



Template SRS RPL SI 2023 2024



Scan to open on Studeersnel

# **Software Requirement Specification**

## **for**

## **InKost**

Dipersiapkan Oleh

701230026 Aldi Darmawan

701230028 Surah Annisa

701230058 Deni Khairul Anam

Program Studi Sistem Informasi Fakultas

Sains dan Teknologi

Universitas Islam Negeri Sultan Thaha Saifuddin Jambi 2025

## DAFTAR ISI

<b>DAFTAR ISI .....</b>	<b>i</b>
<b>DAFTAR GAMBAR .....</b>	<b>iii</b>
<b>DAFTAR TABEL.....</b>	<b>iv</b>
<b>RIWAYAT REVISI.....</b>	<b>v</b>
1 Pendahuluan.....	1
1.1 Tujuan.....	1
1.2 Definisi dan Istilah .....	2
1.3 Referensi.....	2
2 Deskripsi Produk atau Aplikasi .....	3
2.1 Deskripsi umum produk atau aplikasi .....	3
2.2 Kategori Pengguna Produk atau Aplikasi .....	4
2.3 Fitur Produk atau Aplikasi .....	4
2.4 Karakteristik Pengguna Produk atau Aplikasi.....	5
2.5 Batasan Produk atau Aplikasi.....	5
2.6 Dokumentasi Pengguna.....	6
3 Kebutuhan Eksternal Produk atau Aplikasi.....	6
3.1 Karakteristik Tampilan Pengguna .....	6
3.2 Karakteristik <i>Hardware</i> .....	6
3.3 Karakteristik <i>Software</i> .....	7
3.4 Karateristik Komunikasi.....	7
4 Kebutuhan Fungsional Produk atau Aplikasi .....	8
4.1 Kebutuhan fungsional.....	8
4.2 <i>Use case diagram</i> .....	9
4.3 Use case Deskripsi.....	9
4.3.1 <i>Use case</i> deskripsi fitur registrasi .....	10
4.3.2 <i>Use case</i> deskripsi fitur mencari kamar .....	10
4.3.3 <i>Use case</i> deskripsi fitur memesan kamar .....	11
4.3.4 <i>Use case</i> deskripsi fitur mengelola data kamar .....	11
4.4 <i>Activity Diagram</i> .....	12
4.4.1 <i>Activity Diagram</i> fitur <i>login</i> dan registrasi.....	13
4.4.2 <i>Activity Diagram</i> fitur pemesanan kamar.....	14

4.4.3	<i>Activity Diagram</i> fitur Kelola data kamar .....	15
4.5	Kebutuhan <i>database</i> .....	16
4.5.1	<i>Entity Relationship Diagram</i> .....	16
4.5.2	<i>Class Diagram</i> .....	18
4.5.2.1	<i>Class Diagram</i> Fitur penyewaan.....	19
4.5.2.2	<i>Class Diagram</i> Fitur pemilik kost.....	20
4.6	<i>Sequence Diagram</i> .....	21
4.6.1	<i>Sequence Diagram</i> Fitur Penyewa.....	21
4.6.2	<i>Sequence Diagram</i> Fitur Pemilik kost .....	22
5	Kebutuhan <i>Non-Fungsional</i> Produk atau Aplikasi .....	22
5.1	Kebutuhan Produk.....	22
5.1.1	<i>Usability</i> .....	22
5.1.2	<i>Efficiency</i> .....	23
5.1.2.1	<i>Performance</i> .....	23
5.1.2.2	<i>Space</i> .....	23
5.1.2.3	<i>Security</i> .....	23

## **DAFTAR GAMBAR**

Gambar 4.1 <i>Use case Diagram</i> .....	9
Gambar 4.2 <i>Activity Diagram</i> .....	12
Gambar 4.3 <i>Activity Diagram</i> fitur <i>login</i> dan registrasi .....	13
Gambar 4.4 <i>Activity Diagram</i> fitur pemesanan kamar .....	14
Gambar 4.5 <i>Activity Diagram</i> fitur fitur kelola data kamar .....	15
Gambar 4.6 <i>Entity Relationship Diagram</i> (ERD) .....	16
Gambar 4.7 <i>Class Diagram</i> .....	18
Gambar 4.8 <i>Class Diagram</i> Fitur penyewa.....	19
Gambar 4.9 <i>Class Diagram</i> Fitur Pemilik Kost .....	20
Gambar 4.10 <i>Sequence Diagram</i> Fitur Penyewa .....	21
Gambar 4.10 <i>Sequence Diagram</i> Fitur Pemilik Kost.....	22

## **DAFTAR TABEL**

Tabel 2.1 Rincian fitur.....	4
Tabel 2.2 Karakteristik pengguna.....	5
Tabel 4.1 <i>Use case</i> deskripsi fitur registrasi.....	10
Tabel 4.2 <i>Use case</i> deskripsi fitur mencari kamar .....	10
Tabel 4.3 <i>Use case</i> deskripsi fitur memesan kamar .....	11
Tabel 4.4 <i>Use case</i> deskripsi fitur mengelola data kamar .....	11

## **RIWAYAT REVISI**

Keterangan Revisi	Tanggal	Alasan Perubahan	Penanggungjawab Revisi
Revisi 1	Isikan tanggal revisi	Isikan alasan perubahan	Isikan siapa penanggungjawab revisi
Dan seterusnya			

## **1 Pendahuluan**

Dokumen ini berisi perancangan sistem informasi pengelolaan rumah kost berbasis web yang bertujuan mempermudah mahasiswa baru dalam mencari kost atau kontrakan di sekitar kampus tanpa harus survei langsung yang memakan waktu dan biaya. Sistem ini dilengkapi dengan fitur utama seperti *Home* (penyewaan, kamar, penghuni, pembayaran), *Register* (data penyewa), *Login*, Kamar (informasi pemesanan, harga, fasilitas), Penyewaan (data sewa), Penghuni (identitas dan periode sewa), serta Pembayaran (tanggal, metode, dan data penyewa). Alur sistem dimulai dari *Home*, kemudian *Register*, lalu *Login*, dilanjutkan ke Daftar Kamar, setelah itu Sewa Kamar, berikutnya Pembayaran, dan berakhir pada *End*. Pengembangan sistem menggunakan model SDLC *Waterfall* karena lebih terdokumentasi dengan baik, kebutuhan jelas, dan tidak mudah berubah.

### **1.1 Tujuan**

Dokumen *Software Requirement Specification* (SRS) disusun untuk mempermudah mahasiswa baru dalam mencari kost atau kontrakan di sekitar kampus tanpa harus survei langsung yang memakan waktu dan biaya. Tujuan penulisan dokumen ini dibedakan sesuai penggunaanya sebagai berikut:

#### **1. Tujuan bagi Klien**

- a. Menjamin sistem yang dibangun sesuai tujuan awal, yaitu mempermudah mahasiswa untuk mencari kost dan menyewa kost secara *online*.
- b. Menjadi dokumen resmi yang memaparkan fungsi dan sistem secara detail.

#### **2. Tujuan bagi *Developer***

- a. Memastikan setiap fitur berjalan sempurna.
- b. Menjadi pedoman dalam perancangan, pembangunan, pengujian sistem, dan pengembangan.



3. Tujuan bagi *Owner*
  - a. Menjadi penilaian apakah sistem yang dibangun sudah sesuai kebutuhan pengguna dan kebutuhan bisnis.
  - b. Sebagai dasar bagi pihak yang menentukan fitur dan ruang lingkup sistem yang akan dibangun.

## 1.2 Definisi dan Istilah

1. *Software Requirement Specification* (SRS) = Dokumen kebutuhan perangkat lunak.
2. Web Kost = Aplikasi berbasis web untuk pengelolaan data rumah kost.
3. *Admin* = Pengguna yang mengelola data kamar, penghuni, dan pembayaran.
4. *User / Penyewa* = Pengguna yang mencari dan memesan kamar kost.
5. *Database* = Tempat penyimpanan data kamar, penghuni, dan transaksi.
6. PHP (*Hypertext Preprocessor*) = Bahasa pemrograman untuk aplikasi web.
7. MySQL = Sistem manajemen basis data relasional yang digunakan untuk menyimpan data.
8. UML (*Unified Modeling Language*) = Bahasa standar pemodelan untuk analisis dan desain sistem.
9. ERD (*Entity Relationship Diagram*) = Adalah diagram visual yang digunakan untuk merancang dan memodelkan basis data, menjelaskan bagaimana data disimpan dan dihubungkan antar tabel.

## 1.3 Referensi

- Triadi, A., Sepriano, & Kusminar, M. (2024). *Perancangan Sistem Inormasi Pengelolaan Rumah Kost Putri Hana Berbasis Website dengan Menggunakan Framework CodeIgniter*. Jurnal Ilmu Komputer dan Sistem Infomasi (JIKSI), 10(11), 35–43.
- Ramadhana, I., Irawati, & Sugiarti. (2024). Rancang Bangun Sistem Informasi Penyewaan Kamar Kost Berbasis Android pada Kost Alfira di Makassar. *Literatur Informatika & Komputer (Linier)*, 1(3), 223–234.

Azzahra, F. F., Sekamayang, B. R., & Habibi, R. (2025). Integrasi Google OAuth dan Payment Gateway Midtrans dalam KosConnect. *Jurnal Processor*, 20(1), 75–84.

Wijayanto, A., & Wahab, A. (2020). Analisa dan Perancangan Penyewaan Kost Berbasis Web (Studi Kasus: AW Kost) dengan Metode Waterfall. *Jurnal Sistem Informasi dan E-Bisnis (JUSIBI)*, 2(1), 271–280.

## 2 Deskripsi Produk atau Aplikasi

### 2.1 Deskripsi umum produk atau aplikasi

Sistem informasi pengelolaan rumah kost Inkost berbasis *website* merupakan aplikasi yang dirancang bertujuan untuk mempermudah masyarakat umum ataupun mahasiswa untuk mencari, memesan, ataupun mengelola informasi kost di sekitar kampus tanpa harus melakukan survei langsung ke tempat yang mungkin dapat mempersulit dan membutuhkan waktu, biaya dan tenaga. Maka dirancanglah aplikasi berbasis *website* ini untuk mempermudah semua permasalahan yang telah di sebutkan sebelumnya, dan aplikasi ini menyediakan berbagai fitur utama seperti:

#### 1) *Home*

Yang menyediakan informasi umum tentang rumah kost, kamar yang tersedia, fasilitas, harga sewa.

#### 2) *Register* dan *login*

Digunakan untuk mendaftarkan akun penyewa baru agar dapat memiliki akun dan dapat mengakses *dashboard* layanan penyewaan.

#### 3) Kamar

Menyediakan informasi kamar seperti harga kamar, pemesanan, fasilitas dan ketersediaan kamar.

#### 4) Penyewaan

Menyediakan info data penyewaan, lama penyewaan, dan alur pemesanan kost.

#### 5) Penghuni

Menyimpan info data identitas penghuni dan lama sewa yang berjalan.

## 6) Pembayaran

Menyimpan riwayat detail pembayaran dan data penyewa yang melakukan transaksi.

Dengan adanya sistem ini, penyewaan kost dapat dilakukan lebih mudah dan praktis. Dan bagi pemilik kost, aplikasi ini dapat membantu untuk mengelola data penyewa yang terstruktur beserta dengan pembayarannya. Sedangkan bagi penyewa aplikasi ini dapat mempermudah untuk mencari kost secara *online* tanpa membuang tenaga dan waktu.

## 2.2 Kategori Pengguna Produk atau Aplikasi

- 1) *Admin*
- 2) Pemilik kost
- 3) Pengguna (penyewa kost)

## 2.3 Fitur Produk atau Aplikasi

Tabel 2.1 Rincian Fitur

Katagori pengguna	ID fitur	fitur
<i>Admin</i>	Wk-001	Melihat aktivitas terbaru, mengelola data <i>user/pemilik kost</i> , mengelola pembayaran
Pemilik Kost	Wk-002	Menerima data pesanan, mengelola data kamar, mengelola data pembayaran, konfirmasi pembayaran
Penyewa /Pengguna	Wk-003	Mencari kamar, memesan kamar. Membayar sewa

## 2.4 Karakteristik Pengguna Produk atau Aplikasi

Tabel 2.2 karakteristik pengguna

Katagori pengguna	tugas	Hak akses aplikasi	Kemampuan yang harus dimiliki
Admin	-mengelola data pemilik kost dan <i>user</i> -memantau laporan -memantau pembayaran	- Akses penuh ke seluruh modul aplikasi - Mengelola akun pemilik kost dan pengguna - Melihat laporan	- Memahami manajemen data - Menguasai penggunaan aplikasi berbasis web - Teliti dalam verifikasi data
Pemilik Kost	- Menginput dan mengelola data kamar - Mengelola data pembayaran - Mengkonfirmasi pesanan/pembayaran	- Menambah, mengedit, menghapus data kamar - Melihat daftar pemesanan kamar - Mengkonfirmasi pembayaran dari penyewa	- Mampu menggunakan aplikasi web - Memahami manajemen kost sederhana - Dapat memverifikasi transaksi penyewa
Pengguna/Penyewa	- Mendaftar dan <i>login</i> - Melihat daftar kamar kost - Memesan kamar - Melakukan pembayaran sewa	- Registrasi akun - <i>Login</i> ke sistem - Melihat detail kamar - Melakukan pemesanan dan pembayaran	- Dapat menggunakan aplikasi web dasar - Memahami cara transaksi <i>online</i> - Dapat melakukan transaksi dalam bentuk <i>e-money</i> pembayaran

## 2.5 Batasan Produk atau Aplikasi

1. Aplikasi hanya berfokus pada dua fitur inti, yaitu pencarian & pemesanan kost serta manajemen data kost oleh pemilik.
2. Aplikasi berbasis *website* menggunakan *framework Laravel* dengan konsep MVC.
3. *Database* yang digunakan adalah MySQL.

## **2.6 Dokumentasi Pengguna**

Kami hanya akan menyediakan dokumentasi berupa panduan penggunaan sederhana dalam bentuk PDF dan video *YouTube* yang menjelaskan cara memakai situs, serta bantuan *online /FAQ* untuk menjawab masalah umum, ditambah panduan singkat di dalam aplikasi seperti *tooltip* atau *pop-up* agar pengguna lebih mudah memahami fitur.

## **3 Kebutuhan Eksternal Produk atau Aplikasi**

### **3.1 Karakteristik Tampilan Pengguna**

- a) Aplikasi ini berbasis *website* dan hanya difokuskan untuk diakses melalui *Google Chrome*, bukan dalam bentuk aplikasi *mobile* atau *desktop* khusus.
- b) Tampilan *website* dirancang responsif, dengan ukuran layar optimal pada laptop/PC dengan resolusi  $1366 \times 768$  px hingga  $1920 \times 1080$  px.
- c) Navigasi utama ditempatkan pada bagian atas (*header*) halaman web. Tombol-tombol penting seperti *Login*, Daftar, atau Cari Kost diletakkan di posisi strategis (pojok kanan atas atau tengah halaman).
- d) Tombol *Login* dibuat dengan warna yang berbeda dan mudah dikenali, misalnya putih dengan teks hitam, serta memiliki efek *hover* berupa perubahan warna menjadi abu-abu muda.
- e) Desain keseluruhan menggunakan tema warna putih dan hitam untuk kesan sederhana, profesional, dan mudah dibaca.

### **3.2 Karakteristik *Hardware***

Perangkat keras yang mendukung pengoperasian web inkost antara lain:

#### **1. Perangkat *Mobile* (Android/iOS)**

- a. Minimal Android 8.0 (Oreo) atau iOS 12
- b. RAM minimal 2 GB
- c. Penyimpanan kosong minimal 200 MB
- d. Resolusi layar minimal 720p

#### **2. Laptop/Komputer**

- a. Sistem operasi minimal *Windows* 10, *Linux Ubuntu* 20.04, atau *macOS* 10.15
- b. RAM minimal 4 GB
- c. Prosesor setara Intel i3/AMD Ryzen 3
- d. Koneksi internet minimal 5 Mbps

### 3. Perangkat Tambahan

- a. Tidak memerlukan perangkat tambahan khusus, cukup perangkat standar yang dapat terkoneksi dengan internet

## 3.3 Karakteristik Software

Perangkat lunak yang mendukung pengoperasian web inkost antara lain:

### 1. *Browser* yang Didukung

- a. *Google Chrome* versi 80 ke atas
- b. *Mozilla Firefox* versi 75 ke atas
- c. *Microsoft Edge* versi terbaru
- d. *Safari* 13 ke atas (untuk *iOS/macOS*)

### 2. *Software Pendukung*

- a. *PDF Reader* (misalnya *Adobe Acrobat Reader* atau bawaan *browser*) untuk membuka laporan dalam *form at PDF*
- b. *Office* (*Microsoft Office* / *LibreOffice* / *WPS Office*) untuk membuka atau mengolah laporan dalam format *DOCX/XLSX*

## 3.4 Karakteristik Komunikasi

Karakteristik komunikasi pada pengoperasian aplikasi Inkost meliputi penggunaan enkripsi *password* untuk menjaga keamanan data *user*, penerapan protokol *HTTPS* dalam pertukaran informasi, sinkronisasi data secara *real-time* melalui API, dukungan *multi* bahasa (Bahasa Indonesia dan Inggris), serta integrasi API *Google Maps* untuk menampilkan lokasi kost.

## **4. Kebutuhan Fungsional Produk atau Aplikasi**

### **4.1 Kebutuhan fungsional**

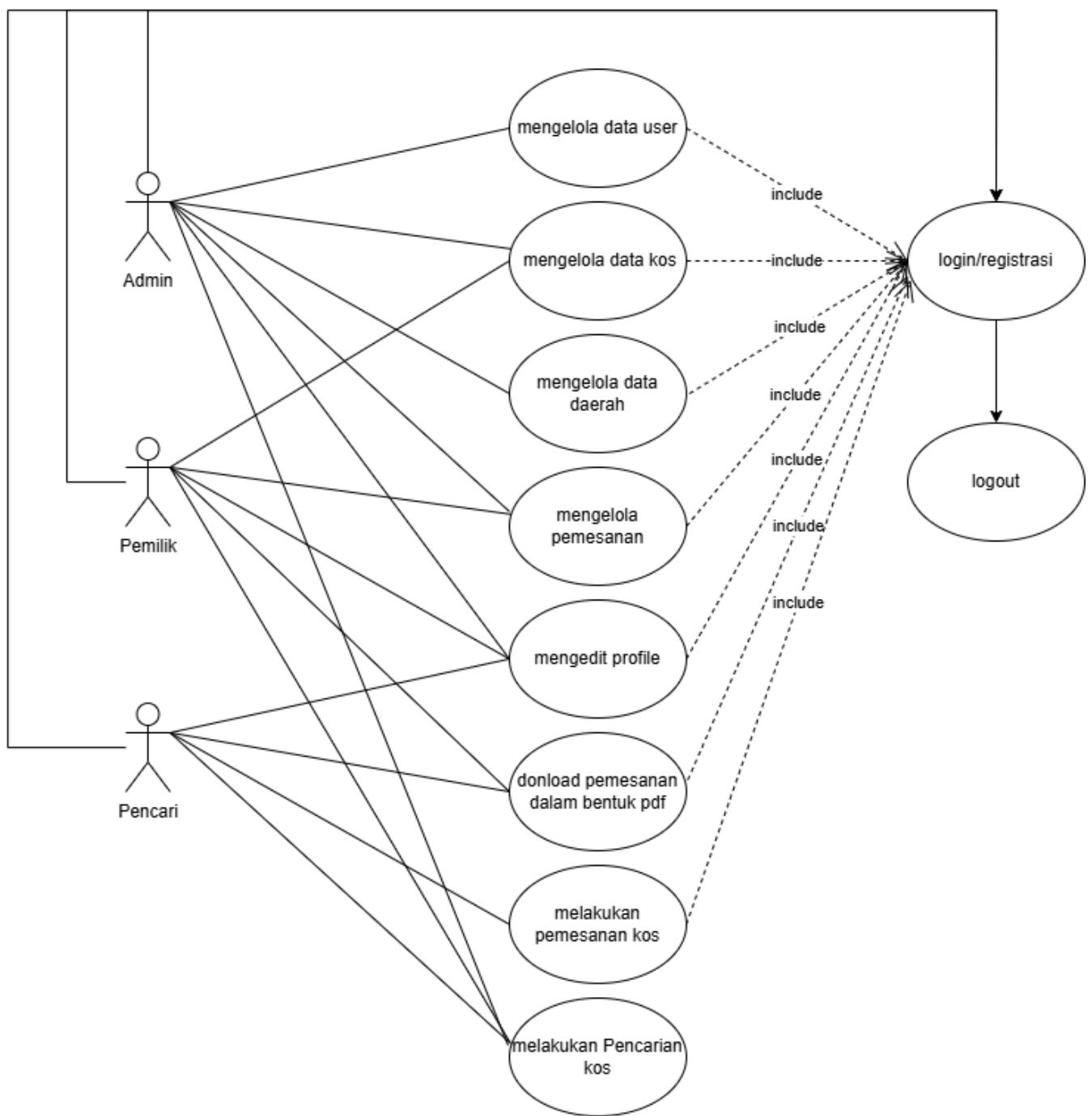
Adapun kebutuhan *fungsional* atau layanan utama yang harus disediakan agar sistem dapat mencapai sesuai tujuan diantaranya yaitu:

- 1) Autentikasi pengguna
  - a) Sistem menyediakan fitur registrasi untuk pengguna baru
  - b) Fitur *login* dan *logout* bagi pengguna yang sudah memiliki akun
  - c) Dan sistem mengelola akses berdasarkan *role* (*Admin*, pemilik kost, *user*/penyewa).
- 2) Profil pengguna
  - a) Pengguna dapat melihat profil, dan mengedit data profil
  - b) Pengguna dapat mengganti *password* dengan *form* khusus.
- 3) *Upload* dan unduh
  - a) *User* dapat mengupload sesuai format yang telah menjadi persyaratan
  - b) *pemilik* dapat mengunduh apa yang telah di *upload* oleh *user*
- 4) Verifikasi (khusus *Admin*)
  - a) *Admin* dapat memverifikasi data yang di *upload* oleh *user*
  - b) *Admin* dapat mengubah status *user*
  - c) Status otomatis tampil di sisi *user* dan *Admin*
- 5) Pencarian

Sistem menyediakan pencarian dengan berdasarkan kata kunci
- 6) *Dashboard*
  - a) *User* dapat melihat rekomendasi kost yang terbaik
  - b) *Admin* dapat melihat ringkasan dan data seluruh *user*
  - c) Pemilik kost dapat melihat ringkasan data penghuni kost dan memverifikasi penyewaan yang dilakukan pencari.

## 4.2 Use case diagram

Gambar 4.1 Use case Diagram



## 4.3 Use case Deskripsi

Berdasarkan dari gambar di atas yang menampilkan *diagram use case*. Gambaran di atas menggambarkan rincian interaksi antara sistem dan “aktor” dan

berikut ini adalah rincian deskripsi dari *Use case diagram*.



### 4.3.1 Use case deskripsi fitur registrasi

Table 4.1 Use case deskripsi fitur registrasi

Nama	Registrasi
<b>aktor</b>	User (Penyewa / Pemilik)
<b>deskripsi</b>	Aktor membuat akun baru dengan memasukkan data diri dan sistem menyimpan ke database
<b>precondition</b>	Aktor sudah membuka halaman registrasi
<b>postconditions</b>	Akun berhasil dibuat dan dapat digunakan untuk login
<b>Exception</b>	Email sudah terdaftar / password tidak valid
<b>Skenario normal</b>	
1. Aktor mengeklik menu registrasi	2. Sistem menampilkan menu registrasi
3. Aktor memasukkan <i>email</i> , nama lengkap, data lengkap, dan <i>password</i> dan menonfirmasi <i>password</i>	
4. Aktor mengeklik daftar	5. Sistem memverifikasi data yang telah di <i>input</i>
	6. Data berhasil dimasukkan ke <i>database</i>
	7. Sistem menghasilkan <i>notifikasi</i> berhasil
<b>Skenario Alternatif</b>	
	1. Data gagal ditambahkan ke <i>database</i>
	2. Sistem menampilkan <i>notifikasi</i> gagal
3. Aktor menganalisis kembali kesalahan yang di <i>input</i>	
4. Mengulang <i>input</i> yang baru	
5. Aktor mengklik <i>login</i>	6. Masuk ke dalam <i>dashboard user</i> .

### 4.3.2 Use case deskripsi fitur mencari kamar

Table 4.2 Use case deskripsi fitur mencari kamar

Nama	Mencari kamar
Aktor	<i>user/penyewa</i>
deskripsi	<i>User</i> mencari kost terdekat ataupun yang sesuai dengan keinginannya
Skenario normal	
1. Aktor mengklik menu pencarian pada <i>dashboard user</i>	
2. Aktor mengisi kolom pencarian	

3. Aktor mengeklik tombol yang terdapat <i>icon</i> pencarian	4. sistem akan menampilkan semua data kost yang di cari yang berisi lokasi, foto kost, dan harga kost.
---	--

### 4.3.3 Use case deskripsi fitur memesan kamar

Table 4.3 Use case deskripsi fitur memesan kamar

Nama	Mencari kamar
aktor	penyewa/user
deskripsi	Aktor harus menginput kos yang telah di pilih untuk di masukkan ke database
Skenario normal	
1. Aktor mengeklik data kamar	2. Sistem menampilkan data kamar
3. Aktor mengeklik pemesanan kamar	4. Sistem akan menampilkan form pemesanan kamar
5. Aktor memasukkan data penyewa id kamar	
6. Aktor mengeklik pesan	7. Sistem akan memverifikasi data yang di input
	1. Sistem akan membuka koneksi ke database
	2. Sistem akan menyimpan data inputan
	3. Sistem menampilkan notifikasi berhasil

### 4.3.4 Use case deskripsi fitur mengelola data kamar

Table 4.4 Use case deskripsi fitur mengelola data kamar

Nama	Mengelola data kamar
aktor	Pemilik kost
deskripsi	Pemilik kost akan mengelola kost yang di input oleh pemilik kost
Skenario normal	
1. Pemilik kost memilih menu Kelola Data Kamar.	2. Sistem menampilkan daftar kamar yang sudah ada.
3. Pemilik kost memilih aksi:	4. Sistem menyimpan perubahan data ke database.
a) Tambah Kamar mengisi form data kamar (nomor, harga, fasilitas, status).	
b) Edit Kamar mengubah data kamar yang sudah ada.	
c) Hapus Kamar menghapus data kamar tertentu.	
	5. Sistem menampilkan pesan sukses (data kamar berhasil ditambah/diubah/dihapus).



#### 4.3.4 Use case deskripsi fitur mengelola user

Table 4.5 Use case deskripsi fitur mengelola user

Nama	Mengelola user
aktor	Admin
deskripsi	Admin mengelola data user penyewa dan pemilik kos seperti tambah, edit, dan hapus user
Preconditions	Admin sudah login dan memiliki akses penuh ke dashboard admin
Postconditions	Data user diperbarui sesuai perubahan
Exception	Data tidak valid / database error
Skenario normal	
1. Admin memilih menu Kelola User.	2. Sistem menampilkan daftar user yang terdaftar.
3. Admin memilih aksi: a) Tambah user, mengisi form user, simpan. b) Edit user memperbarui data user yang sudah ada, simpan. c) Hapus user menghapus user.	4. Sistem memproses dan menyimpan perubahan data ke database.
	5. Sistem menampilkan pesan sukses (data kamar berhasil ditambah/diubah/dihapus).

#### 4.3.5 Use case deskripsi Upload Bukti Pembayaran

Table 4.6 Use case deskripsi fitur mengelola user

Nama	Upload Bukti Pembayaran
aktor	Pencari
deskripsi	Penyewa mengunggah bukti transfer pembayaran untuk pemesanan kos
Preconditions	Penyewa sudah login dan memiliki pesanan aktif
Postconditions	Bukti pembayaran tersimpan dan menunggu validasi Admin/Pemilik
Exception	Format file tidak valid / ukuran file terlalu besar
Skenario normal	
1. Penyewa memilih menu Pemesanan.	2. Sistem menampilkan daftar pesanan menunggu pembayaran.

3. Penyewa menekan tombol Upload Bukti.	4. Penyewa memilih file bukti pembayaran (format: JPG/PNG/PDF).
5. Penyewa menekan tombol <b>Kirim</b> .	6. Sistem menyimpan bukti ke database/storage.
7. Sistem menampilkan pesan berhasil.	
Skenario alternatif	
1. Sistem menampilkan pesan file tidak valid.	2. Penyewa mengunggah ulang file.

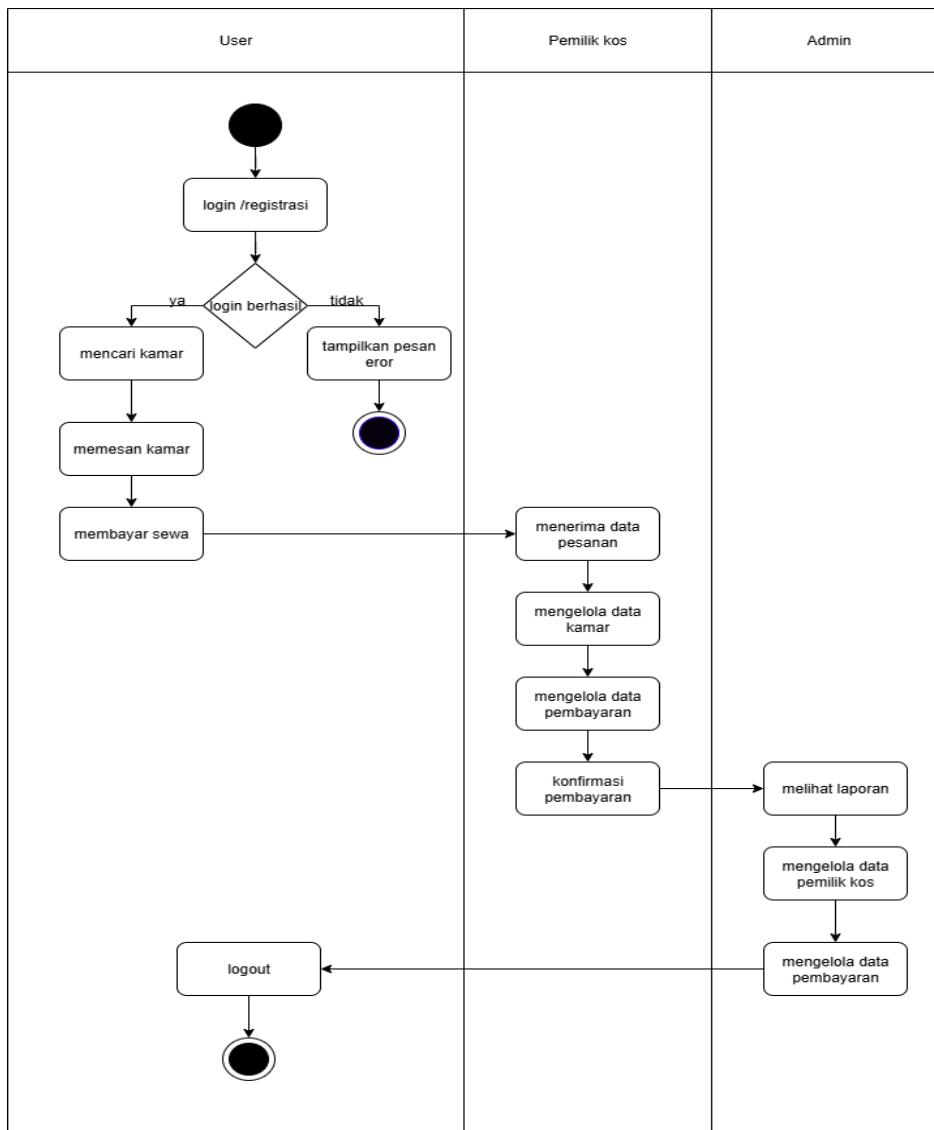
#### 4.3.6 Use case deskripsi Cetak Bukti Pemesanan

Table 4.7 Use case deskripsi Cetak Bukti Pemesanan

Nama	Cetak Bukti Pemesanan
aktor	Pencari
deskripsi	Sistem menyediakan bukti pemesanan berbentuk file PDF yang dapat diunduh atau dicetak
Preconditions	Penyewa sudah melakukan pemesanan
Postconditions	File PDF berhasil diunduh/ditampilkan
Exception	File gagal dihasilkan atau jaringan tidak stabil
Skenario normal	
1. Penyewa memilih menu Pemesanan.	2. Penyewa memilih menu Pemesanan.
3. Penyewa menekan tombol Cetak / Download Bukti Pemesanan.	4. Sistem menghasilkan file PDF bukti pemesanan.
5. File ditampilkan atau tersimpan pada perangkat penyewa.	

## 4.1 Activity Diagram

Gambar 4.2 Activity diagram



#### 4.1.1 *Activity Diagram* fitur login dan registrasi

Gambar 4.3 *Activity Diagram* fitur login dan registrasi

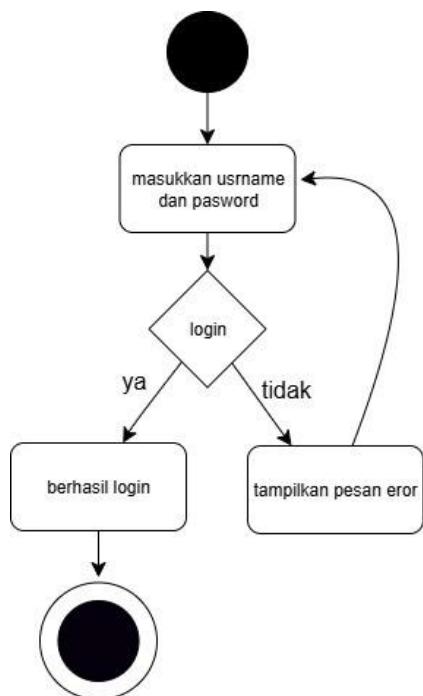


Diagram ini menampilkan alur *login* dan registrasi. *User* melakukan *input* data, sistem melakukan validasi, jika data benar maka *login* berhasil, jika salah maka akan menampilkan pesan *error*.

#### 4.1.2 *Activity Diagram* fitur pemesanan kamar

Gambar 4.4 *Activity Diagram* fitur pemesanan kamar

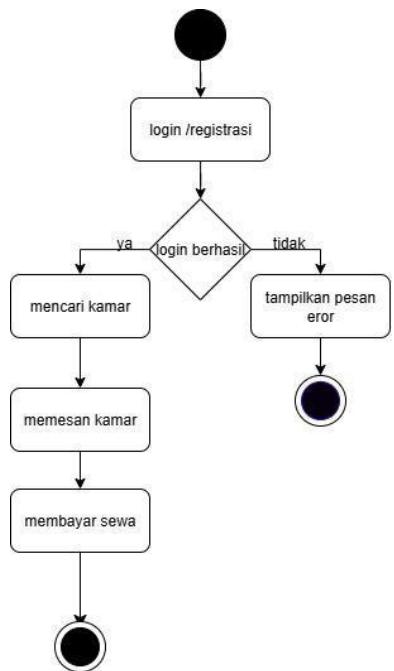
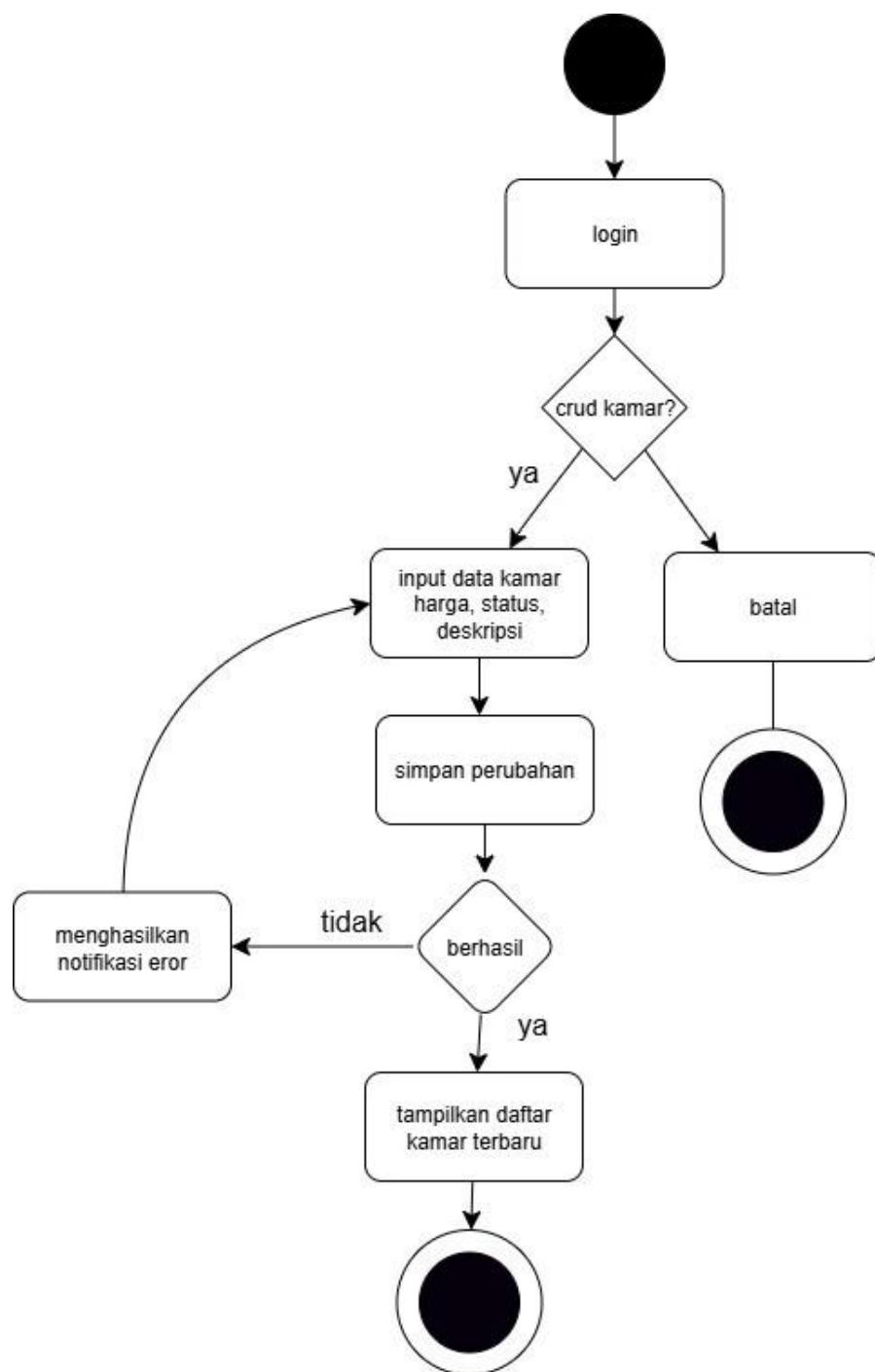


Diagram ini menggambarkan proses pemesanan kamar kost. Setelah *user login* mencari kamar, memesan kamar, dan langsung dapat melakukan pembayaran.

#### 4.1.3 *Activity Diagram* fitur Kelola data kamar

Gambar 4.5 *Activity Diagram* fitur Kelola data kamar

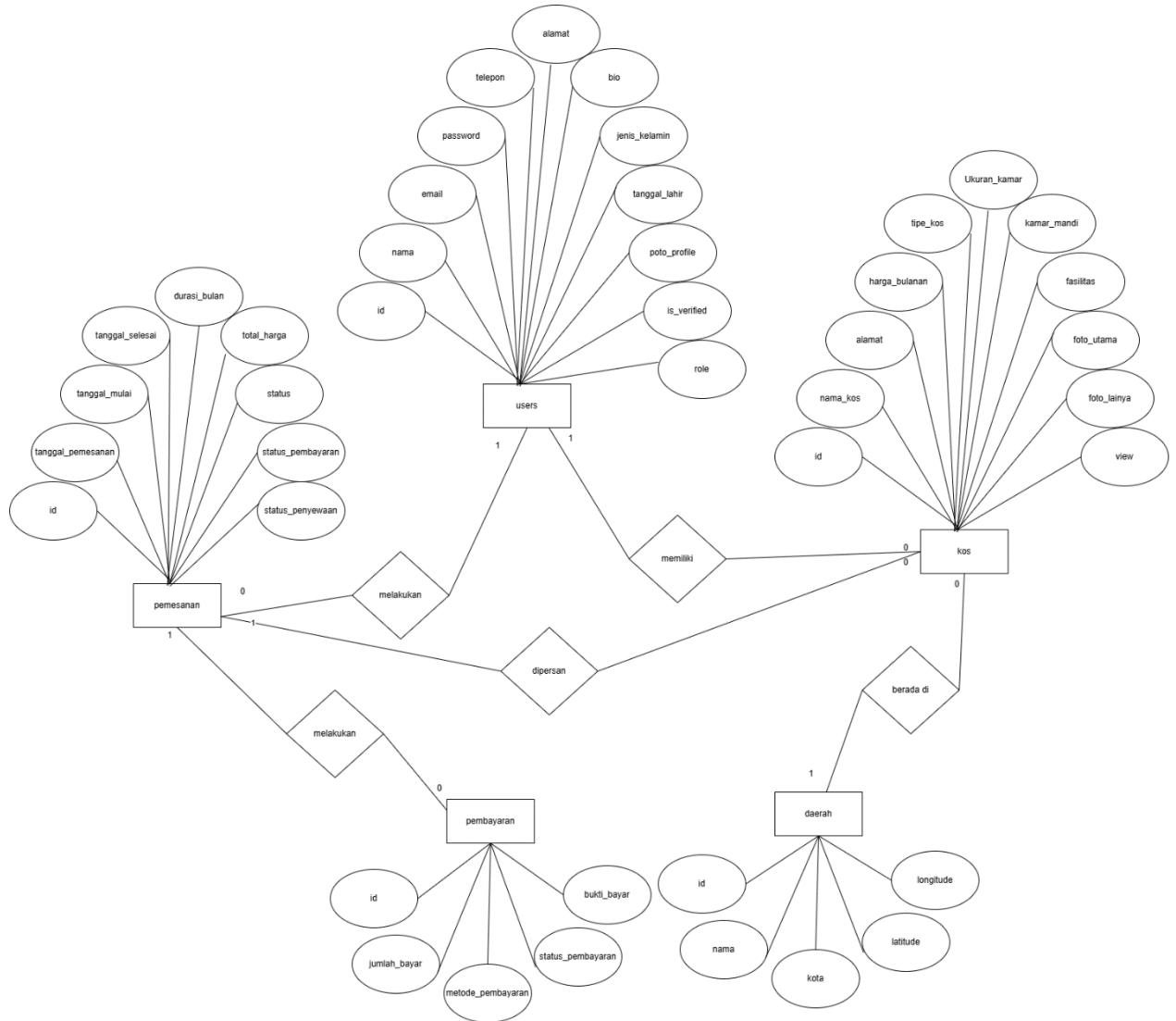


Pemilik kost dapat mengelola data kamar, mulai dari menambah, mengedit, atau menghapus data kamar. Setelah perubahan disimpan, sistem akan menampilkan daftar kamar terbaru.

## 4.2 Kebutuhan database

### 4.2.1 Entity Relationship Diagram

Gambar 4.6 Entity Relationship Diagram(ERD)



#### 1) User

- Menyimpan semua data pengguna aplikasi dari *Admin*, pemilik kost, dan penyewa
- Satu *user* (Pemilik kost) bisa memiliki banyak kamar
- Satu *user* (Penyewa) bisa memiliki banyak pemesanan

- d) Satu *user* (*Admin*) bisa memiliki banyak laporan

- 2) Kamar
  - a) Menyimpan data semua kamar kost
  - b) Satu kamar dapat di pesan oleh banyak penyewa
  - c) Setiap kamar dimiliki oleh seorang *user* (pemilik kost)
- 3) Pemesanan
  - a) Mencatat transaksi pemesanan kamar oleh penyewa
  - b) Satu pemesanan dapat di pesan oleh *user*
  - c) Satu pemesanan terkait dengan satu kamar
  - d) Setiap pemesanan memiliki satu pembayaran
- 4) Pembayaran
  - a) Menyimpan data pembayaran sewa kamar yang dilakukan oleh penyewa
  - b) Satu pembayaran terhubung dengan satu pemesanan
  - c) Satu pembayaran dapat dicatat di banyak laporan
- 5) Laporan
  - a) Berisi laporan transaksi yang dibuat oleh *Admin*
  - b) Laporan dihasilkan berdasarkan data pembayaran.



## 4.2.2 Class Diagram

Gambar 4.7 Class Diagram

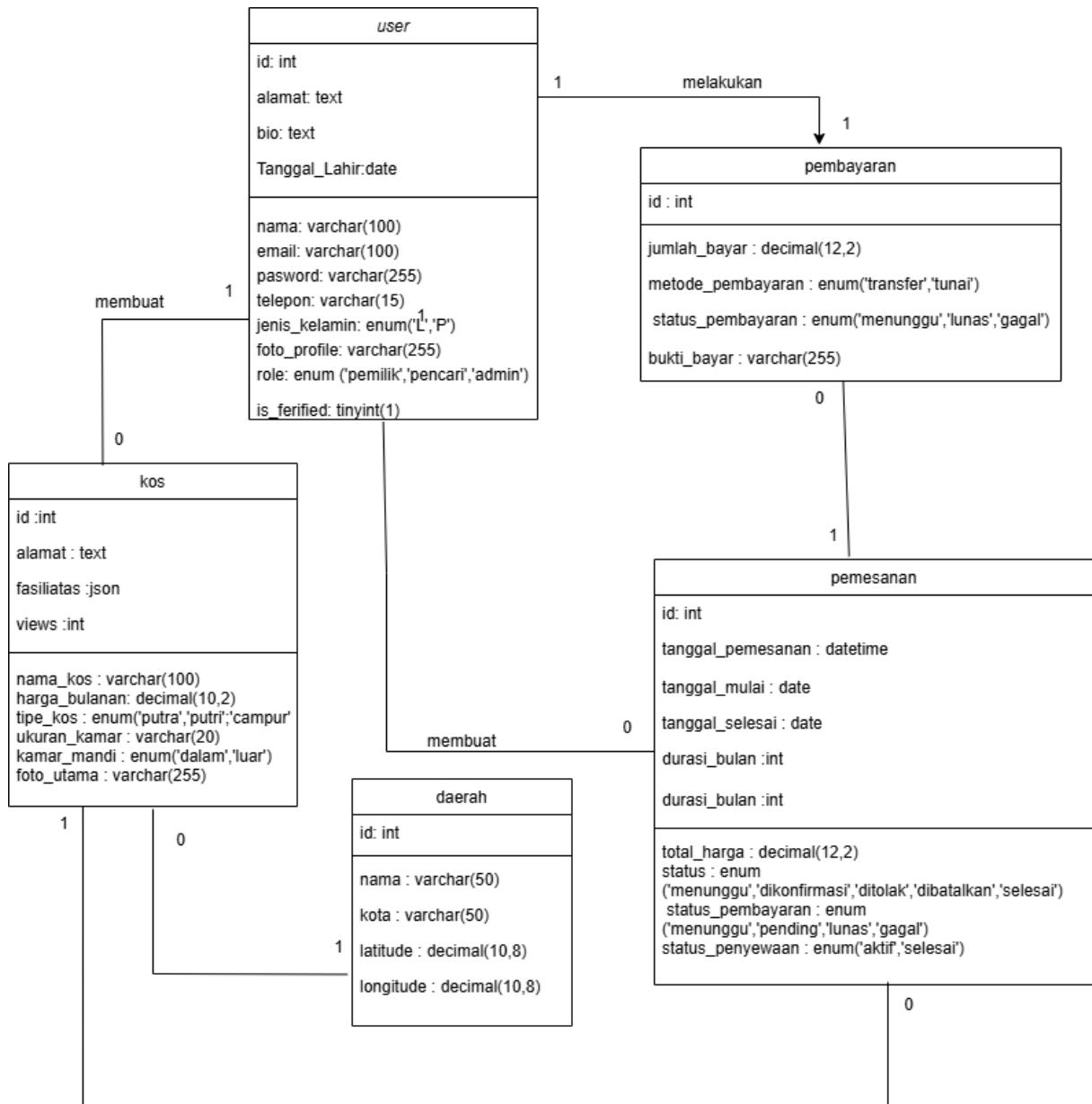


Diagram tersebut menggambarkan struktur database untuk sistem pencarian dan penyewaan kos. Dalam sistem ini terdapat pengguna atau *user* sebagai aktor utama yang dapat berperan sebagai pemilik kos, pencari kos, maupun admin. Setiap *user* memiliki data pribadi seperti nama, email, nomor telepon, alamat, informasi profil, serta status

verifikasi akun. Seorang user yang bertindak sebagai pemilik dapat mendaftarkan banyak kos dalam sistem. Informasi setiap kos disimpan dalam tabel terpisah yang mencakup nama kos, alamat, fasilitas kamar, jenis kos (putra, putri, campur), harga bulanan, ukuran kamar, hingga foto pendukung. Selain itu, setiap kos memiliki hubungan dengan tabel daerah, yang menyimpan informasi wilayah seperti nama, kota, serta titik koordinat lokasi. Relasi ini menunjukkan bahwa satu daerah dapat memiliki banyak kos di dalamnya.

Pencari yang berperan sebagai penyewa dapat melakukan pemesanan terhadap kos yang tersedia. Data pemesanan disimpan dalam tabel *pemesanan*, yang memuat informasi seperti tanggal pemesanan, tanggal mulai dan selesai penyewaan, durasi, total harga, serta status pemesanan dan penyewaan. Setiap pemesanan terkait langsung dengan satu kos dan dilakukan oleh seorang user, tetapi satu kos bisa menerima banyak pemesanan dari user yang berbeda.

Proses selanjutnya dalam penyewaan adalah pembayaran. Setiap pemesanan dapat memiliki satu atau lebih data pembayaran yang tercatat dalam tabel *pembayaran*. Hal ini memungkinkan pembayaran dilakukan secara cicilan atau bertahap. Informasi yang tersimpan meliputi jumlah yang dibayar, metode pembayaran (tunai atau transfer), status pembayaran, serta bukti pembayaran berupa file gambar atau dokumen.

Secara keseluruhan, diagram ini menunjukkan hubungan yang jelas antara pemilik kos, properti kos, lokasi kos, proses pemesanan, hingga transaksi pembayaran yang dilakukan penyewa. Struktur database tersebut dirancang untuk mendukung alur kegiatan utama dalam aplikasi penyewaan kos: pemilik mengelola kos, pengguna mencari dan memesan kos, serta melakukan pembayaran sesuai ketentuan yang ditetapkan

### 4.3 Sequence Diagram

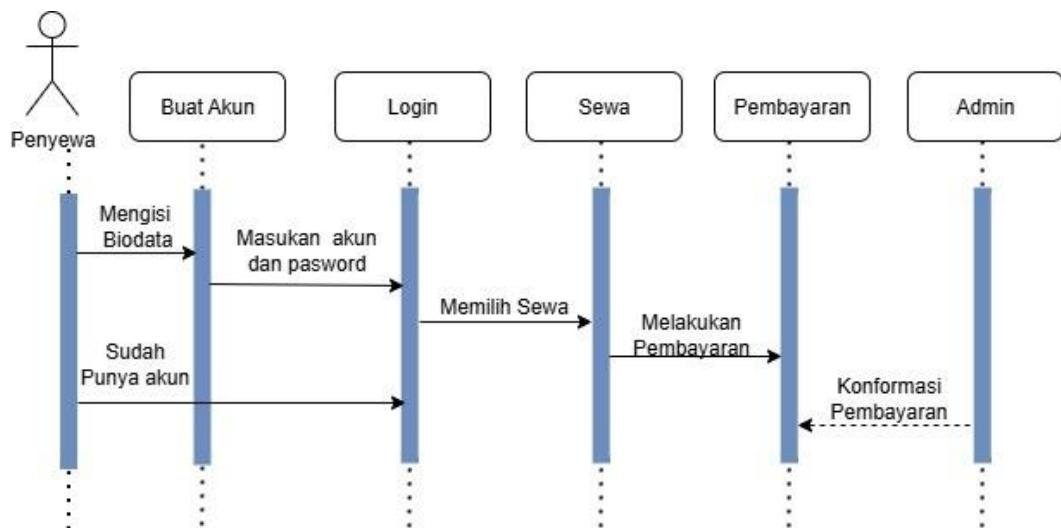
Repository Universitas Komputer Indonesia (Unikom) menjelaskan bahwa *sequence diagram* merupakan salah satu jenis diagram UML yang menggambarkan urutan interaksi antar objek dalam suatu sistem. Diagram ini berfungsi untuk membantu proses analisis, perancangan sistem, pengembangan subsistem, serta identifikasi kebutuhan.

Selain itu, dijelaskan pula komponen utama seperti *lifeline* dan *message*, serta konsep lanjutan mencakup *execution* dan *flow control*.

Secara keseluruhan, *sequence diagram* berperan penting dalam memahami alur komunikasi dan pembagian tugas antar objek atau kelas secara kronologis dalam suatu skenario sistem.

#### 4.3.1 Sequence Diagram Fitur Penyewa

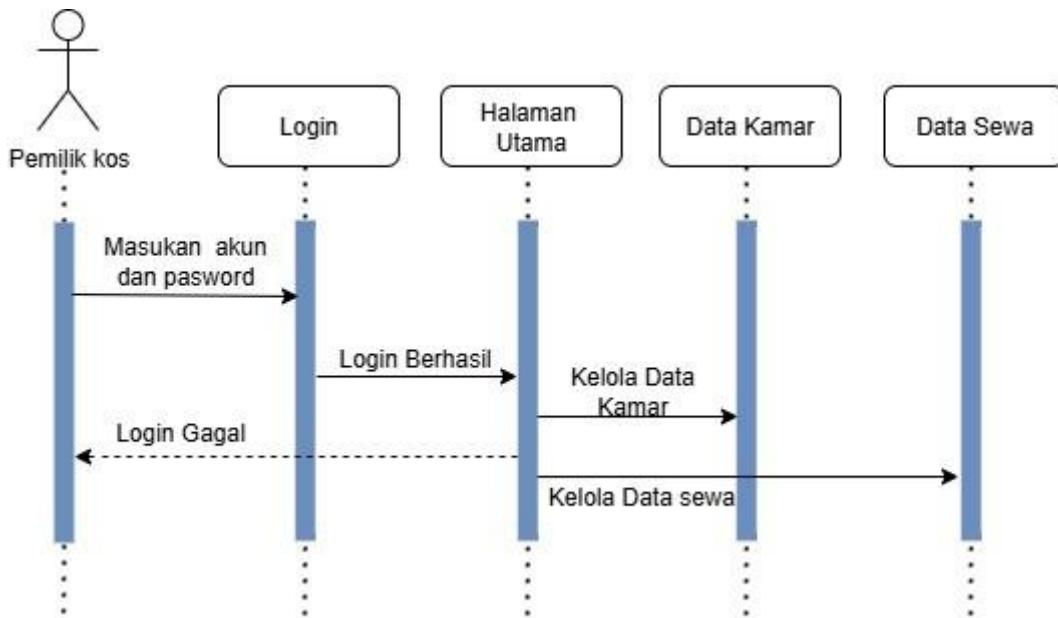
Gambar 4.10 Sequence Diagram Fitur Penyewa





### 4.3.2 Sequence Diagram Fitur Pemilik kos

Gambar 4.11 Sequence Diagram Fitur Pemilik kost



## 5 Kebutuhan Non-Fungsional Produk atau Aplikasi

### 5.1 Kebutuhan Produk

#### 5.1.1 Usability

Kami membuat sistem ini dengan memperhatikan aspek kemudahan penggunaan agar nyaman digunakan oleh para pengguna. Tampilan *layout* dibuat sederhana dengan menggunakan *Bootstrap* dan *CSS*. Tombol-tombol disusun dengan tata letak yang rapi dengan melihat referensi yang ada di web inkost pada posisi yang mudah dicari serta mudah diidentifikasi fungsinya. Setiap menu ditulis dengan jelas dan sesuai penamaan seperti "pesan" serta menggunakan bahasa yang mudah dimengerti oleh pengguna. Tulisan atau teks pada setiap halaman ditampilkan dengan ukuran, warna, dan kontras yang nyaman dibaca sesuai tema kami yaitu inkost. Selain itu, apabila terjadi kesalahan pada sistem, aplikasi akan memberikan pesan *error* yang jelas sehingga pengguna dapat memahami masalah yang terjadi dan mengetahui langkah selanjutnya.

### **5.1.2 Efficiency**

#### **5.1.2.1 Performance**

Aplikasi Inkost dirancang agar berjalan dengan cepat dan stabil. Halaman utama dapat dimuat dalam waktu maksimal tiga detik pada koneksi internet minimal 5 Mbps, sedangkan setiap menu seperti *Login*, Cari Kost, dan Daftar Kost memiliki waktu *loading* sekitar dua detik. Sistem mendukung operasional penuh selama 24 jam tanpa batasan waktu penggunaan. Pada proses *login* tidak ada batasan. Selain itu, fitur pencarian kost mampu menampilkan hasil dengan filter sederhana seperti harga, lokasi, dan fasilitas dalam waktu kurang dari tiga detik.

#### **5.1.2.2 Space**

Inkost berbasis web sehingga tidak memerlukan instalasi khusus pada perangkat pengguna. Penyimpanan hanya dibutuhkan untuk cache *browser* sekitar 50 hingga 100 MB. Untuk memastikan kelancaran penggunaan, perangkat *mobile* disarankan memiliki RAM minimal 2 GB, sedangkan laptop atau PC disarankan memiliki RAM minimal 4 GB. *Bandwidth* internet yang dibutuhkan setidaknya 5 Mbps agar integrasi *Google Maps* dapat ditampilkan secara optimal.

#### **5.1.2.3 Security**

Aplikasi Inkost menerapkan keamanan dengan cara menyimpan *password* dalam bentuk terenkripsi, mewajibkan pengguna membuat *password* minimal enam karakter dan mengandung angka agar lebih sulit ditebak, serta dijalankan pada server yang terjamin baik dan stabil sehingga data pengguna tetap aman.

## **6 Bab Desain Sistem**

### **a. Arsitektur Sistem (Wajib)**

Monolitik

### **b. Activity Diagram (Minimal 2–3 diagram utama)**

Menggambarkan alur proses inti aplikasi.

### **c. ERD Physical (Wajib)**

- Entity
- Relationship

- Atribut
- Primary Key / Foreign Key
- Tipe data (physical model)

**d. Class Diagram (Wajib)**

- Representasi class, atribut, method, dan relasi antar class.

**e. Design UI (Wajib)**

- Tampilan mockup atau wireframe
- Minimal halaman, Contoh :
  - halaman login (jika ada)
  - halaman utama
  - halaman dashboard admin

**f. (Opsional jika kelompok ingin menambah)**

- Sequence Diagram
- DFD
- State Machine Diagram