengujian dan diterima. Rincian tabel hasil pengujian aplikasi dapat ditemukan pada Lampiran 2 dan gambar di bawah ini disediakan sebagai dokumentasi pendukung yang bertujuan untuk memastikan kualitas serta

kelayakan sistem secara menyeluruh. Gambar 5.2.3.1.1 hingga 5.2.3.1.6 menunjukkan hasil pengujian whitebox Auth. Gambar 5.2.3.2.1 hingga 5.2.3.2.2 menunjukkan hasil pengujian whitebox listing, Gambar 5.2.3.3.1 hingga 5.2.3.3.4 menunjukkan hasil pengujian whitebox transaction secara berturut-turut.

## 1. Auth

```

1 {
2     "success": true,
3     "message": "Sign up successfully",
4     "data": {
5         "name": "Labiba Adinda",
6         "email": "labibaadinda901@gmail.com",
7         "password": "$2y$1$29082qkKtLXiYiSH2y14jeRvz4jNhv5AE2I5A5oyu1v/I1RRz4Qa",
8         "updated_at": "2025-01-02T17:52:15.000000Z",
9         "created_at": "2025-01-02T17:52:15.000000Z",
10        "id": 15,
11        "token": "381QF08pCF0wN1gXdx7o8bqTmYKUZfw73EBvtdcQog7303941c"
12    }
13}

```

Gambar 5.2.3.1. 1 Register berhasil

```

1 {
2     "message": "The email has already been taken."
3     "errors": [
4         {
5             "email": [
6                 "The email has already been taken."
7             ]
8         }
9     ]
10 }

```

Gambar 5.2.3.1. 2 Register gagal

The screenshot shows the Postman interface with a successful login response. The request URL is `((BASE_URL))/login`. The response status is 200 OK, and the response body is:

```

1 {
2     "success": true,
3     "message": "Login successfully",
4     "data": [
5         {
6             "id": 12,
7             "name": "Labiba Adinda",
8             "email": "labibaadinda1@gmail.com",
9             "email_verified_at": null,
10            "role": "user",
11            "password": "$2y$12$gEjY03.Q4LwxhXRZengWSuNBzXn9oIzsSBSikJLoEFo7Mjo4nR0",
12            "remember_token": null,
13            "created_at": "2024-11-24T03:23:51.000006Z",
14            "updated_at": "2024-11-24T03:23:51.000006Z",
15        }
16    ]
17 }

```

Gambar 5.2.3.1. 3 Login berhasil

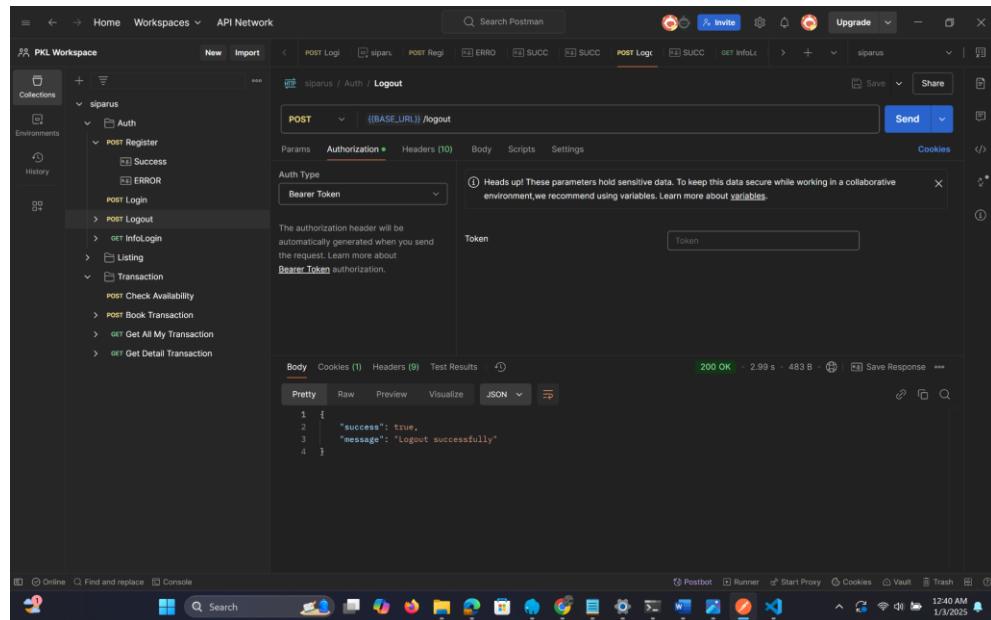
The screenshot shows the Postman interface with an unsuccessful login response. The request URL is `((BASE_URL))/login`. The response status is 422 Unprocessable Content, and the response body is:

```

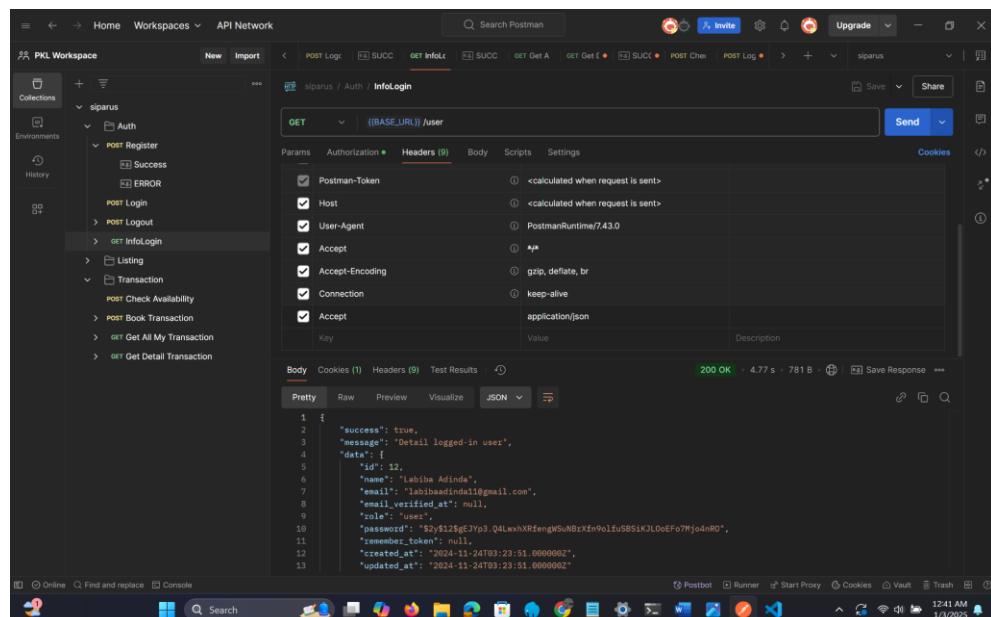
1 {
2     "message": "These credentials do not match our records.",
3     "errors": [
4         {
5             "email": [
6                 "These credentials do not match our records."
7             ]
8         }
9     ]
10 }

```

Gambar 5.2.3.1. 4 Login gagal



Gambar 5.2.3.1. 5 Logout berhasil



Gambar 5.2.3.1. 6 Cek Info Login

## 2. Listing

```

1 {
2   "success": true,
3   "message": "Get all listings",
4   "data": [
5     {
6       "ezxent_page": 1,
7       "data": [
8         {
9           "id": 3,
10          "title": "Sawah Besar Tower I & 2",
11          "slug": "sawah-besar-tower-1-2",
12          "description": "Rusunawas Sawah Besar adalah hunian yang dibangun pada tahun 2018 di Kawasan Gayamsari, Semarang. Lokasi yang strategis memudahkan akses ke berbagai fasilitas publik. menadikannya pilihan tepat bagi mereka yang membutuhkan tempat tinggal terjangkau dan nyaman.",
13          "address": "Jl. Dempo Barat, Samah Besar, Kec. Gayamtasi, Kota Tengah 50163 GAYAMSARI, SAWAH
14          BEsar",
15          "luas_kamar_md": 36,
16          "kapasitas": 84,
17          "price": 210000,
18          "attachments": [
19            "listing/01D764VVP3XV2145YASQNEKBM.jpg"
20          ],
21          "full_support_available": 0,
22        }
23      ]
24    }
25  ]
26}

```

Gambar 5.2.3.2. 1 Cek *Listings* Rusun

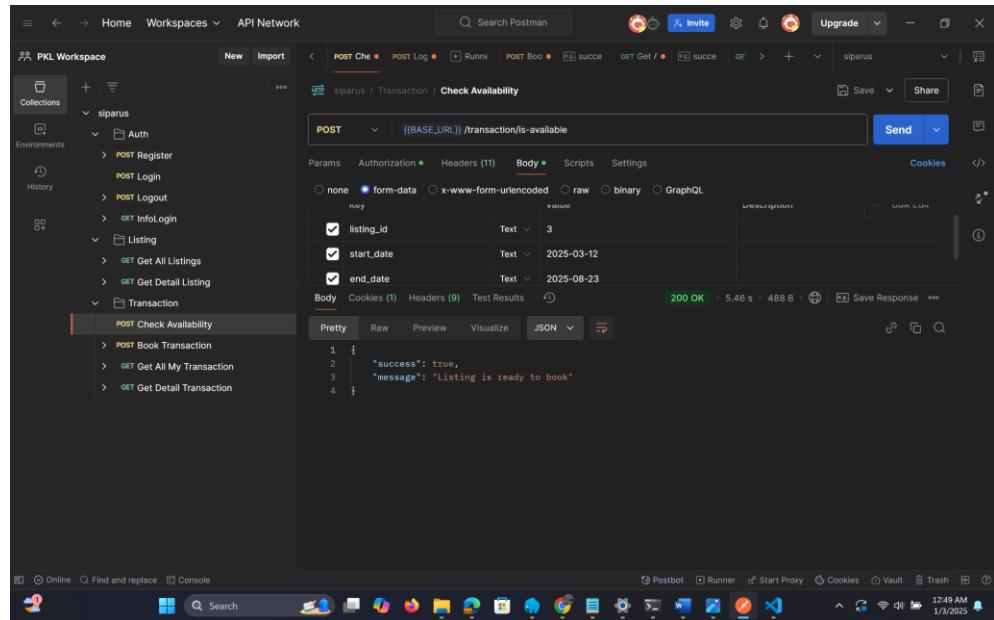
```

1 {
2   "success": true,
3   "message": "Get detail listing",
4   "data": [
5     {
6       "id": 11,
7       "title": "Karangroto C Lama",
8       "slug": "karangroto-clama",
9       "description": "Rusunawa Karangroto C Lama terletak di Kawasan Karangroto, Genuk, Semarang. Dengan lokasi strategis di pusat kota, rusunawa ini memberikan kenyamanan bagi penghuninya dan memudahkan akses ke berbagai fasilitas umum di Semarang.",
10      "address": "Jl. Jatimas XII RT 02 RW 06, Karangroto, Kec. Genuk, Kota Semarang, Jawa Tengah 50117 GENUK, KARANG ROTO",
11      "luas_kamar_md": 27,
12      "kapasitas": 104,
13      "price": 70000,
14      "attachments": [
15        "listing/013F4C00Q418GF3Z01ZHZA2B0P.jpg"
16      ],
17      "full_support_available": 0,
18      "full_support_available": 0
19    }
20  ]
21}

```

Gambar 5.2.3.2. 2 Cek detail *listing* rusun

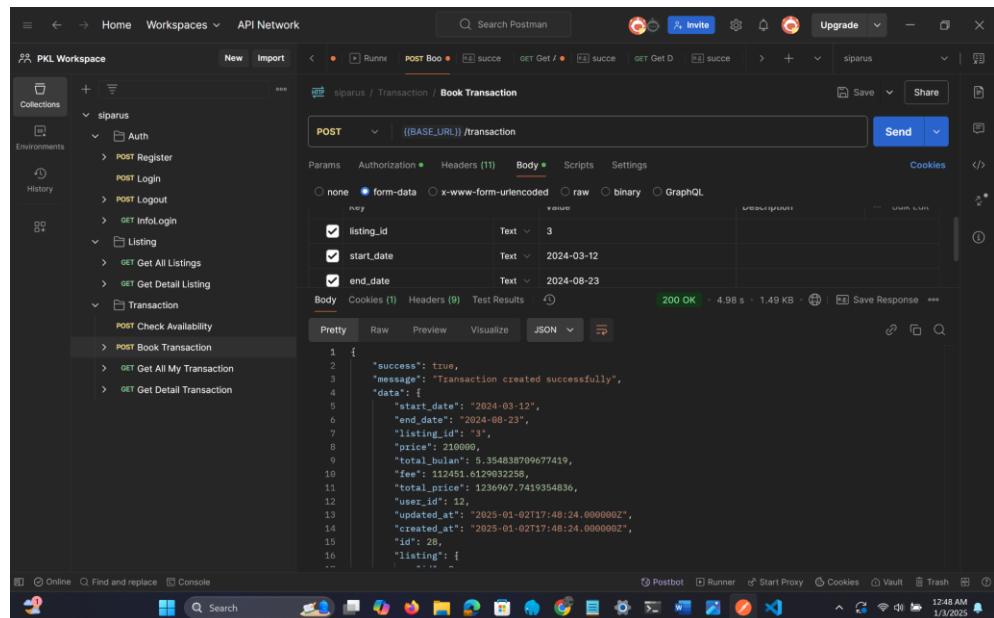
### 3. Transaction



The screenshot shows the Postman application interface. On the left, there's a sidebar with 'PKL Workspace' and a collection named 'siparus'. Under 'siparus', there are several categories like 'Auth', 'Listing', and 'Transaction'. In the 'Transaction' category, there's a 'POST Check Availability' request. The request URL is `({BASE_URL}) /transaction/is-available`. The 'Body' tab is selected, showing a JSON payload with three fields: 'listing\_id' (Text, value: 3), 'start\_date' (Text, value: 2025-03-12), and 'end\_date' (Text, value: 2025-08-23). The response status is 200 OK, and the response body is:

```
1 {  
2   "success": true,  
3   "message": "Listing is ready to book"  
4 }
```

Gambar 5.2.3.3. 1 Cek ketersediaan rusun



This screenshot shows the same Postman session as the previous one, but with a different request. It's a 'POST Book Transaction' request under the 'Transaction' category. The URL is `({BASE_URL}) /transaction`. The 'Body' tab shows the same JSON payload as before. The response status is 200 OK, and the response body is a detailed transaction object:

```
1 {  
2   "success": true,  
3   "message": "Transaction created successfully",  
4   "data": {  
5     "start_date": "2024-03-12",  
6     "end_date": "2024-08-23",  
7     "listing_id": "",  
8     "price": 210000,  
9     "total_bulan": 5.354838709677419,  
10    "fee": 112451.6129032258,  
11    "total_price": 1236967.7419354836,  
12    "user_id": 12,  
13    "updated_at": "2025-01-02T17:48:24.000000Z",  
14    "created_at": "2025-01-02T17:48:24.000000Z",  
15    "id": 28,  
16    "listing": {}  
...  
}
```

Gambar 5.2.3.3. 2 Booking dengan melakukan transaksi

```

1 i
2 "success": true,
3 "message": "List of transaction",
4 "data": [
5     {
6         "current_page": 1,
7         "data": [
8             {
9                 "id": 11,
10                "user_id": 12,
11                "listing_id": 3,
12                "start_date": "2024-03-12",
13                "end_date": "2024-08-23",
14                "price": 782514,
15                "total_bulan": 5,
16                "fee": 419024,
17                "total_price": 4699260,
18                "status": "waiting",
19                "created_at": "2024-11-24T04:20:25.000000Z",
20                "updated_at": "2024-11-24T04:20:25.000000Z"
21            }
22        ]
23    }
24 ]

```

Gambar 5.2.3.3. 3 Transaksi yang telah dilakukan

```

1 i
2 "success": true,
3 "message": "Get Detail of transaction",
4 "data": [
5     {
6         "id": 11,
7         "user_id": 12,
8         "listing_id": 3,
9         "start_date": "2024-03-12",
10        "end_date": "2024-08-23",
11        "price": 782514,
12        "total_bulan": 5,
13        "fee": 419024,
14        "total price": 4699260,
15    }
16 ]

```

Gambar 5.2.3.3. 4 Detail transaksi

### 5.2.3.1 Black Box

Pada pengujian *black box*, fitur autentikasi berfungsi sesuai dengan skenario pengujian. Proses login, registrasi, dan logout berhasil dilakukan, serta sistem mampu memvalidasi kredensial pengguna dengan benar melalui Postman. Fitur daftar rusun berhasil menampilkan informasi secara lengkap, dan

halaman detail dapat diakses tanpa kendala. Pengujian *endpoint* untuk menambah, mengedit, dan menghapus daftar rusun juga memberikan hasil yang sesuai. Pada fitur transaksi, sistem memproses data input dengan benar, termasuk validasi tanggal dan jumlah bulan, perhitungan total harga, dan status transaksi. *Endpoint* transaksi diuji untuk memastikan bahwa setiap status transaksi (waiting, approved, canceled) tercatat dengan benar.

#### 5.2.3.2 *White Box*

Pada pengujian *white box*, logika program dan jalur eksekusi diverifikasi dan tidak ditemukan bug mayor. Validasi form di *backend* berjalan dengan baik, termasuk validasi input kosong atau tidak sesuai format yang diuji melalui Postman.

## **BAB VI**

## **PENUTUP**

### **6.1 Kesimpulan**

Kesimpulan dari pelaksanaan Praktik Kerja Lapangan yang telah dilakukan di Dinas Komunikasi Informatika dan Persandian Kota Semarang, yaitu sebagai berikut.

1. Telah menghasilkan sebuah output berupa Aplikasi SIPARUS yang diimplementasikan dengan bahasa pemrograman PHP, JavaScript, HTML, dan CSS dengan *framework* Laravel dan Next.js. PHPMyAdmin digunakan untuk mengelola database MySQL. Dalam penegembangannya, Aplikasi ini menggunakan model proses *agile* yang spesifiknya adalah *extreme programming* dan diuji menggunakan metode pengujian *black box* dan *white box*.
2. Dengan pengembangan sistem ini, diharapkan dapat mempermudah user dalam menemukan rusun dan membooking kamar pada suatu rusun di Kota Semarang yang dapat digunakan oleh instansi terkait sebagai bentuk layanan publik.

### **6.2 Saran**

Untuk peningkatan lebih lanjut, disarankan agar SIPARUS terus dikembangkan dan diperbarui sesuai dengan dinamika kebutuhan Diskominfo Kota Semarang.

Penting untuk melakukan evaluasi dan penyesuaian secara berkala, serta melibatkan *feedback* dari pengguna untuk memastikan sistem tetap relevan, efektif, dan memenuhi kebutuhan pengguna dengan lebih baik.

## **DAFTAR PUSTAKA**

- Ovan, Andika Saputra. 2020. *CAMI: Aplikasi Uji Validitas dan Reliabilitas Instrumen Penelitian Berbasis Web*. Sulawesi Selatan: Yayasan Ahmar Cendekia Indonesia.
- Mahendra, I., Yanto, D.T.E., 2018. Agile Development Methods Dalam Pengembangan Sistem Informasi Pengajuan Kredit Berbasis Web (Studi Kasus: Bank BRI Unit Kolonel Sugiono). *Jurnal Teknologi dan OpenSource*, 1(2), pp.13-24.
- Dönmez, D.; Grote, G.; Brusoni, S. Routine interdependencies as a source of stability and flexibility. A study of agile software development teams. *Inf. Organ.* 2016, 26, 63–83.
- Abdul Kadir, 2008. *Membuat Aplikasi Web dengan PHP dan Database MySQL*, Penerbit Andi, Yogyakarta
- T. Gumelar, R. Astuti, and A. T. Sunarni, “Sistem Penjualan Online Dengan Metode Extreme Programming,” *J. Telemat.*, vol. 9, no. 2, pp. 87–90, 2017.
- F. Sulianta, Strategi Merancang Arsitektur Sistem Informasi Masa Kini. Jakarta: PT. Elex Media Komputindo, 2019.
- I. G. N. Suryantara, “Merancang Applikasi dengan Metodologi Extreme Programming.” PT. Elex Media Komputindo, Jakarta, 2017.
- M. Melinda, R. I. Borman, and E. R. Susanto, “Rancang Bangun Sistem Informasi Publik Berbasis Web (Studi Kasus : Desa Durian Kecamatan Padang Cermin Kabupaten Pesawaran),” *J. Tekno Kompak*, vol. 11, no. 1, p. 1, 2018.
- Najib, A., & Zain, M. Y. (2020). APLIKASI POINT OF SALE MULTI OUTLET DAN MULTI PAYMENT BERBASIS WEB DAN ANDROID. In *KONVERGENSI* (Vol. 16, Issue 2).

Baxter, G., & Sommerville, I. (2011). "Socio-technical systems: From design methods to systems engineering." *Interacting with Computers*, 23(1), 4–17.

Yuniarti, R., Santi, I. H., & Puspitasari, W. D. (2022). PERANCANGAN APLIKASI POINT OF SALE UNTUK MANAJEMEN PEMESANAN BAHAN PANGAN BERBASIS FRAMEWORK LARAVEL. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 6, Issue 1).

A. E. Kumala, R. I. Borman, and P. Prasetyawan, “Sistem Informasi Monitoring Perkembangan Sapi Di Lokasi Uji Performance (Studi Kasus : Dinas Peternakan Dan Kesehatan Hewan Provinsi Lampung),” *J. Tekno Kompak*, vol. 12, no. 1, p. 5, 2018.

Wira, D., Putra, T., & Andriani, R. (2019). Unified Modelling Language ( UML ) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD. 7(1).

Nugroho, I., Listiyono, H., & Anwar, S. N. (2017). PERANCANGAN UNIFIED MODELLING LANGUAGEAPLIKASI PRASARANA SARANA PENDUKUNG PARIWISATA KOTA SEMARANG, 3(1), 90–95.

Booch, G., Rumbaugh, J., & Jacobson, I. (1998). *The Unified Modeling Language User Guide*. Addison-Wesley Professional.

Bin Tahir, T., Rais, M., & Hs, M. A. (2019). Aplikasi Point Of Sales Menggunakan Framework Laravel Point Of Sales Application using Laravel Framework. *Jurnal Informatika Dan* <https://doi.org/10.33387/jiko>

Sitinjak, D., Maman, & Suwita, J. (2020). 164-355-1-PB. JURNAL IPSIKOM, 8. Stikom. (2020). Memahami Robustness Diagram dan Contohnya.

Sari, A.P., Suhendi, 2020. Rancang Bangun Sistem Informasi Pengelolaan Talent Filem Berbasis Aplikasi Web. *Jurnal Informatika Terpadu*, 6(1), pp.29-37.

Yani, A., Saputra, B, 2018. Rancang Bangun Sistem Informasi Evaluasi Siswa dan Kehadiran Guru Berbasis Web (Studi Kasus di SMK Nusa Putra Kota Tangerang). *Jurnal Petir*, 11(2), pp.107-124.

Pressman, Roger S, (2010), Rekayasa Perangkat Lunak: Pendekatan Praktisi (Buku Satu), Andi, Yogyakarta.

Thakkar, J. (2020). *Full-Stack React Projects*. Springer.

[https://doi.org/10.1007/978-1-4842-5869-9\\_3](https://doi.org/10.1007/978-1-4842-5869-9_3)

Libby, A. (2023). *Practical Next.js Projects*. Springer.

<https://doi.org/10.1007/978-1-4842-9612-7>

Cholifah, W. N., Yulianingsih, & Sagita, S. M. (2018). Pengujian Black Box Testing Pada Aplikasi Action & Strategy Berbasis Android Dengan Teknologi Phonegap. *Jurnal String*, 206-210.

Irawan, Y. (2017). Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja UPT BLK Kabupaten Kudus. *Sentra Penelitian Engineering dan Edukasi*, 56-63.

## **LAMPIRAN-LAMPIRAN**

Lampiran 1. Rencana Pengujian Aplikasi

Kelas Uji	Butir Uji	Use Case	Kode Uji	Aktor
User dan admin dapat melakukan <i>login</i>	Admin memasukkan rmail dan password yang benar untuk <i>login</i> menggunakan kredensial admin di web admin.	SRS-F-1	U-1-A	Admin
	User memasukkan email dan password yang benar untuk <i>login</i> menggunakan kredensial user di web user.		U-1-B	User
	Admin memasukkan email dan password yang salah		U-1-C	Admin
	User memasukkan email dan password yang salah		U-1-D	User
User dan admin dapat melakukan <i>logout</i>	Admin keluar dari aplikasi dengan menekan pilihan menu <i>logout</i>	SRS-F-2	U-2-A	Admin
	User keluar dari aplikasi dengan menekan pilihan menu <i>logout</i>		U-2-B	User
User dapat melakukan registrasi akun	User melakukan registrasi dengan email yang belum pernah registrasi sebelumnya	SRS-F-3	U-3-A	User
	User melakukan registrasi dengan email yang pernah registrasi sebelumnya		U-3-B	
Admin dapat menghapus data transaksi	Admin menghapus data transaksi user dari database.	SRS-F-4	U-4-A	Admin
Admin dapat menambahkan data rusun	Admin menambahkan data rusun baru ke dalam sistem.	SRS-F-5	U-5-A	Admin

Admin dapat mengubah detail rusun	Admin mengubah detail rusun yang ada di sistem.	SRS-F-6	U-6-A	Admin
Admin dan User dapat melihat informasi lengkap rusun	Admin melihat informasi lengkap rusun tertentu di web admin.	SRS-F-7	U-7-A	Admin
	User melihat informasi lengkap rusun tertentu di web user.		U-7-B	User
Admin dapat menghapus data rusun	Admin menghapus data rusun yang ada di sistem.	SRS-F-8	U-8-A	Admin
Admin dan User dapat melihat daftar rusun	Admin melihat daftar rusun yang tersedia di sistem web admin.	SRS-F-9	U-9-A	Admin
	User melihat daftar rusun yang tersedia di sistem web user.		U-9-B	User
User dapat membooking kamar	User membooking kamar berdasarkan tanggal dan ketersediaan kamar dalam kondisi sudah <i>login</i> .	SRS-F-10	U-10-A	User
	User membooking kamar berdasarkan tanggal dan ketersediaan kamar dalam kondisi belum <i>login</i> .		U-10-B	User
Admin dan User dapat melihat daftar transaksi	Admin melihat daftar transaksi	SRS-F-11	U-11-A	Admin
	User melihat daftar transaksi		U-11-B	User
User dapat mengunggah bukti	User mengunggah bukti pembayaran melalui media komunikasi WhatsApp untuk mengkonfirmasi bukti pembayaran.	SRS-F-12	U-12-A	User

pembayaran melalui media komunikasi WhatsApp.				
Admin dapat memberikan konfirmasi pembayaran yang telah divalidasi kepada User.	Admin memberikan konfirmasi pembayaran yang telah divalidasi kepada User.	SRS-F-13	U-13-A	Admin

Lampiran 2. Hasil Pengujian Aplikasi

Kode Uji	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil	Kesimpulan
U-1-A	Admin memasukkan email dan password yang benar	1. Memasukkan email dan password yang benar 2. Menekan tombol <i>Sign In</i>	Email dan password yang benar	<i>Redirect</i> ke halaman landing page web admin	<i>Redirect</i> ke halaman landing page web admin	Diterima
U-1-B	User memasukkan email dan password yang benar	1. Memasukkan email dan password yang benar 2. Menekan tombol <i>Sign In</i>	Email dan password yang benar	<i>Redirect</i> ke halaman landing page dengan notifikasi berhasil <i>Sign In</i>	<i>Redirect</i> ke halaman landing page dengan notifikasi berhasil <i>Sign In</i>	Diterima
U-1-C	Admin memasukkan email dan password yang salah	1. Memasukkan email dan password yang salah 2. Menekan tombol <i>Sign In</i>	Email dan password yang salah	Tetap di halaman login dengan notifikasi gagal <i>login</i> .	Tetap di halaman login dengan notifikasi gagal <i>login</i> .	Diterima
U-1-D	User memasukkan email dan password yang salah	1. Memasukkan email dan password yang salah 2. Menekan tombol <i>Sign In</i>	Email dan password yang salah	Tetap di halaman login dengan notifikasi gagal <i>login</i> .	Tetap di halaman login dengan notifikasi gagal <i>login</i> .	Diterima

U-2-A	Admin melakukan <i>logout</i>	1. Menekan menu <i>Logout</i> dengan mengarahkan kursor ke icon profile, terdapat menu <i>logout</i> .	-	<i>Redirect</i> ke halaman <i>Login</i> .	<i>Redirect</i> ke halaman <i>Login</i> .	Diterima
U-2-B	User melakukan logout	1. Menekan menu <i>Logout</i> dengan mengarahkan kursor ke icon profile, terdapat menu <i>logout</i> .	-	<i>Redirect</i> ke halaman <i>Landing Page</i> dengan kondisi belum masuk akun	<i>Redirect</i> ke halaman <i>Landing Page</i> dengan kondisi belum masuk akun	Diterima
U-3-A	User melakukan registrasi dengan email yang belum terdaftar	1. Menekan tombol <i>sign up</i> 2. Isi data nama, nomor hp, email yang belum terdaftar dan password 3. Menekan <i>I agree with terms and conditions</i> 4. Menekan <i>Sign Up</i>	Email yang belum terdaftar	<i>Redirect</i> ke halaman <i>landing page</i> dengan notifikasi berhasil <i>Sign Up</i> dan dalam kondisi sudah masuk akun	<i>Redirect</i> ke halaman <i>landing page</i> dengan notifikasi berhasil <i>Sign Up</i> dan dalam kondisi sudah masuk akun	Diterima
U-3-B	User melakukan	1. Menekan	Email yang	Tetap di halaman	Tetap di halaman	Diterima

	registrasi dengan email yang sudah terdaftar	2. tombol <i>sign up</i> 3. Isi data nama, nomor hp, email yang sudah terdaftar dan password 3. Menekan <i>I agree with terms and conditions</i> 4. Menekan <i>Sign Up</i>	sudah terdaftar	<i>Sign Up</i> dengan <i>pop up Sign Up</i> gagal “The email has already been taken”	<i>Sign Up</i> dengan <i>pop up Sign Up</i> gagal “The email has already been taken”	
U-4-A	Admin menghapus data transaksi user	1. Menekan menu <i>Transactions</i> di web admin 2. Menekan transaksi yang ingin dihapus 3. Redirect ke halaman edit transaksi 4. Menekan tombol <i>delete</i> 5. Muncul pop up untuk mengkonfirmasi transaksi yang ingin dihapus 6. Menekan <i>Confirm</i>	-	<i>Redirect</i> ke halaman daftar transaksi dan transaksi sudah berhasil dihapus dengan <i>pop up Deleted</i>	<i>Redirect</i> ke halaman daftar transaksi dan transaksi sudah berhasil dihapus dengan <i>pop up Deleted</i>	Diterima

U-5-A	Admin menambahkan data rusun	<ol style="list-style-type: none"> <li>1. Menekan menu <i>Listings</i> di web admin</li> <li>2. <i>Redirect</i> ke halaman <i>Listings</i>, admin menekan tombol <i>New listing</i> untuk menambahkan data rusun</li> <li>3. <i>Redirect</i> ke halaman <i>Create Listing</i>, isi listing baru dengan benar</li> <li>4. Menekan tombol <i>Create</i></li> </ol>	Data rusun yang ingin ditambahkan	<i>Redirect</i> ke halaman <i>Edit Listings</i> dengan pop up <i>Created</i>	<i>Redirect</i> ke halaman <i>Edit Listings</i> dengan pop up <i>Created</i>	Diterima
U-6-A	Admin mengubah detail rusun	<ol style="list-style-type: none"> <li>1. Menekan menu <i>Listings</i> di web admin</li> <li>2. <i>Redirect</i> ke halaman <i>Listings</i>, admin menekan menu edit</li> <li>3. Edit detail rusun</li> <li>4. Menekan tombol <i>Save Changes</i></li> </ol>	Data rusun yang ingin diubah sesuai ketentuan	Tetap di halaman edit <i>listings</i> dengan pop up <i>Saved</i> dan kondisi sudah di edit	Tetap di halaman edit <i>listings</i> dengan pop up <i>Saved</i> dan kondisi sudah di edit	Diterima

		5. Muncul <i>pop up</i> <i>Saved</i>				
U-7-A	Admin melihat informasi lengkap rusun tertentu di web admin.	1. Menekan menu <i>Listings</i> di web admin 2. <i>Redirect</i> ke halaman <i>Listings</i> 3. Admin melihat daftar rusun 4. Admin klik rusun yang ingin dilihat detail nya	-	<i>Redirect</i> ke halaman <i>listings</i>	<i>Redirect</i> ke halaman <i>listings</i>	Diterima
U-7-B	User melihat informasi lengkap rusun tertentu di web user.	1. User berada di <i>landing page</i> dan melihat <i>List Rusun</i> 2. User klik rusun yang ingin dilihat detailnya	-	<i>Redirect</i> ke halaman detail <i>listings</i> dan admin dapat melihat informasi lengkap rusun	<i>Redirect</i> ke halaman detail <i>listings</i> dan admin dapat melihat informasi lengkap rusun	Diterima
U-8-A	Admin menghapus data rusun yang ada di sistem.	1. Menekan menu <i>Listings</i> di web admin 2. <i>Redirect</i> ke halaman <i>Listings</i> 3. Admin klik	-	Tetap di halaman <i>listings</i> dengan kondisi rusun sudah dihapus	Tetap di halaman <i>listings</i> dengan kondisi rusun sudah dihapus	Diterima

		<p>tombol <i>delete</i> di bagian rusun yang ingin di delete</p> <ol style="list-style-type: none"> <li>4. Muncul pop up konfirmasi <i>delete</i> rusun tersebut</li> <li>5. Klik <i>confirm</i></li> </ol>				
U-9-A	Admin melihat daftar rusun yang tersedia di sistem web admin.	<ol style="list-style-type: none"> <li>1. Menekan menu <i>Listings</i> di web admin</li> <li>2. <i>Redirect</i> ke halaman <i>Listings</i></li> <li>3. Admin melihat daftar rusun</li> </ol>	-	<i>Redirect</i> ke halaman <i>listings</i> dan admin dapat melihat daftar rusun	<i>Redirect</i> ke halaman <i>listings</i> dan admin dapat melihat daftar rusun	Diterima
U-9-B	User melihat daftar rusun yang tersedia di sistem web user.	<ol style="list-style-type: none"> <li>1. User berada di <i>landing page</i> dan melihat <i>List Rusun</i></li> </ol>	-	Tetap di halaman <i>landing page</i> dan dapat melihat daftar rusun di <i>landing page</i>	Tetap di halaman <i>landing page</i> dan dapat melihat daftar rusun di <i>landing page</i>	Diterima
U-10-A	User membooking kamar berdasarkan tanggal dan ketersediaan kamar dalam kondisi sudah login.	<ol style="list-style-type: none"> <li>1. User berada di <i>landing page</i> dan melihat <i>List Rusun</i></li> <li>2. User klik rusun yang ingin dibooking</li> </ol>	Data rusun yang ingin dibooking	<i>Redirect</i> ke Whatsapp dan mengkonfirmasi pembayaran	<i>Redirect</i> ke Whatsapp dan mengkonfirmasi pembayaran	Diterima

		<p>3. Redirect ke halaman detail rusun dan user mengeset tanggal menyewa rusun yang ingin di booking</p> <p>4. Klik Book Now</p> <p>5. Redirect ke halaman checkout dengan informasi tujuan transfer.</p> <p>6. User mentransfer ke nomor rekening yang tertera</p> <p>7. User klik “<i>I agree with terms &amp; conditions</i>”</p> <p>8. User klik <i>Make a Payment</i></p> <p>9. User klik Message Owner untuk konfirmasi pembayaran</p>			
--	--	--	--	--	--

U-10-B	User membooking kamar berdasarkan tanggal dan ketersediaan kamar dalam kondisi belum login.	<ol style="list-style-type: none"> <li>1. User berada di <i>landing page</i> dan melihat <i>List Rusun</i></li> <li>2. User klik rusun yang ingin dibooking</li> <li>3. <i>Redirect</i> ke halaman detail rusun dan user mengeset tanggal menyewa rusun yang ingin dibooking</li> <li>4. Klik <i>Book Now</i></li> <li>5. Muncul <i>pop up</i> “<i>Something went wrong, please login first</i>”</li> </ol>	Data rusun yang ingin dibooking	Tetap dihalaman yang sama dengan <i>pop up</i> “ <i>Something went wrong, please login first</i> ”	Tetap dihalaman yang sama dengan <i>pop up</i> “ <i>Something went wrong, please login first</i> ”	Diterima
U-11-A	Admin melihat daftar transaksi yang tersedia di sistem web admin.	<ol style="list-style-type: none"> <li>1. Menekan menu <i>Transactions</i> di web admin</li> <li>2. <i>Redirect</i> ke halaman <i>transactions</i></li> <li>3. Admin melihat daftar transaksi</li> </ol>	-	<i>Redirect</i> ke halaman <i>Transactions</i> dan admin dapat melihat daftar transaksi	<i>Redirect</i> ke halaman <i>Transactions</i> dan admin dapat melihat daftar transaksi	Diterima
U-11-B	User melihat daftar	1. User	-	<i>Redirect</i> ke	<i>Redirect</i> ke	Diterima

	transaksi yang tersedia di sistem web user.	<p>mengarahkan cursor ke ikon profile dan mengeklik <i>Dashboard</i>.</p> <ol style="list-style-type: none"> <li>2. <i>Redirect</i> ke <i>dashboard</i> User</li> <li>3. User klik menu <i>My Transactions</i> di web User</li> <li>4. <i>Redirect</i> ke halaman <i>my transactions</i></li> <li>5. User melihat daftar transaks</li> </ol>		halaman <i>my transactions</i> dan User dapat melihat daftar transaksi	halaman <i>my transactions</i> dan User dapat melihat daftar transaksi	
U-12-A	User mengunggah bukti pembayaran melalui media komunikasi WhatsApp untuk mengkonfirmasi bukti pembayaran.	<ol style="list-style-type: none"> <li>1. User klik <i>Message Owner</i> untuk konfirmasi pembayaran</li> <li>2. User mengirim pesan konfirmasi dan mengirim bukti pembayaran</li> </ol>	Data bukti pembayaran	<i>Redirect</i> ke Whatsapp dan mengirim bukti pembayaran untuk konfirmasi pembayaran	<i>Redirect</i> ke Whatsapp dan mengirim bukti pembayaran untuk konfirmasi pembayaran	Diterima
U-13-A	Admin memberikan konfirmasi pembayaran yang telah divalidasi	1. Menekan menu <i>Transactions</i> di web admin	-	Tetap di halaman <i>transactions</i> dengan transaksi	Tetap di halaman <i>transactions</i> dengan transaksi	Diterima

	kepada User.	<ol style="list-style-type: none"> <li>2. Menekan transaksi yang ingin di <i>approved</i></li> <li>3. Muncul <i>pop up</i> untuk mengkonfirmasi transaksi pembayaran</li> <li>4. Menekan <i>Confirm</i></li> </ol>		<p>yang di <i>approve</i> oleh admin status <i>waiting</i> berubah menjadi <i>approved</i> dan muncul <i>pop up</i> “<i>Transactions has been approved</i>”</p>	<p>yang di <i>approve</i> oleh admin status <i>waiting</i> berubah menjadi <i>approved</i> dan muncul <i>pop up</i> “<i>Transactions has been approved</i>”</p>	
--	--------------	--	--	---	---	--

### Lampiran 3. Surat Keterangan Telah Melaksanakan PKL



### PEMERINTAH KOTA SEMARANG DINAS KOMUNIKASI, INFORMATIKA, STATISTIK DAN PERSANDIAN

Jl. Pemuda 148 Telp. (024)3549446 Fax. (024)3549446 Semarang 50132

Nomor : B / 1119 / 423.4 / III / 2024

Semarang, 1 Maret 2024

Lamp. :

Perihal : Surat Keterangan Selesai Kerja Praktek  
(Magang)

Kepada :

Yth. Dekan Fakultas Sains dan Matematika

Universitas Diponegoro

di

SEMARANG

1. Dasar : Surat Dekan Fakultas Sains dan Matematika Universitas Diponegoro Semarang Nomor AK.015 Tanggal 27 Oktober 2023 Perihal Permohonan Surat Pengantar.

2. Schubungan hal tersebut diatas, diberitahukan mahasiswa Universitas Diponegoro Semarang atas nama:

NAMA	NIM	PROGRAM STUDI
Labiba Adinda Zahwana	24060121140111	Informatika

Telah selesai melaksanakan Kerja Praktek (Magang) di Dinas Komunikasi, Informatika, Statistik dan Persandian Kota Semarang mulai tanggal 2 Januari 2024 sampai dengan tanggal 2 Maret 2024. Mahasiswa tersebut melaksanakan tugas dengan baik dan bertanggung jawab.

3. Demikian surat keterangan ini kami buat, atas perhatiannya dan kerjasamanya kami ucapan terimakasih.

**An. Kepala Dinas Komunikasi, Informatika,  
Statistik dan Persandian Kota Semarang  
Kepala Bidang Pengembangan Komunikasi Publik**

SUCAHYO KUSWIRANTOMO, SH, S.Sos, MH.  
  


#### Lampiran 4. Dokumen Penilaian dari Instansi

DEPARTEMEN INFORMATIKA  
FAKULTAS SAINS DAN MATEMATIKA  
UNIVERSITAS DIPONEGORO

#### Formulir Penilaian PKL

Nama Mahasiswa : Labiba Adinda Zahwana  
Nomor Induk Mahasiswa : 24060121140111  
Judul PKL : Sistem Informasi Pembayaran dan Managemen Invoice Rumah Susun Kota Semarang  
Instansi / Perusahaan : Dinas Komunikasi, Informatika, Statistik, dan Persandian Kota Semarang  
Periode PKL : 2 Januari 2024 s.d. 2 Maret 2024  
Pembimbing Lapangan : Bapak Asdani Kindarto, S.Sos, M.Eng, Ph.D  
Kontak Pembimbing : 08121352952

No	Materi Penilaian	Nilai (0-100)*
1.	Kedisiplinan dan etika	97
2.	Kemampuan berkomunikasi dan bekerja sama	97
3.	Pemahaman terhadap permasalahan	97
4.	Pengetahuan baik teoritis maupun praktik	97
Rata-rata		97

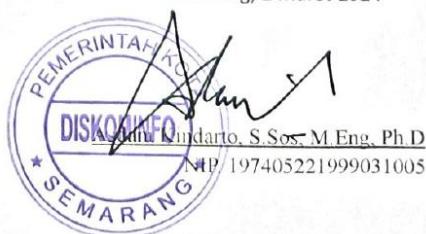
\*) diisi sesuai tabel nilai dan kategori di akhir halaman ini

Catatan:

Ditingkatkan kemampuan pemrograman.

Demikian keputusan nilai dari kami, untuk dapat dipergunakan sebaik-baiknya.

Semarang, 2 Maret 2024



Lampiran 5. Kartu Keikutsertaan Seminar PKL



**KARTU KEIKUTSERTAAN SEMINAR PKL**  
**DEPARTEMEN ILMU KOMPUTER/INFORMATIKA**  
**FAKULTAS SAINS DAN MATEMATIKA UNDIP**

Nama : LABIBA ADINDA ZAHWANA  
 NIM : 24060121140111

Seminar yang diikuti					
No.	Tanggal	Judul	Oleh: (Nama/NIM)	Keaktifan	TTD Dosen
1.	7 Juni 2023	Web based Application MyBranch Mandiri Tunas Finance	Jason Al-Hilal Sabda Dewa 24060120130003	X	
2.	7 Juni 2023	web based Application MyBranch Mandiri Tunas finance	Nadiatur Salam 24060120120006	X	
3.	8 Juni 2023	Web based Application MyBranch Mandiri Tunas finance	Fellia Gessangie Yekanchana Puteri 24060120130059	X	
4.	26/06/2023	Aplikasi E-Taraf Berbasis web di Dinas Penkerdungan Perempuan dan Perlindungan Anak kota Semarang	Maya Nurul Hikmah 24060120120004		
5.	26/06/2023	Sistem Informasi Pinjaman Berbasis Web berdasarkan Pengembangan dan Peningkatan Keterbukaan Akses Semarang	Nadya Muntazah 24060120120027		
6.	26/06/2023	Sistem Informasi Pinjaman Mitra UMKM Pt TASPEN	Alisyah Rahma S. 24060120120021		
7.	26/06/2023	Sistem Informasi Pinjaman Mitra UMKM PT TASPEN	Ratih Adiningrum 24060120140113		
8.	26/06/2023	Sistem Informasi Gelar Karya Berbasis web	Muhammad Sultan Jodi Perwira 24060120140009		

Keterangan  
 Aktif : ✓ (Nilai 4)  
 Tidak Aktif : ✗ (Nilai 2)

Koordinator PKL/KKL

*Sandy Kurniawan, S.Kom., M.Kom.*  
 NIP. 199603032022041001



**KARTU KEIKUTSERTAAN SEMINAR PKL**  
**DEPARTEMEN ILMU KOMPUTER/INFORMATIKA**  
**FAKULTAS SAINS DAN MATEMATIKA UNDIP**

Nama : Labiba Adinda Zahwana  
 NIM : 24060121140111

Seminar yang diikuti					
No.	Tanggal	Judul	Oleh: (Nama/NIM)	Keaktifan	TTD Dosen
9.	26/06/2023	Sistem Informasi Gelar Karya Berbasis web	Alfian Afra Zahwana 24060120130114		
10.	26/06	Sentralisasi Kegiatan Pengembangan Pendidikan Berbasis Media Interaktif. Hasil dibuatkan Pidildingan dengan media Digitalisasi sistem order B2B	Fathani Durghamy 24060120140009		
II.					

Keterangan  
 Aktif : ✓ (Nilai 4)  
 Tidak Aktif : ✗ (Nilai 2)

Koordinator PKL/KKL

*Sandy Kurniawan, S.Kom., M.Kom.*  
 NIP. 199603032022041001

Lampiran 6. Kartu Bimbingan PKL



**KARTU BIMBINGAN PKL  
DEPARTEMEN ILMU KOMPUTER/INFORMATIKA  
FAKULTAS SAINS DAN MATEMATIKA UNDIP**

Nama : Lubiba Adinda Zahwana  
 NIM : 24060121140111  
 Dosen Pembimbing : Prajanto Wahyu Adi  
 Mulai Bimbingan : Semester 7..... Tahun Ajaran 2024.

Bimbingan ke-	Tanggal	Materi	TTD Dosen
1	21/12 2024	Latar Belakang Masalah	
2	23/12 2024	Testing website	
3	27/12 2024	Revisi Laporan	
4	30/12 2024	Revisi diagram & laporan	
5	2/01 2024	Revisi laporan	
6	5/01 2024	Revisi laporan	

Koordinator PKL/KKL

Sandy Kurniawan, S.Kom, M.Kom  
 NIP. 199603032022041001

Dosen Pembimbing PKL

Prajanto Wahyu Adi, M.Kom  
 NIP. 1988032220121010

Lampiran 7. Daftar Hadir Seminar PKL

DEPARTEMEN INFORMATIKA  
FAKULTAS SAINS DAN MATEMATIKA  
UNIVERSITAS DIPONEGORO

DAFTAR HADIR  
SEMINAR PRAKTIK KERJA LAPANGAN

Nama : Labiba Adinda Zahwana  
 NIM : 24060121140111  
 Dosen Pembimbing : Prajanto Wahyu Adi, M.Kom.  
 Judul PKL : Sistem Informasi Pembayaran dan Manajemen *Invoice Rumah Susun*  
 Kota Semarang  
 Hari, tanggal : Jumat, 06 Januari 2025  
 Waktu Seminar : 10.10 – 12.10 WIB

No.	Nama	NIM	Tanda Tangan
1.	Alfonso Clement Sugiantio	24060122130080	1.
2.	Bisma Wira Adi Wicaksana	24060122140120	2.
3.	Azzam Saefudin Rasyidi	24060122130076	3.
4.	Akbar Maryon Bagaskara	24060121120019	4.
5.	Muhammad Rizki	24060121130086	5.
6.	Nur Annisa Meutia Fajrin	24060121120010	6.
7.	Muhammad Afiat Yulianto	24060121140191	7.
8.	Angga Syaputra	24060121140137	8.
9.	M. Alpy Mukhlis	24060121120001	9.
10.	Muhammad Nafal Arkan	24060121130073	10.
11.	Clara Zira Nitella	24060122130057	11.
12.	Fendi Ardianto	24060122130677	12.
13.	Alizza Afra Afifah	24060121140135	13.
14.	Ayeng Nurhaliza Ar R.	24060121140109	14.
15.	Kharisma Andini Putri	24060121140119	15.
16.	Arynda Anna Salasbiela	24060121120023	16.
17.	Muhamad Ridwan R.	24060121130075	17.
18.	Andre Fitara	24060121140129	18.
19.			19.
20.			20.
21.			21.
22.			22.
23.			23.
24.			24.
25.			25.
26.			26.
27.			27.
28.			28.
29.			29.
30.			30.

Lampiran 8. Tanya Jawab Seminar PKL

No	Pertanyaan	Jawaban
1	<p>Apakah terdapat constraint ketika 2 user memesan rusun secara bersamaan jika kamar tersisa 1 pada sebuah rusun</p> <p>Azzam Sacfudin 24060122130076</p>	Tidak ada, karena untuk user yang waktu pembookingannya lebih cepat dari user lain akan diapprove admin jika kamar tersisa 1.
2	<p>Ketika user memiliki transaksi pending, apakah bisa membooking kamar di rusun lain?</p> <p>Alfonso Clement Sutantio 24060122130080</p>	User bisa membooking kamar di rusun lain walaupun memiliki transaksi pending.
3	<p>Ketika terdapat oknum yang tidak bertanggung jawab dengan melakukan transaksi di berbagai rusun, apa yang admin akan lakukan?</p> <p>Alfonso Clement Sutantio 24060122130080</p>	Transaksi tersebut akan diabaikan, karena admin hanya akan approve sebuah transaksi jika sebuah user telah mengirim bukti transaksinya.
4	<p>Bagaimana user mengetahui status bookingnya?</p> <p>Nur Annisa Merarti Fajrin 24060121120010</p>	User bisa menuju ke halaman dashboard dan menekan menu my transactions atau menekan pilihan my transactions saat menghover ke ikon profile di landing page.

Mengetahui,  
Dosen Pembimbing

Prajanto Wahyu Adi, M.Kom  
NIP. 198803222020121010

Notulis,

Akbar Maryan Bagaskara  
NIM. 24060121120019