

# **DOKUMEN DESKRIPSI PERANCANGAN PERANGKAT LUNAK**

## **Management Information System Roomify**

### **1. PENDAHULUAN**

#### **1.1 Tujuan Dokumen**

Software Design Document (SDD) ini disusun untuk menjelaskan secara rinci arsitektur, desain sistem, struktur data, dan komponen perangkat lunak dari aplikasi Roomify, yaitu sistem informasi kos-kosan berbasis web yang dikembangkan menggunakan Laravel Breeze, SQLite, dan Bootstrap. Dokumen ini berfungsi sebagai panduan teknis bagi pengembang, penguji, serta pemangku kepentingan lainnya agar proses implementasi, pengujian, dan pemeliharaan sistem dapat dilakukan secara konsisten dan sesuai dengan rancangan yang telah ditetapkan.

#### **1.2 Ruang Lingkup**

Dokumen ini mencakup:

- Deskripsi umum sistem Roomify.
- Arsitektur perangkat lunak (Web/PWA).
- Desain modul-modul utama (Pencarian, Booking, Transaksi, Admin).
- Deskripsi database dan ERD.
- Alur proses bisnis (Business Logic).
- Interface pengguna.
- Keamanan dan integrasi (Payment Gateway).

#### **1.3 Definisi dan Akronim**

- **SDD:** *Software Design Document*
- **MVC:** *Model-View-Controller*
- **PWA:** *Progressive Web App* (Aplikasi web dengan kapabilitas *mobile*)
- **DB:** Database
- **CRUD:** *Create, Read, Update, Delete*
- **ERD:** *Entity Relationship Diagram*
- **Midtrans:** Layanan gerbang pembayaran (*Payment Gateway*)
- **Booking:** Proses reservasi jadwal kunjungan survei kos

#### **1.4 Referensi**

- Dokumen Wawancara Sistem - Pemilik Kos.
- Dokumen Wawancara Sistem - Pencari Kos.
- Diagram UML Roomify (Use Case, Sequence, Class, ERD).

## **2. DESKRIPSI SISTEM SECARA UMUM**

### **2.1 Visi dan Misi Sistem**

**Visi:** Menyediakan platform informasi kos-kosan yang terintegrasi, mudah diakses, dan terpercaya, sehingga mempermudah proses pencarian, pengelolaan, serta transaksi penyewaan kos bagi penyewa, pemilik kos, dan admin sistem.

**Misi:** - Menyediakan layanan pencarian kos yang lengkap, akurat, dan dapat difilter berdasarkan kebutuhan spesifik. - Mendukung proses penyewaan kos secara digital, mulai dari melihat detail, booking kunjungan, hingga pembayaran. - Menyediakan sarana komunikasi langsung antara penyewa dan pemilik kos melalui sistem chat. - Memfasilitasi pemilik kos dalam mengelola informasi kos, termasuk pendaftaran, pembaruan status, dan monitoring penyewa. - Menjamin validitas data kos melalui proses verifikasi oleh admin. - Menghadirkan platform yang responsif, aman, dan mudah digunakan oleh seluruh pengguna.

### **2.2 Fitur-Fitur Utama Sistem**

#### **1. Modul Pencarian dan Filter Kos:**

- Pencarian kos berdasarkan lokasi, harga, fasilitas, atau kriteria lain.
- Menampilkan daftar kos yang telah diverifikasi admin.
- Menyediakan detail kos lengkap, termasuk foto, fasilitas, pemilik, dan ulasan

#### **2. Modul Booking Kunjungan:**

- Penyewa dapat melakukan pemesanan jadwal kunjungan ke kos.
- Notifikasi terkirim ke pemilik kos.

#### **3. Modul Pembayaran Sewa:**

- Penyewa dapat membayar sewa melalui sistem.
- Transaksi dicatat otomatis oleh sistem

#### **4. Modul Chat Penyewa–Pemilik**

- Fasilitas komunikasi dua arah untuk bertanya, negosiasi, atau diskusi kos.
- Mendukung pengiriman pesan real-time

#### **5. Modul Rating dan Ulasan:**

- Penyewa dapat memberikan rating dan komentar setelah masa sewa.
- Ulasan ditampilkan pada halaman detail kos.

#### **6. Modul Manajemen Kos untuk Pemilik:**

- Fitur pendaftaran kos baru.
- Pembaruan status kos (tersedia/terisi).
- Upload foto, fasilitas, dan informasi pendukung.

#### **7. Modul Verifikasi Kos (Admin):**

- Admin memverifikasi kos yang didaftarkan sebelum tampil ke publik.
- Menjamin integritas dan keaslian data kos

## **8. Dashboard Admin:**

- Menampilkan statistik kos, penyewa, status verifikasi, dan aktivitas sistem.
- Mendukung pengawasan operasional Roomify.

### **2.3 Target Pengguna**

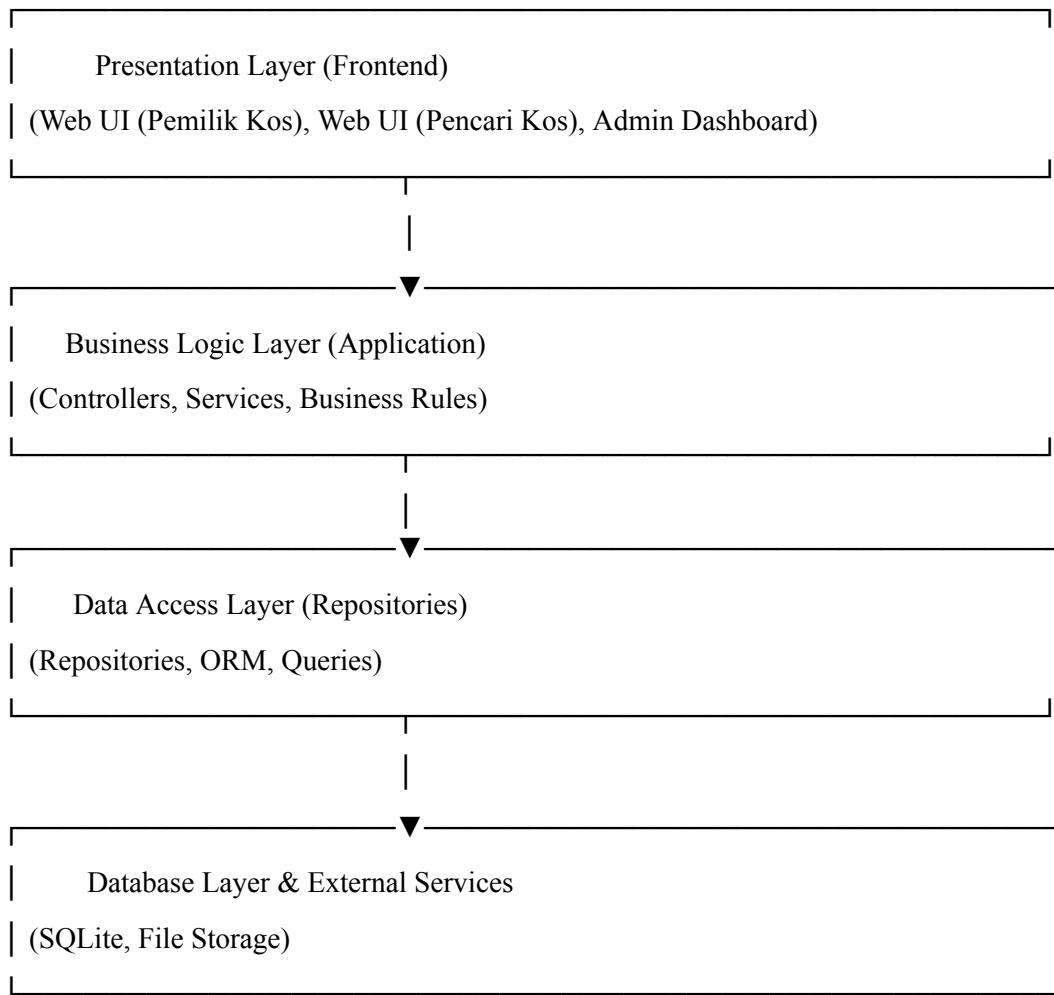
- Pencari Kos (Mahasiswa/Umum)
- Pemilik Kos
- Administrator Sistem

### **2.4 Lingkungan Operasional**

- **Sistem Operasi:** Windows, macOS, Linux, Android
  - **Browser:** Chrome, Firefox, Safari, Edge (versi terbaru)
  - **Server:** On-premise.
  - **Database:** SQLite
  - **Framework:** Bootstrap
-

### 3. ARSITEKTUR PERANGKAT LUNAK

#### 3.1 Arsitektur Umum



#### 3.2 Pola Arsitektur

- **Pattern**: Model-View-Controller (MVC)
- **Prinsip**: Separation of Concerns, DRY (Don't Repeat Yourself)
- **Desain**: Modular and layered architecture

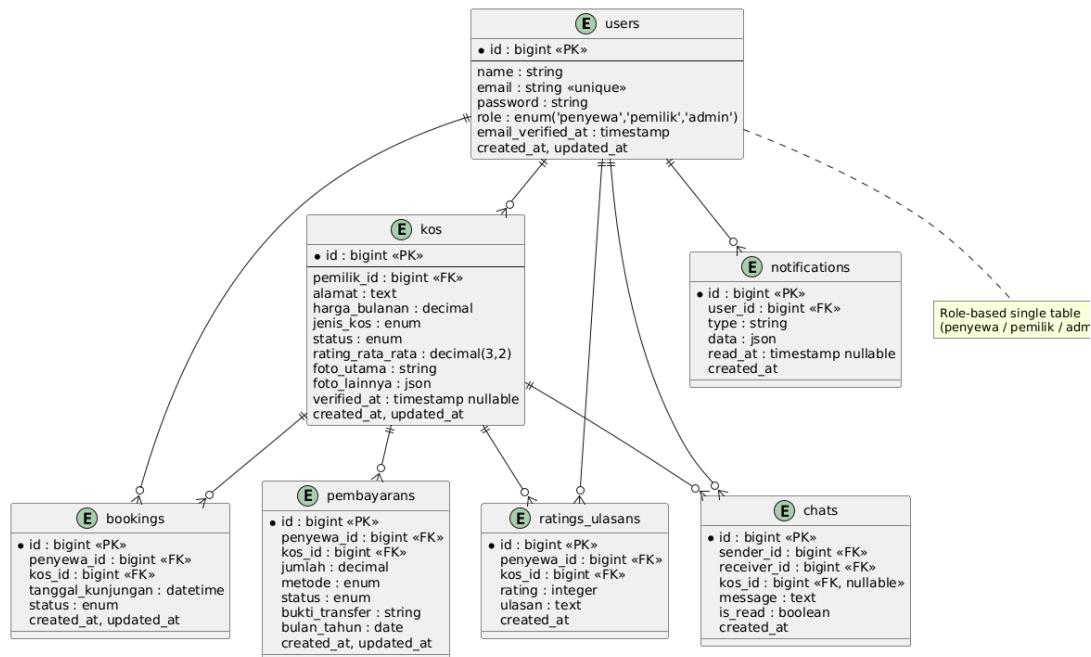
#### 3.3 Komponen Utama Sistem

Komponen	Deskripsi
<b>Frontend Application</b>	Interface web responsif untuk pengguna
<b>Backend Server</b>	business logic engine
<b>Database Management System</b>	Penyimpanan data terstruktur

<b>Authentication Service</b>	Manajemen user, role, dan permission
<b>File Storage Service</b>	Penyimpanan dokumen dan sertifikat
<b>Notification Service</b>	Email dan in-app notifications
<b>Reporting Engine</b>	Pembuatan laporan dan export data

## 4. DESAIN DATABASE

### 4.1 Entity Relationship Diagram (ERD)



Gambar 1. *Entity Relationship Diagram*

### 4.2 Deskripsi Entitas Utama

#### 4.2.1 Tabel Users (Pengguna)

**Tujuan:** Menyimpan data seluruh pengguna sistem yang terdaftar **Atribut Utama:**

- id (Primary Key) - name - email (Unique) - password - role (enum: 'penyewa', 'pemilik', 'admin') - email\_verified\_at - created\_at - updated\_at

#### *4.2.2 Tabel Kos (Data Kos)*

**Tujuan:** Menyimpan informasi detail mengenai kos yang didaftarkan oleh pemilik  
**Atribut Utama:** - id (Primary Key) - pemilik\_id (FK ke Users) - alamat - harga\_bulanan - jenis\_kos (enum: 'putra', 'putri', 'campur') - status (enum: 'tersedia', 'penuh', 'perbaikan') - rating\_rata\_rata - foto\_utama - foto\_lainnya (JSON) - verified\_at - created\_at - updated\_at

#### *4.2.3 Tabel Booking (Reservasi Kunjungan)*

**Tujuan:** Menyimpan data jadwal kunjungan survei yang diajukan penyewa **Atribut Utama:** - id (Primary Key) - penyewa\_id (FK ke Users) - kos\_id (FK ke Kos) - tanggal\_kunjungan - status (enum: 'pending', 'diterima', 'ditolak', 'selesai') - created\_at - updated\_at

#### *4.2.4 Tabel Pembayaran (Transaksi Sewa)*

**Tujuan:** Mencatat riwayat pembayaran sewa bulanan penyewa **Atribut Utama:** - id (Primary Key) - penyewa\_id (FK ke Users) - kos\_id (FK ke Kos) - jumlah - metode (enum: 'transfer\_bank', 'midtrans', 'va') - status (enum: 'pending', 'valid', 'ditolak') - bukti\_transfer - bulan\_tahun - created\_at - updated\_at

#### *4.2.5 Tabel Rating\_Ulasan (Ulasan)*

**Tujuan:** Menyimpan feedback dan penilaian bintang dari penyewa terhadap kos  
**Atribut Utama:** - id (Primary Key) - penyewa\_id (FK ke Users) - kos\_id (FK ke Kos) - rating (Integer) - ulasan (Text) - created\_at

#### *4.2.6 Tabel Chat (Pesan Langsung)*

**Tujuan:** Menyimpan riwayat percakapan antara penyewa dan pemilik **Atribut Utama:** - id (Primary Key) - sender\_id (FK ke Users) - receiver\_id (FK ke Users) - kos\_id (FK ke Kos, Nullable) - message - is\_read - created\_at

#### *4.2.7 Tabel Notification (Notifikasi Sistem)*

**Tujuan:** Menyimpan riwayat notifikasi in-app untuk pengguna **Atribut Utama:** - id (Primary Key) - user\_id (FK ke Users) - type - data (JSON) - read\_at - created\_at

### **4.3 Relasi Antar Tabel**

- Users memiliki banyak Kos (1:M).
- Users memiliki banyak Booking (1:M).
- Users memiliki banyak Pembayaran (1:M).
- Kos memiliki banyak Booking (1:M).

- Kos memiliki banyak Pembayaran (1:M).
- Kos memiliki banyak Rating\_Ulasan (1:M).
- Users (Sender/Receiver) memiliki banyak Chat (1:M).

#### 4.4 Normalisasi Database

- Semua tabel sudah dalam bentuk 3NF (Third Normal Form).
  - Tidak ada redundansi data yang signifikan.
  - Integritas referensial dijaga melalui foreign keys.
- 

### 5. DESAIN MODUL

#### 5.1 Modul Pencarian dan Filter Kos (Kos Search & Filter Module)

##### 5.1.1 Tujuan Modul

Menyediakan fitur pencarian kos secara cepat, akurat, dan relevan berdasarkan kriteria yang dimasukkan oleh pengguna, sehingga memudahkan penyewa menemukan kos yang sesuai.

##### 5.1.2 Fitur Utama

- Pencarian berdasarkan kata kunci alamat
- Filter harga bulanan
- Filter jenis kos (putra, putri, campur)
- Filter fasilitas (menggunakan JSON array pada kolom fasilitas)
- Filter status (aktif, tidak aktif, penuh)
- Preview kos (foto utama, harga, rating rata-rata)

##### 5.1.3 Alur Proses

1. Penyewa memasukkan kata kunci atau filter
2. Sistem mengeksekusi query ke tabel kos
3. Sistem menghitung rating rata-rata dari tabel rating\_ulasan (jika diperlukan)
4. Sistem menampilkan list kos sesuai kriteria

5. Penyewa memilih kos
6. Sistem menampilkan halaman detail

#### *5.1.4 Database Tables*

- kos
  - rating\_ulasan
- 

## **5.2 Modul Detail Kos & Ulasan (Kos Detail & Review Module)**

### *5.2.1 Tujuan Modul*

Menampilkan seluruh informasi kos termasuk detail pemilik, foto, serta ulasan.

### *5.2.2 Fitur Utama*

- Detail kos (alamat, harga, fasilitas, foto)
- Foto utama + foto tambahan
- Data pemilik (relasi kos.pemilik\_id → users.id)
- Menampilkan rating & ulasan
- Rekomendasi kos serupa (berdasarkan harga/jenis)

### *5.2.3 Alur Proses*

1. Rekomendasi kos serupa (berdasarkan harga/jenis)
2. Sistem mengambil data dari tabel kos.
3. Sistem mengambil semua ulasan terkait dari rating\_ulasan
4. Sistem menghitung rating rata-rata dari kolom rating\_ulasan.rating
5. Sistem menampilkan kos lain yang relevan (opsional)

### *5.2.4 Database Tables*

- kos
  - users (pemilik)
  - rating\_ulasan
-

### **5.3 Modul Booking Kunjungan (Visit Booking Module)**

#### *5.3.1 Tujuan Modul*

Mengelola permintaan kunjungan kos oleh penyewa.

#### *5.3.2 Fitur Utama*

- Mengajukan jadwal kunjungan
- Status booking (pending, diterima, ditolak, selesai)
- Notifikasi ke pemilik kos (via dashboard atau notifikasi internal)
- Riwayat booking

#### *5.3.3 Alur Proses*

1. Penyewa mengisi tanggal kunjungan
2. Sistem membuat entri di tabel booking
3. Pemilik kos menerima notifikasi
4. Pemilik menyetujui/menolak booking
5. Sistem mengirimkan status pada penyewa

#### *5.3.4 Database Tables*

- booking
  - users
  - kos
- 

### **5.4 Modul Pembayaran Sewa (Payment Module)**

#### *5.4.1 Tujuan Modul*

Mengelola pembayaran sewa kos dan mencatat riwayat transaksi

#### *5.4.2 Fitur Utama*

- Metode pembayaran (transfer\_bank, midtrans, va)

- Upload bukti transfer
- Status pembayaran (pending, valid, ditolak)
- Riwayat transaksi pengguna
- Laporan pembayaran untuk pemilik kos

#### *5.4.3 Alur Proses*

1. Penyewa memilih kos lalu klik “Bayar”
2. Sistem membuat entri pembayaran
3. Redirect ke halaman pembayaran
4. Gateway mengirim callback (optional)
5. Admin/pemilik memvalidasi bukti transfer
6. Status diperbarui menjadi valid/ditolak

#### *5.4.4 Database Tables*

- pembayaran
  - users
  - kos
- 

### **5.5 Modul Manajemen Kos untuk Pemilik (Kos Management Module)**

#### *5.5.1 Tujuan Modul*

Memberikan pemilik kemampuan mengelola kos yang mereka miliki

#### *5.5.2 Fitur Utama*

- Menambah kos baru
- Edit harga, foto, fasilitas
- Mengatur status (aktif, tidak aktif, penuh)
- Melihat ulasan dari penyewa
- Melihat pembayaran terkait kos miliknya

### *5.5.3 Alur Proses Pelatihan*

1. Pemilik membuka dashboard
2. Pemilik menambah atau memperbarui data kos
3. Sistem menyimpan ke tabel kos
4. Admin memverifikasi kos (kolom verified\_at)
5. Kos tampil di halaman publik setelah diverifikasi

### *5.5.4 Database Tables*

- kos
  - users (pemilik)
  - rating\_ulasan
  - pembayaran
- 

## **5.6 Modul Verifikasi Kos (Kos Verification Module)**

### *5.6.1 Tujuan Modul*

Menjamin kos yang ditampilkan sudah valid

### *5.6.2 Fitur Utama*

- Daftar kos dengan verified\_at = null
- Admin memverifikasi kos
- Penolakan/approval
- Notifikasi ke pemilik kos
- Audit melalui timestamp updated\_at

### *5.6.3 Alur Proses*

1. Pemilik menambahkan kos
2. Admin melihat kos dengan status belum diverifikasi
3. Admin memeriksa data

4. Admin menyetujui/menolak
5. Sistem mengisi kolom verified\_at

#### *5.6.4 Database Tables*

- Kos
- Users

### **5.7 Modul Chat**

#### *5.7.1 Tujuan Modul*

Menyediakan fitur komunikasi langsung antara penyewa dan pemilik kos

#### *5.7.2 Fitur Utama*

- Audit melalui timestamp updated\_at
- Riwayat percakapan per kos
- Notifikasi pesan masuk
- Menandai pesan sebagai sudah dibaca (is\_read)
- Chat hanya antara dua pengguna yang terkait dengan satu kos

#### *5.7.3 Alur Proses*

1. Penyewa membuka halaman chat
2. Sistem memuat pesan dari tabel chat
3. Penyewa mengirim pesan
4. Pesan dikirim ke real-time service
5. Pesan disimpan ke database
6. Pemilik merespons melalui dashboard

#### *5.7.4 Database Tables*

- chat
- users

- kos

## 5.8 Dashboard & Reporting Module

### 5.8.1 Tujuan Modul

Memberikan gambaran aktivitas sistem kepada admin, pemilik, dan penyewa

### 5.8.2 Fitur Utama

- Dashboard sesuai role user
- Statistik jumlah kos, booking, transaksi, dan ulasan
- Grafik data bulanan
- Riwayat transaksi penyewa
- Export CSV/PDF

### 5.8.3 Alur Proses

1. User login
2. Sistem menyesuaikan dashboard berdasarkan role.
3. Sistem mengambil data statistik dari tabel terkait.
4. Sistem menampilkan grafik & laporan.
5. User dapat mengekspor laporan

### 5.8.4 Database Tables

- pembayaran
  - kos
  - rating\_ulasan
  - booking
  - users
-

## 6. DESAIN USER INTERFACE

### 6.1 Prinsip Desain UI

- **Responsif:** Tata letak menyesuaikan layar Desktop dan Mobile (PWA).
- **User-Centric:** Navigasi intuitif untuk pengguna non-teknis (pemilik kos tua).
- **Konsisten:** Konsistensi visual dan interaksi di seluruh aplikasi
- **Minimalis:** Fokus pada konten (foto kos) dan aksi utama (*booking*).

### 6.2 Layout dan Navigasi

#### 6.2.1 Layout Umum

- **Header:** Top navigation
- **Footer:** Bottom navigation

#### 6.2.2 Navigation Structure

- **Top Navigation:** Logo, search bar, user profile, dashboard, logout button
- **Bottom Navigation:** Copyright, social media link, contact, navigation link, terms and conditions, about us, privacy policy.

### 6.3 Wireframe Halaman Utama

#### 6.3.1 Landing Page

Menampilkan overview sistem dengan widget berikut: - Welcome message - Superiority features - Tutorials - Review users - Call-to-action button

#### 6.3.2 Halaman Jelajahi Kos

- Search bar
- Filter options
- List kos

#### 6.3.3 Halaman Detail Kos

- Foto kos
- Detail kos (alamat, harga, rating, jenis kos)
- CTA button booking sekarang dan hubungi pemilik

- Review dan rating

## 6.4 Komponen UI Standar

- Buttons (Primary, Secondary, Danger)
- Input fields (Text, Number, Date, Select, Checkbox, Radio)
- Modals/Dialogs
- Alerts dan notifications
- Cards
- Tables
- Pagination (link())

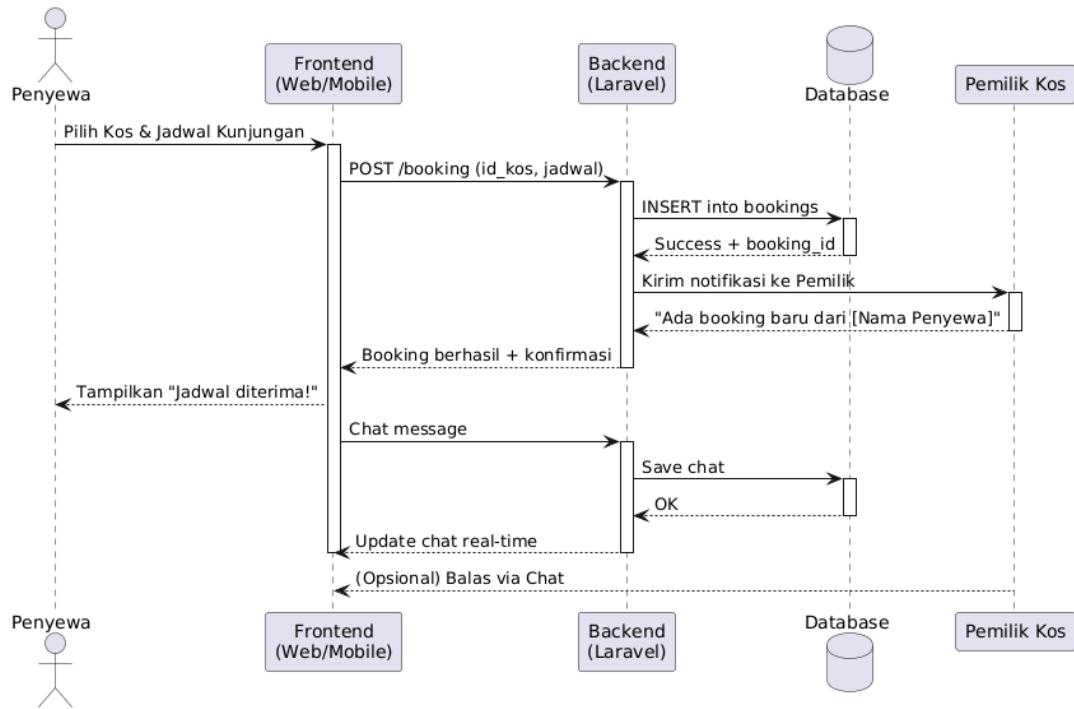
## 6.5 Color Scheme & Typography

- **Primary Color:** (#10a37f)
- **Secondary Color:** (#059669)
- **Success Color:** Green (#22c55e)
- **Warning Color:** Orange (#f59e0b)
- **Danger Color:** Red (#ef4444)
- **Font:** (Segoe UI, font-size 1.5rem, font-weight 700)

---

# 7. ALUR PROSES BISNIS

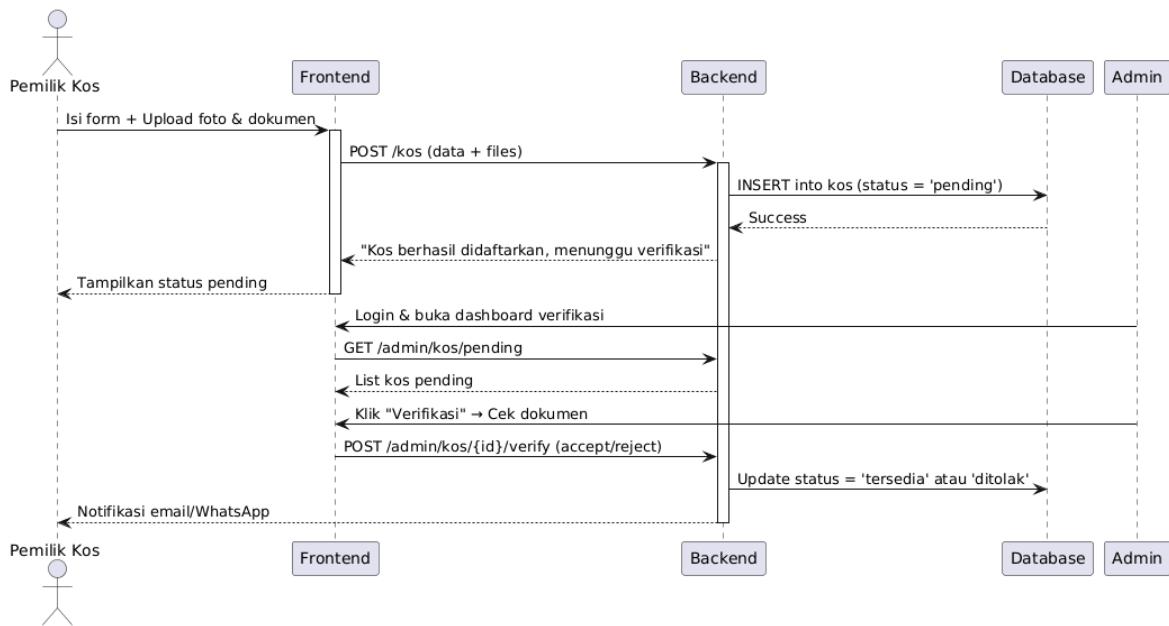
## 7.1 Alur Booking Kunjungan



Gambar 2. Sequence Diagram Alur Booking Kunjungan

1. Penyewa memilih jadwal
2. Sistem menyimpan data booking (status: pending)
3. Penyewa chat ke Pemilik
4. Pemilik login -> Klik Terima/Tolak
5. Sistem update status -> Notifikasi balik ke Penyewa

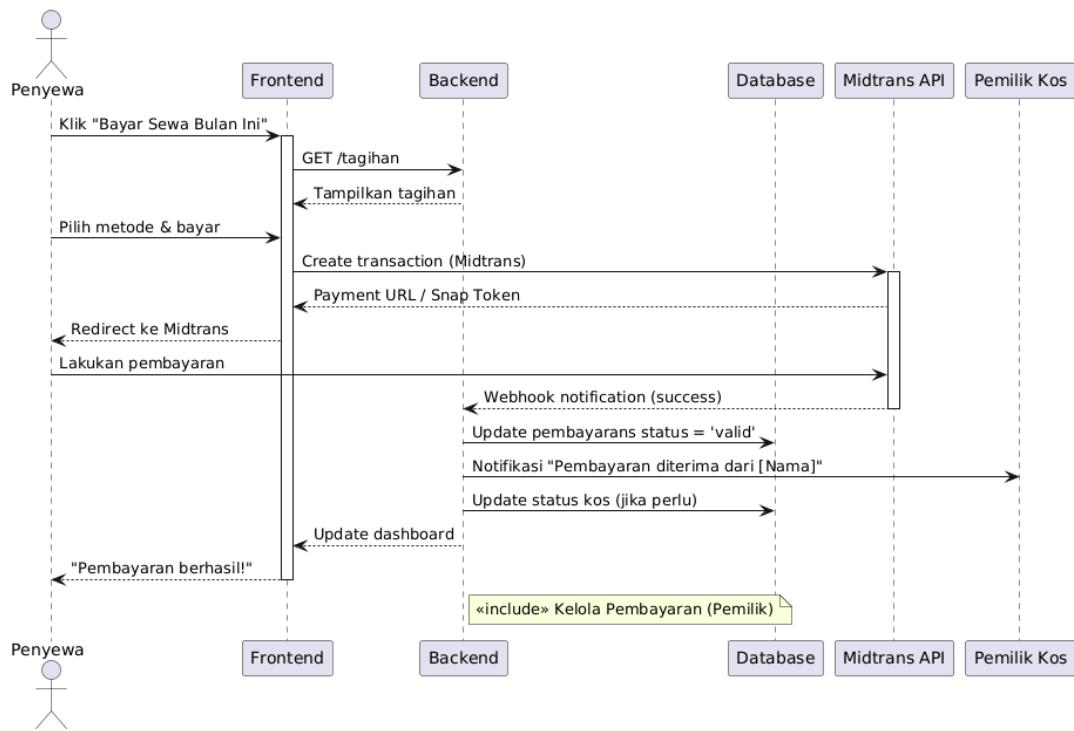
## 7.2 Alur Pendaftaran Kos Baru



Gambar 3. Sequence Diagram Alur Pendaftaran Kos Baru

1. Pemilik isi form & upload dokumen.
2. Sistem simpan data (status: pending verification).
3. Admin menerima notifikasi tugas verifikasi.
4. Admin cek kelengkapan -> Approve.
5. Status kos berubah menjadi 'Tersedia' & tampil di pencarian.

### 7.3 Alur Pembayaran Sewa



Gambar 4. Sequence Diagram Alur Pembayaran Sewa

1. Penyewa lihat tagihan -> Pilih Metode Bayar
2. Sistem request token ke Midtrans
3. Penyewa bayar di aplikasi pembayaran
4. Midtrans kirim Webhook (sukses)
5. Sistem update status tagihan -> Lunas

## 8. KEAMANAN & AUTENTIKASI

### 8.1 Authentication (Autentikasi)

- **Metode:** Username/Password + 2-Factor Authentication (optional)
- **Password Policy:** Minimal 8 karakter, kombinasi upper/lower case, number, special char
- **Session Management:** Session timeout setelah inaktif 30 menit
- **Password Reset:** Via email verification link
- **Account Lock:** Locked setelah 5 failed login attempts

### 8.2 Authorization (Otorisasi)

- **Role-Based Access Control (RBAC)**
  - Admin: Full access
  - Leadership: Approval & viewing
  - Staff: Limited access sesuai departemen
  - Viewer: Read-only access
- **Permission Levels:**
  - Create
  - Read
  - Update
  - Delete
  - Approve
  - Publish
  - Archive

### 8.3 Data Security

- **Encryption:** SSL/TLS untuk komunikasi data
- **Database:** Encrypted fields untuk sensitive data (passwords, identities)
- **File Storage:** Secure server dengan access control
- **Backup:** Regular automated backups
- **Data Retention:** Policy untuk deletion/archival

### 8.4 Audit & Logging

- **Audit Log:** Semua aktivitas user dicatat (action, timestamp, user, IP)
- **Monitoring:** Real-time monitoring untuk suspicious activities
- **Alerts:** Automated alerts untuk security incidents
- **Compliance:** Logs disimpan minimum 1 tahun

### 8.5 GDPR & Privacy

- **Data Protection:** Compliant dengan regulasi privasi data

- **Consent Management:** Explicit consent untuk data collection
  - **Right to Deletion:** User dapat request data deletion
  - **Data Portability:** User dapat export data mereka
- 

## 9. INTEGRASI SISTEM

### 9.1 Email Integration

- **Purpose:** Notification dan communication
- **Templates:** Email templates untuk berbagai notifications

### 9.2 File Storage Integration

- **Type:** Local server atau cloud storage (AWS S3, Google Cloud)
- **Usage:** SK documents, certificates, training materials
- **Security:** Private bucket dengan access control

### 9.3 Digital Signature Integration

- **Provider:** [E-signature provider]
- **Usage:** SK signing, certificate verification
- **Compliance:** Sesuai dengan regulasi digital signature

## 10. DEPLOYMENT & MAINTENANCE

### 10.1 Environment

- **Development:** Local environment untuk development
- **Staging:** Pre-production environment untuk testing
- **Production:** Live environment untuk users

### 10.2 Deployment Process

1. Code review
2. Automated testing
3. Build & compile
4. Deployment ke staging

5. QA testing
6. Approval
7. Deployment ke production
8. Monitoring

### 10.3 Maintenance & Support

- **Bug Fixes:** Priority-based fixing
  - **Feature Updates:** Scheduled releases
  - **Performance Monitoring:** Regular monitoring & optimization
  - **Support:** Help desk untuk user issues
- 

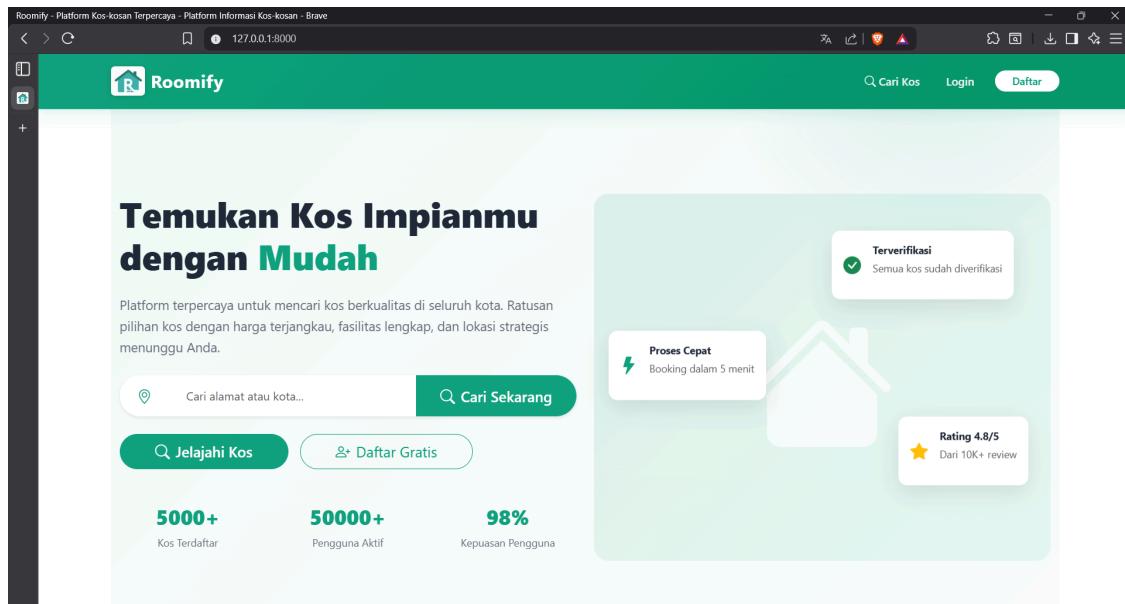
## 11. GLOSSARY

Term	Definition
<b>Kos</b>	Tempat tinggal sewa (boarding house) yang dikelola pemilik
<b>Penyewa</b>	Pengguna yang mencari dan memesan kos
<b>Pemilik Kos</b>	Pengelola/owner kos yang mendaftarkan properti dalam sistem
<b>Booking</b>	Pemesanan jadwal kunjungan survei kos sebelum menyewa
<b>Kunjungan (Visit)</b>	Aktivitas melihat kos secara langsung oleh calon penyewa
<b>Pembayaran Sewa</b>	Transaksi pembayaran sewa kos atau periodik
<b>Ulasan (Review)</b>	Feedback penyewa berupa rating dan komentar
<b>Rating</b>	Penilaian numerik terhadap kos
<b>Verifikasi Kos</b>	Proses pengecekan data kos oleh admin tampil ke publik
<b>Status Kos</b>	Kondisi kos (tersedia, penuh, dalam perbaikan)
<b>Fasilitas Kos</b>	Properti atau fitur yang disediakan dalam kos

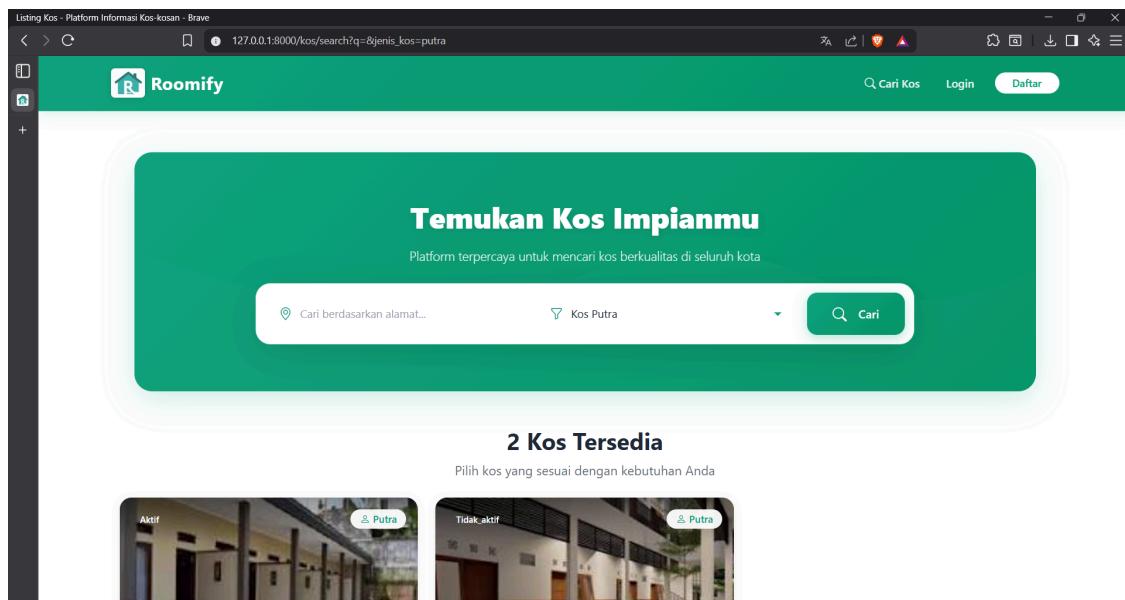
---

## 12. LAMPIRAN

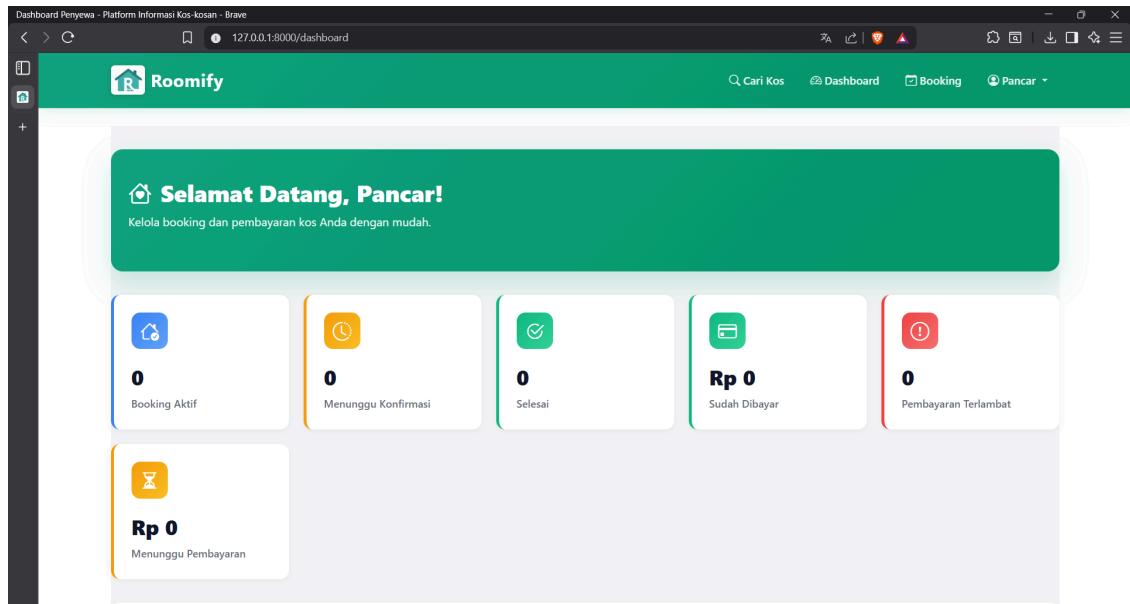
### 12.1 Wireframe & Mockups



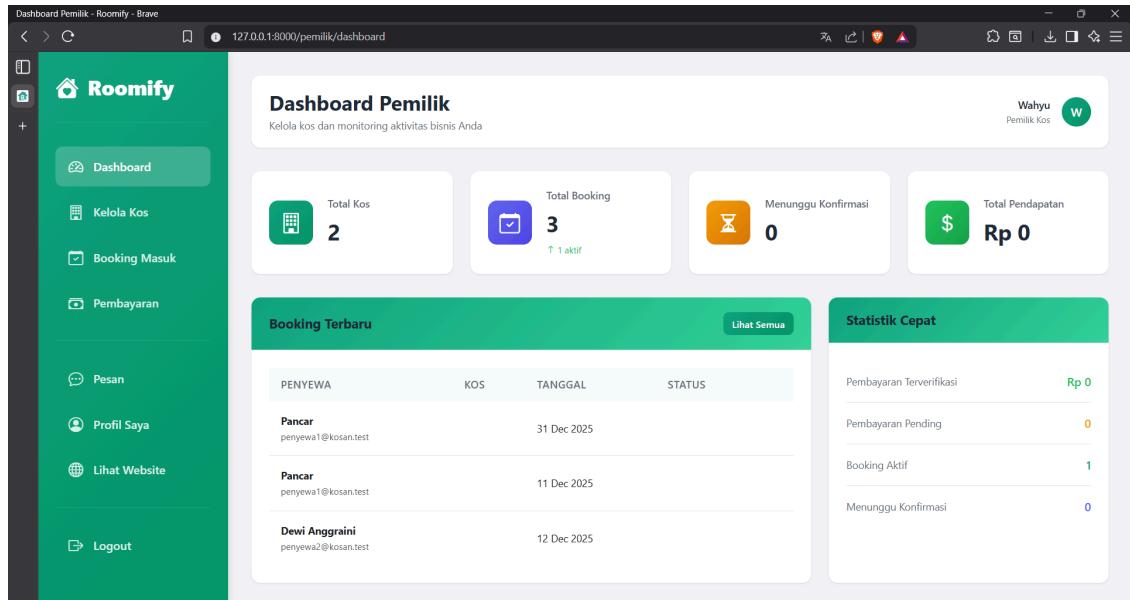
Gambar 5. Landing Page



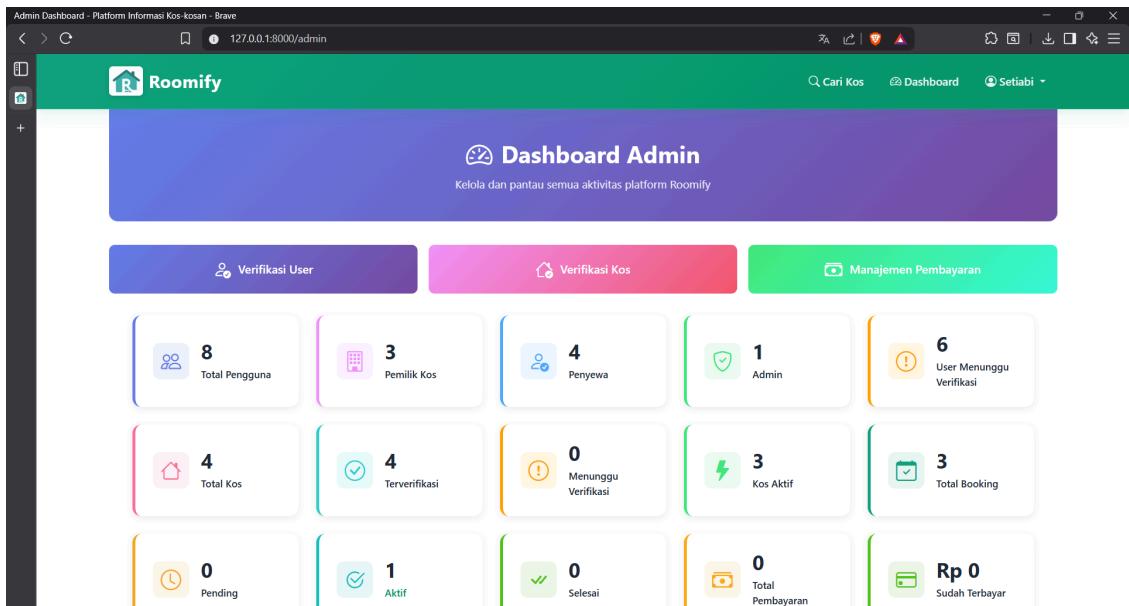
Gambar 6. List Kos Page



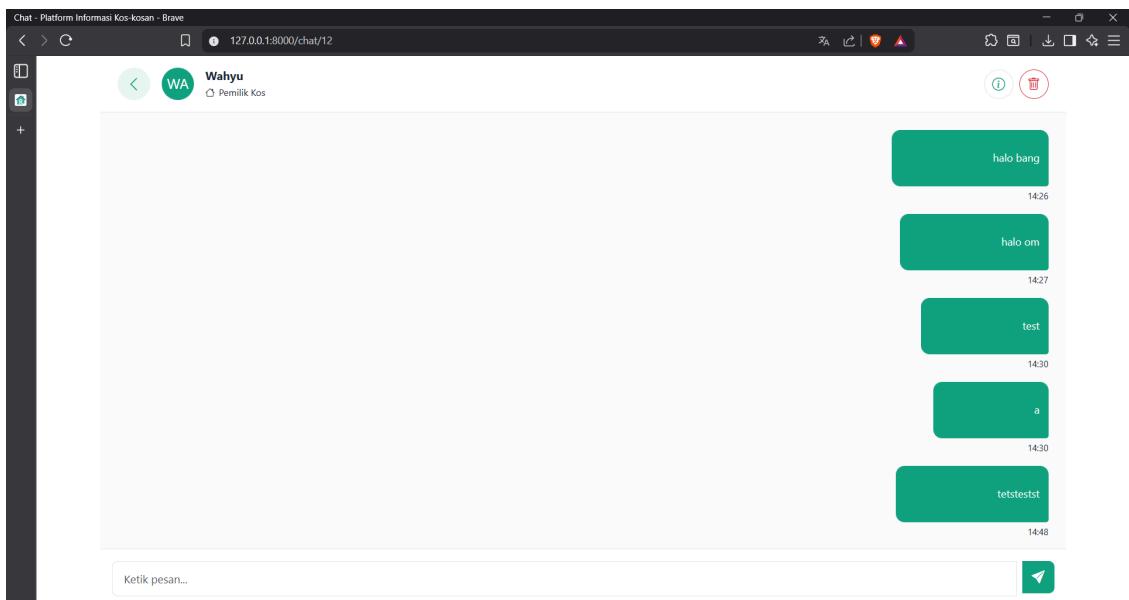
Gambar 7. Penyewa Kos Dashboard Page



Gambar 8. Pemilik Kos Dashboard Page



Gambar 9. Admin Dashboard Page



Gambar 10. Chat Page

## 12.2 Database Schema Script

```
PRAGMA foreign_keys = ON;  
CREATE TABLE users (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,
```

```
name TEXT NOT NULL,  
email TEXT NOT NULL UNIQUE,  
role TEXT NOT NULL DEFAULT 'penyewa' CHECK (role IN  
('penyewa','pemilik','admin')),  
is_verified INTEGER NOT NULL DEFAULT 0,  
verified_at DATETIME NULL,  
email_verified_at DATETIME NULL,  
password TEXT NOT NULL,  
remember_token TEXT NULL,  
created_at DATETIME NULL,  
updated_at DATETIME NULL  
);
```

```
CREATE TABLE password_reset_tokens (  
email TEXT PRIMARY KEY,  
token TEXT NOT NULL,  
created_at DATETIME NULL  
);
```

```
CREATE TABLE sessions (  
id TEXT PRIMARY KEY,  
user_id INTEGER NULL,  
ip_address TEXT NULL,  
user_agent TEXT NULL,  
payload TEXT NOT NULL,  
last_activity INTEGER NOT NULL,  
FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE SET NULL  
);
```

```
CREATE TABLE cache (  
key TEXT PRIMARY KEY,  
value TEXT NOT NULL,  
expiration INTEGER NOT NULL  
);
```

```
CREATE TABLE cache_locks (  
key TEXT PRIMARY KEY,  
owner TEXT NOT NULL,  
expiration INTEGER NOT NULL  
);
```

```
CREATE TABLE kos (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    pemilik_id INTEGER NOT NULL,
    alamat TEXT NULL,
    harga_bulanan NUMERIC NOT NULL DEFAULT 0.00,
    jenis_kos TEXT NOT NULL CHECK (jenis_kos IN ('putra','putri','campur')),
    status TEXT NOT NULL DEFAULT 'aktif' CHECK (status IN
    ('aktif','tidak_aktif','penuh')),
    rating_rata_rata NUMERIC NOT NULL DEFAULT 0.00,
    foto_utama TEXT NULL,
    foto_lainnya TEXT NULL CHECK (foto_lainnya IS NULL OR
    json_valid(foto_lainnya)),
    verified_at DATETIME NULL,
    created_at DATETIME NULL,
    updated_at DATETIME NULL,
    FOREIGN KEY (pemilik_id) REFERENCES users(id) ON DELETE CASCADE
);
CREATE INDEX kos_pemilik_id_index ON kos(pemilik_id);
CREATE INDEX kos_status_index ON kos(status);
```

```
CREATE TABLE booking (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    penyewa_id INTEGER NOT NULL,
    kos_id INTEGER NOT NULL,
    tanggal_kunjungan DATE NULL,
    status TEXT NOT NULL DEFAULT 'pending' CHECK (status IN
    ('pending','aktif','selesai','dibatalkan')),
    created_at DATETIME NULL,
    updated_at DATETIME NULL,
    FOREIGN KEY (penyewa_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (kos_id) REFERENCES kos(id) ON DELETE CASCADE
);
CREATE INDEX booking_penyewa_id_index ON booking(penyewa_id);
CREATE INDEX booking_kos_id_index ON booking(kos_id);
CREATE INDEX booking_status_index ON booking(status);
```

```
CREATE TABLE pembayaran (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    penyewa_id INTEGER NOT NULL,
```

```

kos_id INTEGER NOT NULL,
jumlah NUMERIC NOT NULL,
metode TEXT NOT NULL DEFAULT 'transfer' CHECK (metode IN
('transfer','tunai','e_wallet')),
status TEXT NOT NULL DEFAULT 'pending' CHECK (status IN
('pending','lunas','terlambat')),
bukti_transfer TEXT NULL,
bulan_tahun DATE NOT NULL,
created_at DATETIME NULL,
updated_at DATETIME NULL,
FOREIGN KEY (penyewa_id) REFERENCES users(id) ON DELETE CASCADE,
FOREIGN KEY (kos_id) REFERENCES kos(id) ON DELETE CASCADE
);
CREATE INDEX pembayaran_penyewa_id_index ON pembayaran(penyewa_id);
CREATE INDEX pembayaran_kos_id_index ON pembayaran(kos_id);
CREATE INDEX pembayaran_status_index ON pembayaran(status);

```

```

CREATE TABLE rating_ulasan (
id INTEGER PRIMARY KEY AUTOINCREMENT,
penyewa_id INTEGER NOT NULL,
kos_id INTEGER NOT NULL,
rating INTEGER NULL,
ulasan TEXT NULL,
created_at DATETIME NULL,
updated_at DATETIME NULL,
FOREIGN KEY (penyewa_id) REFERENCES users(id) ON DELETE CASCADE,
FOREIGN KEY (kos_id) REFERENCES kos(id) ON DELETE CASCADE,
UNIQUE (penyewa_id, kos_id)
);
CREATE INDEX rating_ulasan_penyewa_id_index ON rating_ulasan(penyewa_id);
CREATE INDEX rating_ulasan_kos_id_index ON rating_ulasan(kos_id);

```

```

CREATE TABLE notification (
id INTEGER PRIMARY KEY AUTOINCREMENT,
user_id INTEGER NOT NULL,
type TEXT NULL,
data TEXT NULL CHECK (data IS NULL OR json_valid(data)),
read_at DATETIME NULL,
created_at DATETIME NULL,
updated_at DATETIME NULL,

```

```

    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
);
CREATE INDEX notification_user_id_index ON notification(user_id);

CREATE TABLE chat (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    sender_id INTEGER NOT NULL,
    receiver_id INTEGER NOT NULL,
    message TEXT NOT NULL,
    is_read INTEGER NOT NULL DEFAULT 0,
    created_at DATETIME NULL,
    updated_at DATETIME NULL,
    deleted_at DATETIME NULL,
    deleted_by INTEGER NULL,
    FOREIGN KEY (sender_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (receiver_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (deleted_by) REFERENCES users(id) ON DELETE SET NULL
);
CREATE INDEX chat_sender_id_index ON chat(sender_id);
CREATE INDEX chat_receiver_id_index ON chat(receiver_id);
CREATE INDEX chat_deleted_by_index ON chat(deleted_by);

CREATE TABLE job (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    queue TEXT NOT NULL,
    payload TEXT NOT NULL,
    attempts INTEGER NOT NULL,
    reserved_at INTEGER NULL,
    available_at INTEGER NOT NULL,
    created_at INTEGER NOT NULL
);
CREATE INDEX job_queue_index ON job(queue);

CREATE TABLE job_batches (
    id TEXT PRIMARY KEY,
    name TEXT NOT NULL,
    total_jobs INTEGER NOT NULL,
    pending_jobs INTEGER NOT NULL,
    failed_jobs INTEGER NOT NULL,
    failed_job_ids TEXT NOT NULL,

```

```
options TEXT NULL,  
cancelled_at INTEGER NULL,  
created_at INTEGER NOT NULL,  
finished_at INTEGER NULL  
);
```

```
CREATE TABLE failed_jobs (  
id INTEGER PRIMARY KEY AUTOINCREMENT,  
uuid TEXT NOT NULL UNIQUE,  
connection TEXT NOT NULL,  
queue TEXT NOT NULL,  
payload TEXT NOT NULL,  
exception TEXT NOT NULL,  
failed_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP  
);
```

### 12.3 Testing Plan

[Sertakan test cases dan testing strategy di sini]

### 12.4 Approval & Sign-off

Role	Name	Signature	Date
Project Manager	Pancar Wahyu		12/12/2025
Business Analyst	Satria Megantara		12/12/2025
QA Lead	Muhammad Umar Faiz		12/12/2025
Technical Lead	Muhammad Umar Faiz		12/12/2025

### Document Control

Attribute	Value
<b>Document Version</b>	1.0
<b>Last Updated</b>	12/12/2025
<b>Status</b>	Pending
<b>Owner</b>	Muhammad Umar Faiz
<b>Author</b>	Muhammad Umar Faiz
<b>Distribution</b>	Confidential - Internal Use Only