
Software Requirement Specification

Document

JuraganKost: Sistem Manajemen Kost

Gibran Rafi Pratama

Document ID	SOFTWARE REQUIREMENT SPECIFICATION-v0.1
Version Number	0.1
Issue Date	December 01, 2025
Classification	Public

Copyright Notice

© COMPANYNAME, (original issue year – current issue year)

All Rights Reserved

The information contained in this document is the property of JuraganKost. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means; mechanical, photocopying, recording, or otherwise, without the prior written consent of JuraganKost. Under the law, copying includes translating into another language or format. Legal action will be taken against any infringement.

The information contained in this document is subject to change without notice and does not carry any contractual obligation for JuraganKost. JuraganKost reserves the right to make changes to any products or services described in this document at any time without notice. JuraganKost shall not be held responsible for the direct or indirect consequences of the use of the information contained in this document.

Revision History

Date	Version	Description	Author (s)
11/12/2025	0.1	Draft Version dan Revisi Usecase	Gibran Rafi Pratama
20/12/2025	0.1	Membuat rancangan Database	Gibran Rafi Pratama

Reviewed By (Customer)	Signature	Date

The reviewer signoff shall signify the recommendation for acceptance of this document.

Sign Off

Prepared By	Acknowledged By
<Name>	<Name>
Title: <Position>	Title: <Position>
COMPANYNAME	COMPANYNAME
Date:	Date:

Accepted By	Accepted By
<Name>	<Name>
Title: <Position>	Title: <Position>
<Customer Company Name >	<Customer Company Name >
Date:	Date:

Daftar Isi

1	Introduction.....	7
1.1	Purpose	7
1.2	Document Conventions	7
1.3	Intended Audience and Reading Suggestions.....	7
1.4	Project Scope	7
1.5	References	Error! Bookmark not defined.
1.6	Acronyms and Abbreviations	Error! Bookmark not defined.
2	Overall Description	8
2.1	Product Perspective	8
2.2	Product Features.....	8
2.3	User Classes and Characteristics.....	9
2.4	Operating Environment	9
2.5	Design and Implementation Constraints	9
2.6	User Documentation.....	Error! Bookmark not defined.
2.7	Assumptions and Dependencies	Error! Bookmark not defined.
3	System Features.....	12
3.1	System Feature 1	Error! Bookmark not defined.
3.1.1	Description and Priority.....	Error! Bookmark not defined.
3.1.2	Stimulus/Response Sequences.....	Error! Bookmark not defined.
3.1.3	Functional Requirements	Error! Bookmark not defined.
3.2	System Feature 2 (and so on)	Error! Bookmark not defined.
4	External Interface Requirements	17
4.1	User Interfaces	17
4.2	Hardware Interfaces	17
4.3	Software Interfaces.....	17
4.4	Communications Interfaces	Error! Bookmark not defined.
5	Other Nonfunctional Requirements.....	18

5.1	Performance Requirements.....	18
5.2	Safety Requirements.....	18
5.3	Security Requirements.....	18
5.4	Software Quality Attributes	18
	Appendix A: Glossary	19
	Appendix B: Analysis Models.....	Error! Bookmark not defined.
	Appendix C: Issues List	Error! Bookmark not defined.

1 Introduction

1.1 Purpose

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SRS) ini disusun untuk mendefinisikan secara menyeluruh kebutuhan teknis dan fungsional dari proyek Aplikasi Manajemen Kos "JuraganKost" Berbasis Android. Perangkat lunak yang dikembangkan adalah Versi 1.0 (Proyek Tugas Akhir). Tujuan utama proyek ini adalah menyediakan solusi digital berbasis *mobile* untuk menggantikan pencatatan buku manual dalam pengelolaan rumah kos.

1.2 Document Conventions

Dokumen ini menggunakan konvensi penulisan sebagai berikut:

- **Bold (Tebal):** Istilah penting, nama fitur, atau elemen UI.
- *Italic (Miring):* Istilah teknis asing (contoh: REST API, RecyclerView).
- Prioritas fitur: **HIGH** (Tinggi), **MEDIUM** (Sedang), **LOW** (Rendah).

1.3 Intended Audience and Reading Suggestions

Dokumen SRS ini ditujukan untuk:

- **Developer:** Panduan implementasi aplikasi Android dengan arsitektur MVVM dan komunikasi REST API menggunakan database MySQL sebagai penyimpanan data utama.
- **Dosen/Penguji:** Mengevaluasi kelayakan proyek Tugas Akhir.
- **Pemilik Kost:** Sebagai pengguna utama untuk memvalidasi alur bisnis.

1.4 Project Scope

JuraganKost adalah aplikasi manajemen properti berbasis Client-Server di mana aplikasi Android bertindak sebagai client dan server XAMPP dengan database MySQL berperan sebagai penyimpanan data utama. Aplikasi ini bertujuan untuk:

1. Mengelola data master kamar secara terpusat.
2. Mencatat transaksi penghuni secara *real-time*.
3. Menyediakan sinkronisasi data antar perangkat admin.

Sistem ini dibangun menggunakan arsitektur terdistribusi di mana aplikasi Android bertindak sebagai *Client* dan server XAMPP (MySQL) bertindak sebagai penyimpan data utama (*Backend*). Koneksi jaringan (Wi-Fi/Internet) diperlukan untuk operasional penuh.

1.5 References

- Entity Relationship Diagram (ERD) JuraganKost
- Flowchart Aplikasi JuraganKost
- Use Case Diagram JuraganKost

1.6 Acronyms and Abbreviations

Acronym	Definition
SRS	Software Requirement Specification
MVVM	Model-View-ViewModel (Arsitektur Android)
MySQL	Library Database Server
DAO	Data Access Object
UI/UX	User Interface / User Experience

2 Overall Description

2.1 Product Perspective

Aplikasi ini merupakan bagian dari sistem yang lebih besar yang terdiri dari dua komponen utama:

1. **Aplikasi Mobile (Android):** Antarmuka pengguna untuk melakukan input dan manajemen data.
2. **Web Server (Backend):** Hosting database MySQL dan REST API menggunakan XAMPP.

Berbeda dengan sistem offline, JuraganKost menyimpan seluruh data di server pusat, memungkinkan konsistensi data yang tinggi dan keamanan sentralisasi.

2.2 Product Features

Fitur yang disediakan meliputi:

- **Otentikasi Lokal:** Login sederhana untuk mengamankan data admin.
- **Manajemen Kamar:** CRUD data kamar (Nomor, Harga, Status).
- **Manajemen Penghuni:** Proses *Check-In* (Masuk) dan *Check-Out* (Keluar) yang otomatis mengupdate status kamar.
- **Pencatatan Keuangan:** Input pembayaran bulanan penghuni.
- **Dashboard Visual:** Indikator warna (Merah/Hijau) untuk status hunian.

2.3 User Classes and Characteristics

Hanya terdapat satu kelas pengguna dalam sistem ini:

- Admin (Pemilik Kost): Pengguna umum yang mungkin tidak memiliki latar belakang teknis tinggi. Karakteristik pengguna membutuhkan antarmuka yang sederhana, ikon yang jelas, dan alur kerja yang tidak rumit.

2.4 Operating Environment

Spesifikasi	Jenis
Platform Mobile	Android (Minimal versi 8.0 Oreo)
Database Server	MySQL / MariaDB (via XAMPP)
Web Server	Apache (via XAMPP)
Bahasa Backend	PHP Native / Laravel (untuk API)
Koneksi Internet	Wajib (Wi-Fi/LAN/Hotspot) untuk sinkronisasi

2.5 Design and Implementation Constraints

Penggunaan	Jenis	Penjelasan
Bahasa Pemrograman	Kotlin	Digunakan sebagai bahasa utama dalam pengembangan aplikasi Android Native karena kompatibilitasnya yang baik dengan library Jetpack.

Basis Data	MySQL	Digunakan sebagai penyimpanan data terpusat (<i>Centralized Database</i>) yang diakses melalui jaringan.
Model Arsitektur Sistem	MVVM (Model - View - ViewModel)	Digunakan untuk memisahkan logika bisnis, antarmuka pengguna (UI), dan data agar aplikasi lebih modular, mudah diuji, dan kode lebih rapi.
Komunikasi Data	REST API	Menggunakan protokol HTTP dengan format JSON untuk pertukaran data antara Android dan Server.
Library Network	Retrofit / Volley	Library Android untuk menangani <i>request</i> ke server (GET, POST, PUT, DELETE).

2.6 User Documentation

1. Login

Admin membuka aplikasi.

- Masukkan username dan password.
- Tekan tombol **Login/Register** jika tidak memiliki akun.
- Jika benar, masuk ke Dashboard. Jika salah, muncul pesan error.

2. Mengakses Dashboard

Halaman utama menampilkan kartu-kartu kamar.

- **Warna Hijau:** Kamar Kosong (Bisa diisi).
- **Warna Merah:** Kamar Terisi (Ada penghuninya).
- Menu Navigasi: Kelola Kamar, Riwayat Transaksi, Logout.

3. Kelola Data Kamar

- **Menambah Kamar:** Klik tombol (+), isi Nomor Kamar dan Harga, klik Simpan.
- **Edit/Hapus:** Tekan lama pada salah satu kamar di list untuk opsi Edit atau Hapus.

4. Proses Check-In (Penghuni Masuk)

- Pilih salah satu **Kamar Kosong** di Dashboard.

- Klik tombol "**Tambah Penghuni**".
- Isi Form: Nama, No HP, Upload Foto KTP.
- Klik **Simpan**. Status kamar otomatis berubah menjadi "Terisi".

5. Pencatatan Pembayaran

- Pilih **Kamar Terisi** di Dashboard.
- Klik tombol "**Catat Bayar**".
- Pilih Bulan dan masukkan Nominal.
- Klik Simpan. Data masuk ke menu Riwayat Transaksi.

6. Proses Check-Out (Penghuni Keluar)

- Pilih **Kamar Terisi**.
- Klik tombol "**Check-Out**".
- Konfirmasi dialog "Ya".
- Data penghuni diarsipkan, status kamar kembali menjadi "Kosong".

2.7 *Assumptions and Dependencies*

- Pengguna menyediakan perangkat Android dengan ruang penyimpanan yang cukup untuk menyimpan foto KTP.
- Aplikasi tidak akan hilang datanya kecuali aplikasi di-uninstall atau data di-clear manual via pengaturan HP.

3 System Features

3.1 System Feature: Otentikasi (Auth)

3.1.1 Registrasi Akun Baru

Description: Fitur ini memungkinkan pengguna baru (Pemilik Kost) untuk mendaftarkan akun agar dapat mengakses sistem. **Priority:** High.

Stimulus/Response:

- Saat membuka aplikasi pertama kali, Admin memilih menu "Register".
- Admin mengisi Username dan Password.
- Sistem melakukan validasi (tidak boleh kosong).
- Sistem menyimpan dan memverifikasi data akun Admin melalui database MySQL di server menggunakan REST API.

Functional Requirements:

- **REQ-AUTH-REG-01:** Sistem harus memvalidasi agar username dan password tidak boleh kosong.
- **REQ-AUTH-REG-02:** Sistem harus menyimpan *hash* password untuk keamanan, bukan teks biasa.

3.1.2 Login (Masuk Sistem)

Description: Memverifikasi identitas Admin sebelum memberikan akses ke Dashboard. **Priority:** High.

Stimulus/Response:

- Admin memasukkan Username dan Password di halaman Login.
- Sistem mencocokkan data dengan tabel User.
- Jika cocok, sistem mengarahkan ke Dashboard. Jika salah, muncul pesan error.

Functional Requirements:

- **REQ-AUTH-LOG-01:** Sistem harus menolak akses jika username tidak ditemukan atau password salah.

- **REQ-AUTH-LOG-02:** Sistem harus mengarahkan pengguna ke Dashboard Utama setelah login berhasil.

3.1.3 Logout (Keluar Sistem)

Description: Mengakhiri sesi akses admin dan kembali ke halaman Login untuk mencegah akses tidak sah. **Priority:** Medium.

Stimulus/Response:

- Di halaman Dashboard, Admin menekan tombol "Logout".
- Sistem menghapus sesi aktif dan mengembalikan tampilan ke halaman Login.

Functional Requirements:

- **REQ-AUTH-OUT-01:** Sistem harus memastikan pengguna tidak bisa menekan tombol "Back" kembali ke Dashboard setelah Logout (Clear Task Stack).

3.2 System Feature: Manajemen Data Kost (Multi-Kost)

3.2.1 Description and Priority

Admin dapat menambahkan cabang kost baru, mengubah alamat, atau menghapus data kost. Ini adalah syarat utama sebelum admin bisa menambahkan kamar.

Priority: High

3.2.2 Stimulus/Response Sequences

- Admin login -> Sistem mengecek apakah admin sudah punya kost.
- Jika belum, admin diarahkan ke form "Tambah Kost".
- Admin mengisi Nama Kost dan Alamat -> Klik Simpan.
- Sistem menyimpan ke server MySQL.

3.2.3 Functional Requirements

- **REQ-KOST-ADD-01:** Sistem harus mengizinkan satu akun Admin memiliki banyak cabang Kost (One-to-Many).

- **REQ-KOST-ADD-02:** Sistem wajib meminta input Nama Kost dan Alamat tidak boleh kosong.
- **REQ-KOST-VIEW-01:** Sistem harus menampilkan daftar kost yang dimiliki Admin setelah login berhasil.

3.3 System Feature: Manajemen Kamar (Room Management)

Fitur ini mencakup pengelolaan data master kamar kost.

3.3.1 Melihat Daftar Kamar (Dashboard)

Description: Menampilkan status seluruh kamar (Terisi/Kosong) dalam bentuk Grid/List. **Priority:** High.

Stimulus/Response:

- Admin berhasil Login -> Sistem menampilkan Dashboard.
- Sistem mengambil (*Query*) semua data dari tabel Room.
- Sistem menampilkan kartu kamar dengan indikator warna (Hijau: Kosong, Merah: Terisi).

Functional Requirements:

- **REQ-ROOM-VIEW-01:** Sistem harus menampilkan Nomor Kamar dan Harga Sewa.
- **REQ-ROOM-VIEW-02:** Sistem harus memberikan indikator visual (warna/ikon) berdasarkan status hunian.

3.3.2 Menambah Kamar Baru.

Description: Menambahkan data kamar baru ke dalam database. **Priority:** High.

Stimulus/Response:

- Admin menekan tombol "Tambah Kamar" -> Muncul Form Input.
- Admin mengisi Nomor Kamar dan Harga -> Klik Simpan.

- Sistem menyimpan ke database -> List di Dashboard bertambah.

Functional Requirements:

- **REQ-ROOM-ADD-01:** Sistem harus memvalidasi agar Nomor Kamar tidak duplikat (unik).
- **REQ-ROOM-ADD-02:** Sistem menyimpan data dengan status *default* "Kosong" (False).

3.3.3 Edit & Hapus Kamar

Description: Mengubah informasi harga atau menghapus kamar yang sudah tidak ada. **Priority: Medium.**

Stimulus/Response:

- Admin memilih salah satu kamar -> Klik Edit/Hapus.
- Jika Edit -> Update data di database.
- Jika Hapus -> Muncul konfirmasi "Yakin hapus?" -> Data terhapus dari database.

Functional Requirements:

- **REQ-ROOM-MOD-01:** Sistem harus menyediakan fungsi Update dan Delete pada tabel Room.
- **REQ-ROOM-MOD-02:** Sistem harus mencegah penghapusan jika kamar sedang status "Terisi".

3.4 System Feature: Manajemen Penghuni (Check-In/Check-Out)

3.4.1 Check-In (Penghuni Masuk)

Description: Proses memasukkan penghuni baru ke dalam kamar yang kosong.

Priority: High.

Stimulus/Response:

- Admin klik kamar berwarna Hijau (Kosong) -> Muncul Form "Check-In".
- Admin mengisi Nama, No HP, dan Upload Foto KTP.

- Admin klik Simpan -> Sistem update status kamar jadi Merah (Terisi).

Functional Requirements:

- **REQ-TENANT-IN-01:** Sistem harus menyimpan data diri ke tabel Tenant.
- **REQ-TENANT-IN-02:** Sistem harus menyimpan *path* (lokasi file) Foto KTP.
- **REQ-TENANT-IN-03:** Sistem otomatis mengubah status *is_occupied* menjadi True pada tabel Room.

3.4.2 Check-Out (Penghuni Keluar)

Description: Proses mengeluarkan penghuni dan mengosongkan kamar. **Priority:** High.

Stimulus/Response:

- Admin klik kamar berwarna Merah (Terisi) -> Muncul Detail Penghuni.
- Admin klik tombol "Check-Out" -> Konfirmasi Dialog "Ya".
- Sistem mengosongkan kamar -> Kamar kembali berwarna Hijau.

Functional Requirements:

- **REQ-TENANT-OUT-01:** Sistem mengubah status *is_occupied* menjadi False pada tabel Room.
- **REQ-TENANT-OUT-02:** Data penghuni tidak dihapus permanen, tapi diarsipkan (opsional) atau dihapus dari relasi kamar aktif.

3.5 System Feature: Pencatatan Pembayaran (Billing)

3.5.1 Catat Pembayaran Sewa

Description: Mencatat pembayaran bulanan dari penghuni. **Priority:** Medium.

Stimulus/Response:

- Pada Detail Penghuni, Admin klik "Catat Bayar".
- Input Nominal dan Bulan -> Klik Simpan.
- Sistem menyimpan riwayat transaksi.

Functional Requirements:

- **REQ-BILL-ADD-01:** Sistem menyimpan data ke tabel Payment dengan *Foreign Key* ID Penghuni.
- **REQ-BILL-ADD-02:** Sistem mencatat tanggal aktual transaksi (Timestamp).

3.5.2 Lihat Riwayat Transaksi

Description: Menampilkan daftar uang masuk. **Priority: Low.**

Stimulus/Response:

- Admin masuk menu Riwayat -> Sistem menampilkan list semua pembayaran.

Functional Requirements:

- **REQ-BILL-VIEW-01:** Sistem menampilkan daftar transaksi urut berdasarkan tanggal terbaru (*Descending*).

4 External Interface Requirements

4.1 User Interfaces

- **Halaman Dashboard:** Menggunakan *RecyclerView* dengan layout Grid. Item berupa *CardView*.
- **Form Input:** Menggunakan *TextInputLayout* dan *TextInputEditText* untuk validasi error yang rapi.
- **Dialog:** Menggunakan *AlertDialog* untuk konfirmasi Check-Out atau Hapus data.

4.2 Hardware Interfaces

- **Camera/Gallery:** Untuk mengambil/memilih foto KTP penghuni.
- **Internal Storage:** Untuk menyimpan file gambar KTP.

4.3 Software Interfaces

- **OS:** Android SDK API Level 26+.
- **Library:** Android Jetpack (ViewModel, LiveData).

4.4 Communications Interfaces

Sistem menggunakan protokol komunikasi standar HTTP/HTTPS:

- **Format Data:** JSON (JavaScript Object Notation).
- **Metode:** GET (Ambil data), POST (Kirim data), PUT (Update data), DELETE (Hapus data).
- **Keamanan:** Autentikasi token sederhana (opsional) atau validasi sisi server.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

- Aplikasi harus mampu memuat data dashboard dalam waktu kurang dari 3 detik pada kondisi koneksi internet yang normal.
- Database harus mampu menampung minimal hingga 1000 data transaksi tanpa penurunan performa yang signifikan.

5.2 Safety Requirements

- Sistem harus memiliki peringatan sebelum menghapus data penting (seperti menghapus kamar atau penghuni).

5.3 Security Requirements

- Akses masuk aplikasi harus dilindungi dengan Username dan Password yang tersimpan dalam bentuk hash di database MySQL pada server.

5.4 Software Quality Attributes

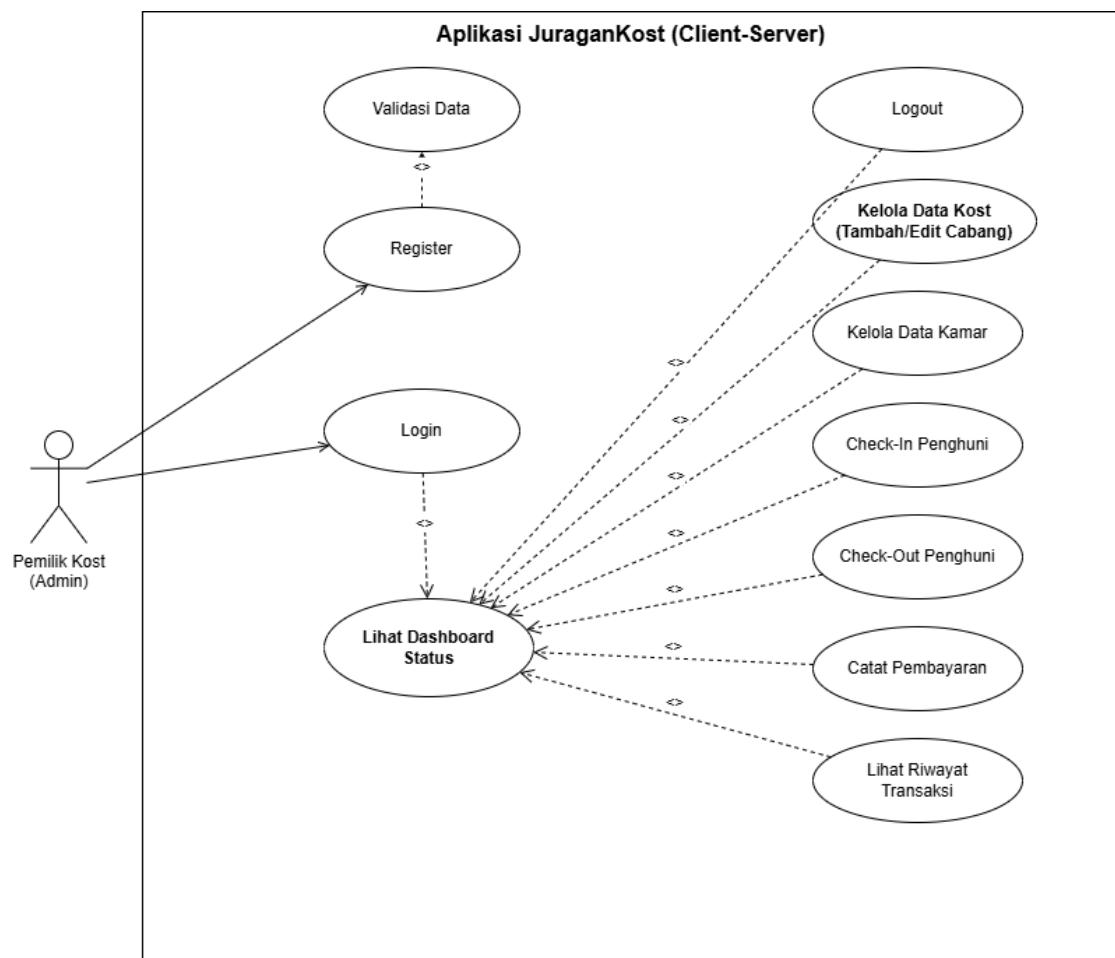
- Usability: Desain UI harus intuitif agar mudah digunakan oleh pemilik kost yang awam teknologi.
- Reliability: Aplikasi tidak boleh *force close* saat menangani input data yang tidak valid.

Appendix A: Glossary

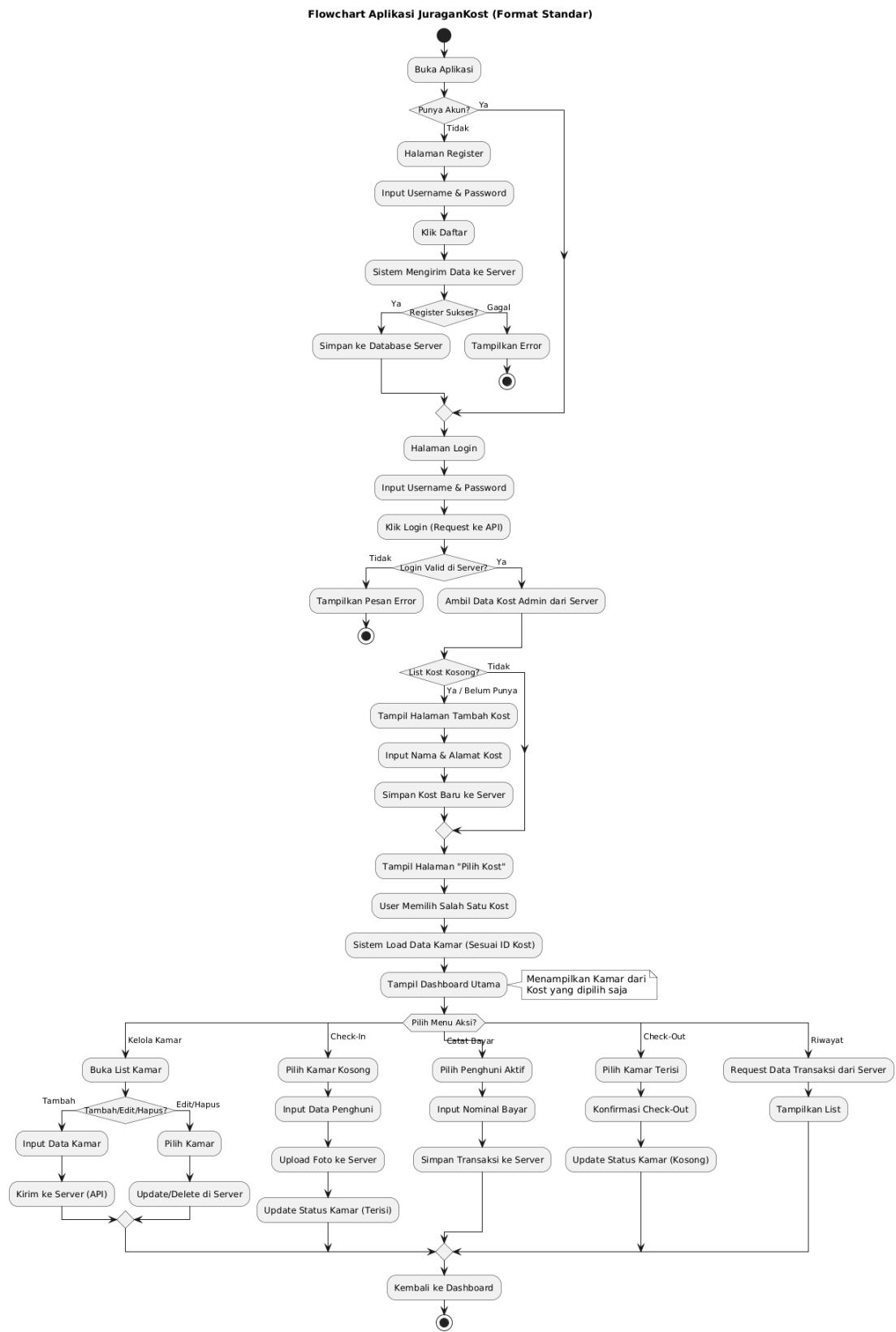
- SRS: Software Requirement Specification.
- MVVM: Model-View-ViewModel (Pola arsitektur perangkat lunak).
- MySQL: Database terpusat pada server
- Check-In: Proses pendaftaran masuk penghuni baru.
- Check-Out: Proses penyelesaian masa sewa penghuni.

Appendix B: Analysis Models

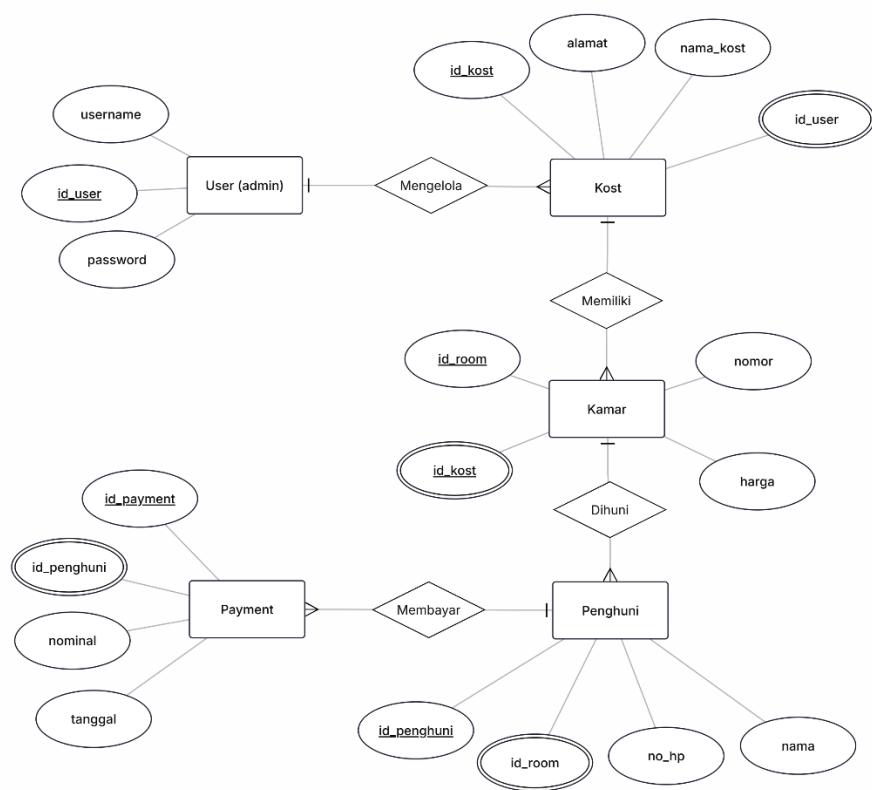
1. Use Case



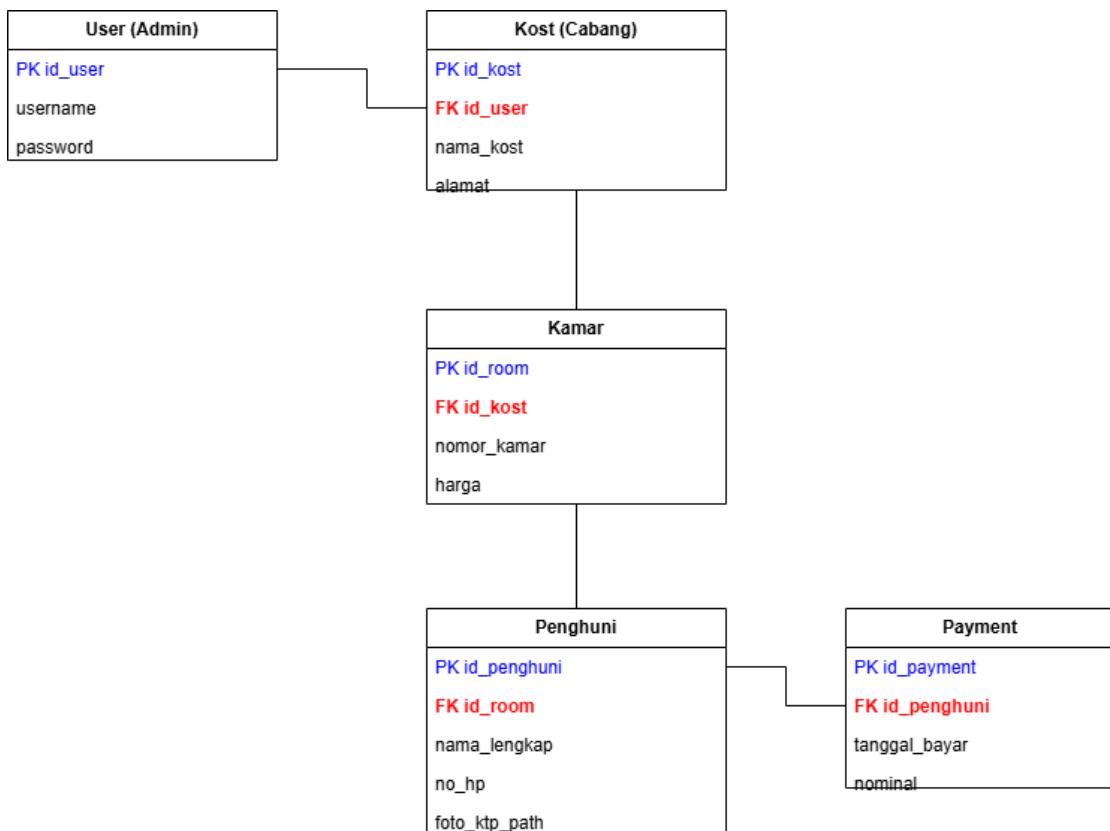
2. FlowChart



3. ERD

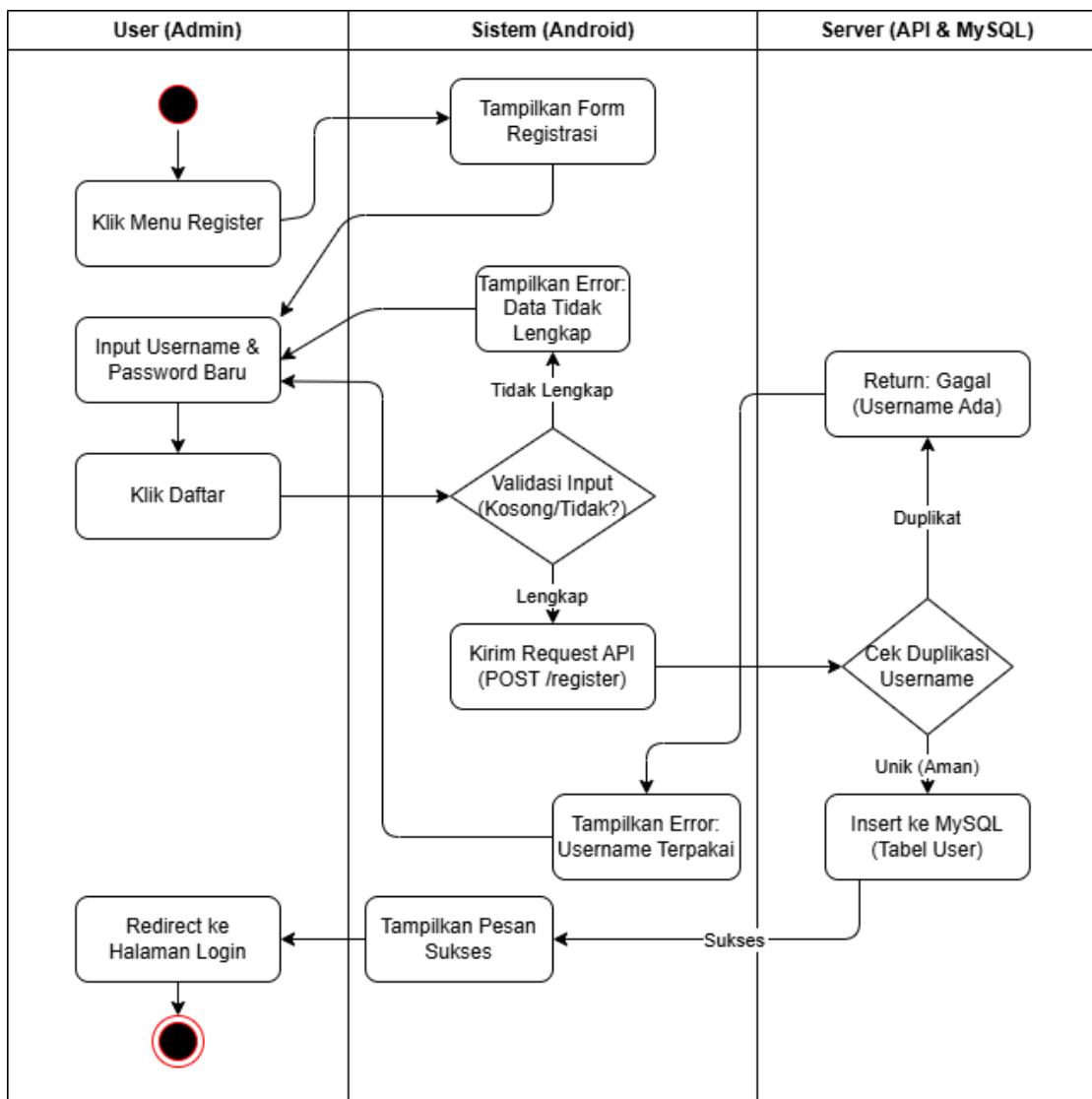


4. RAT

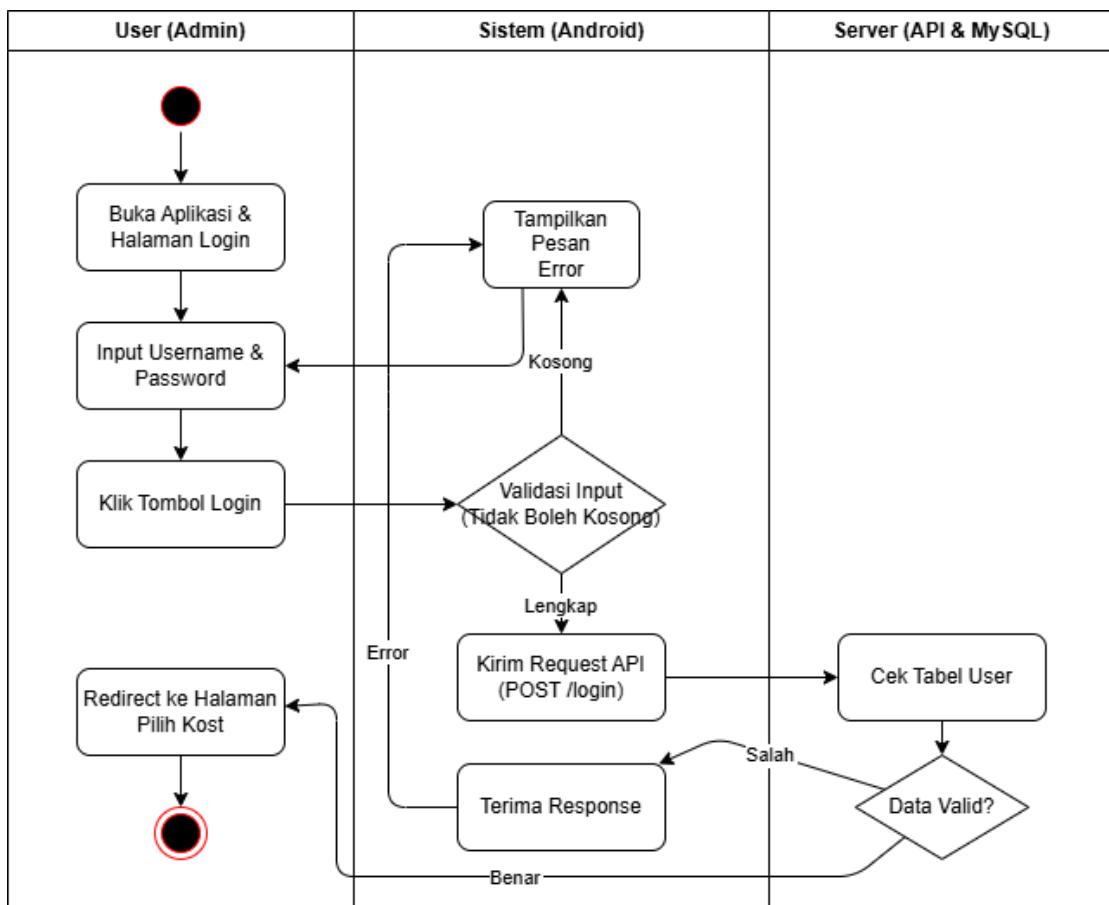


5. Activity Diagram

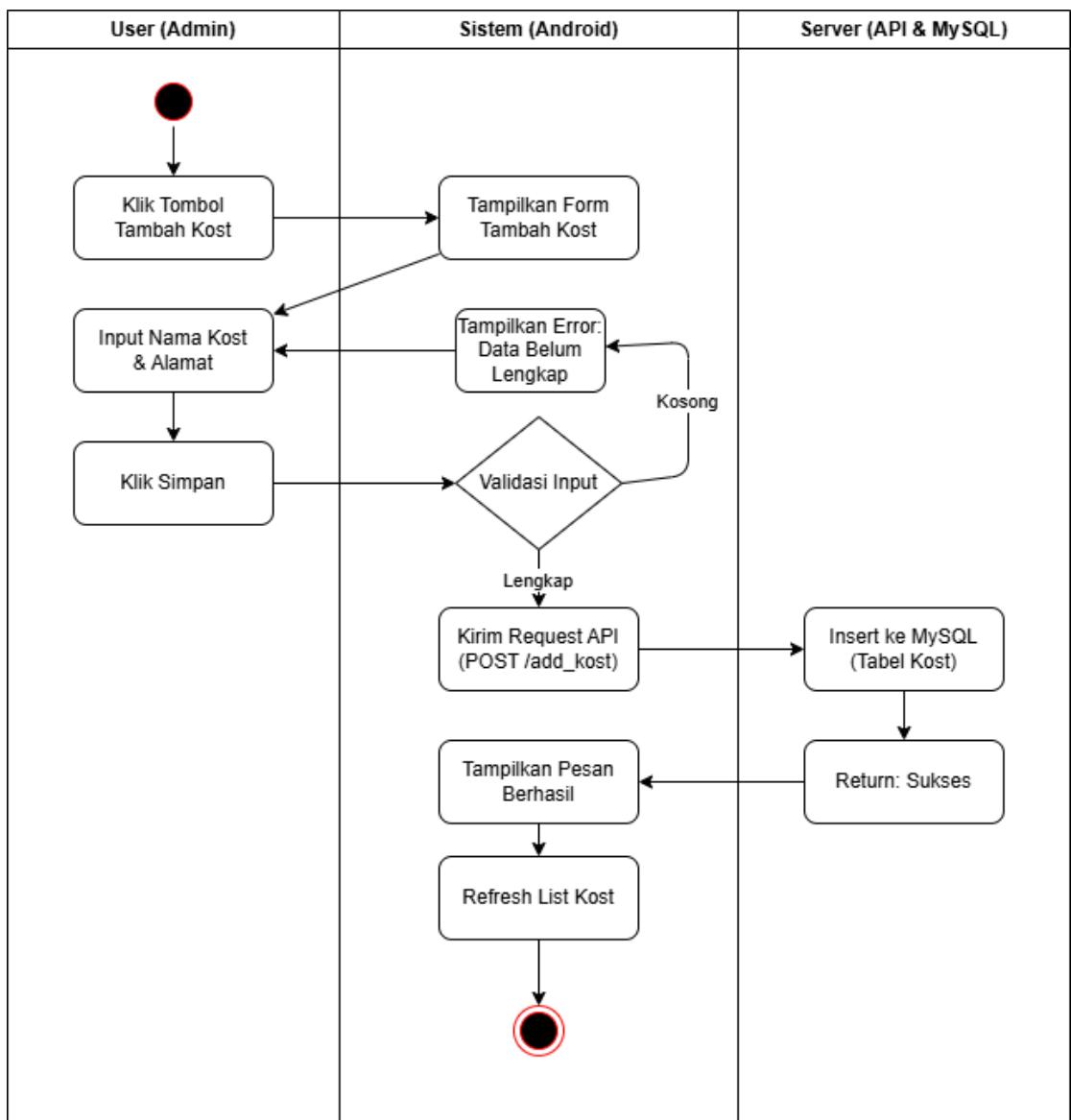
Register



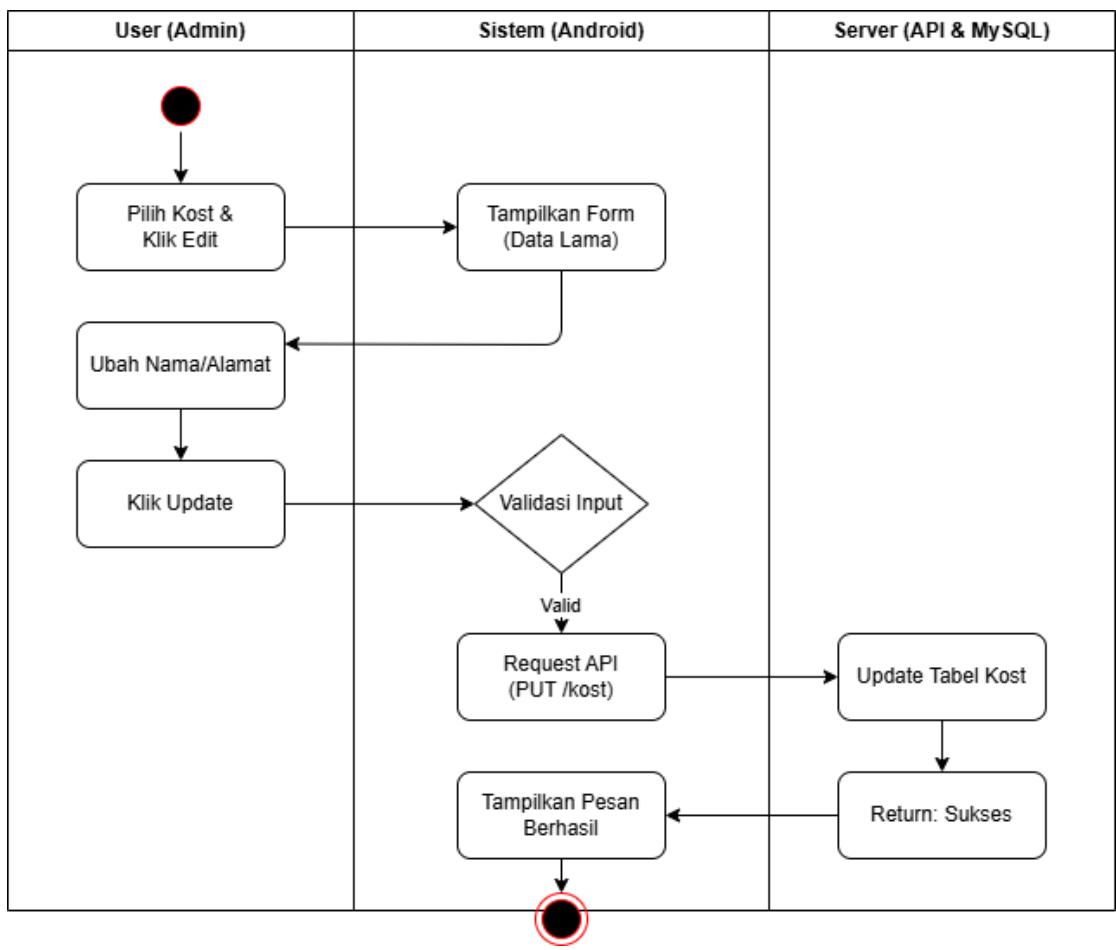
Login



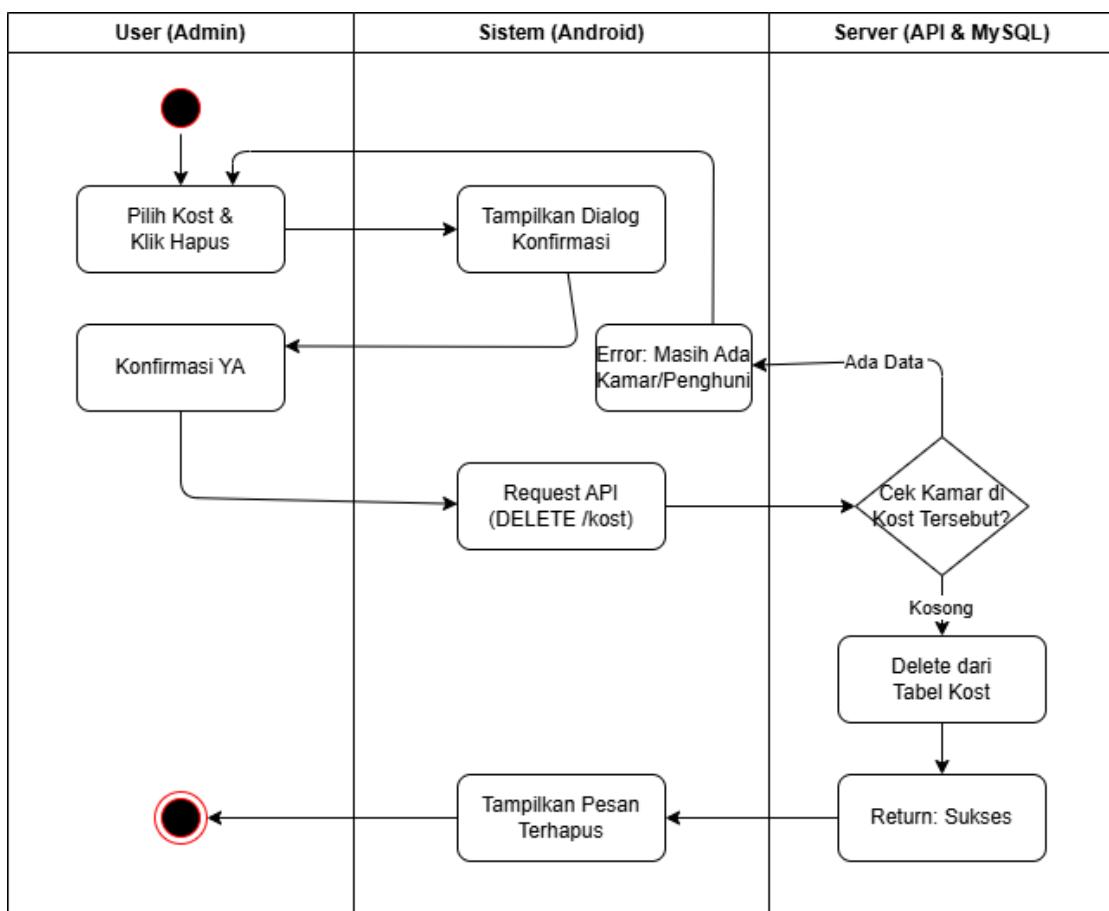
Tambah Data Kost



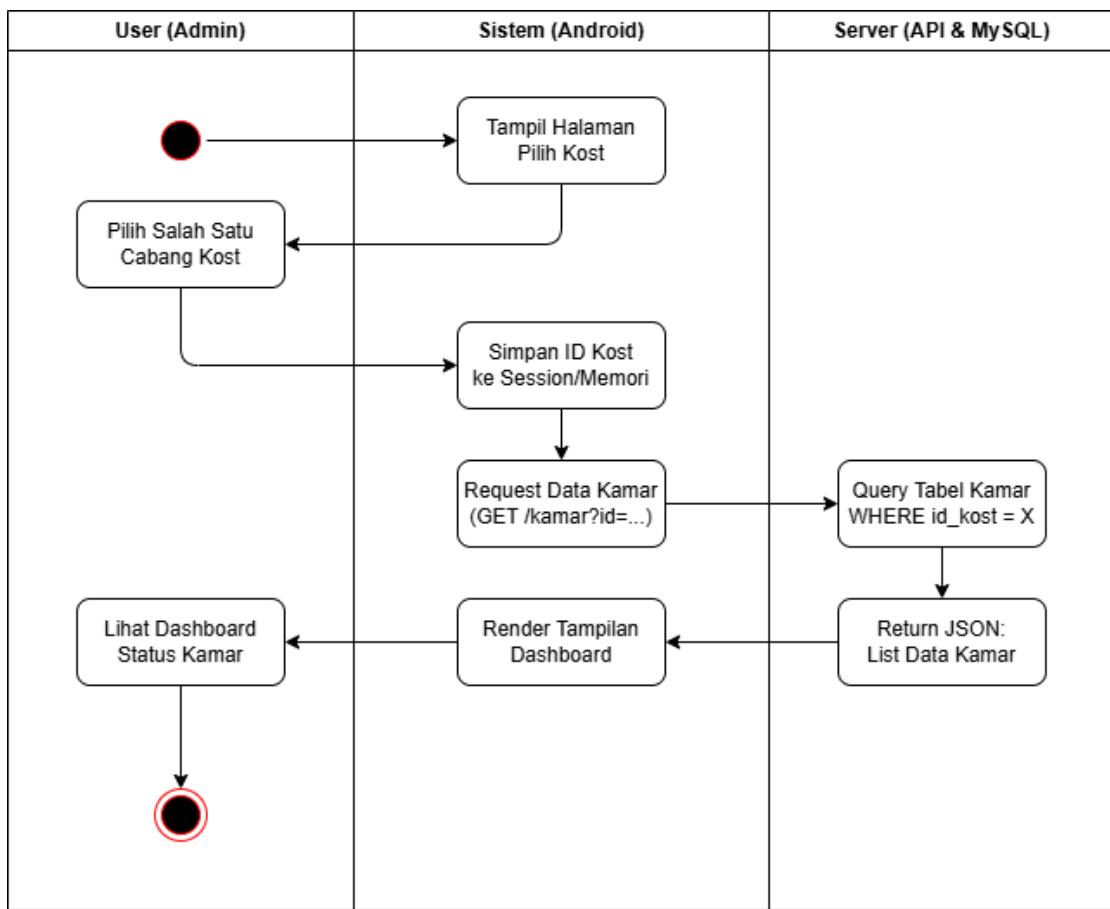
Edit Kost



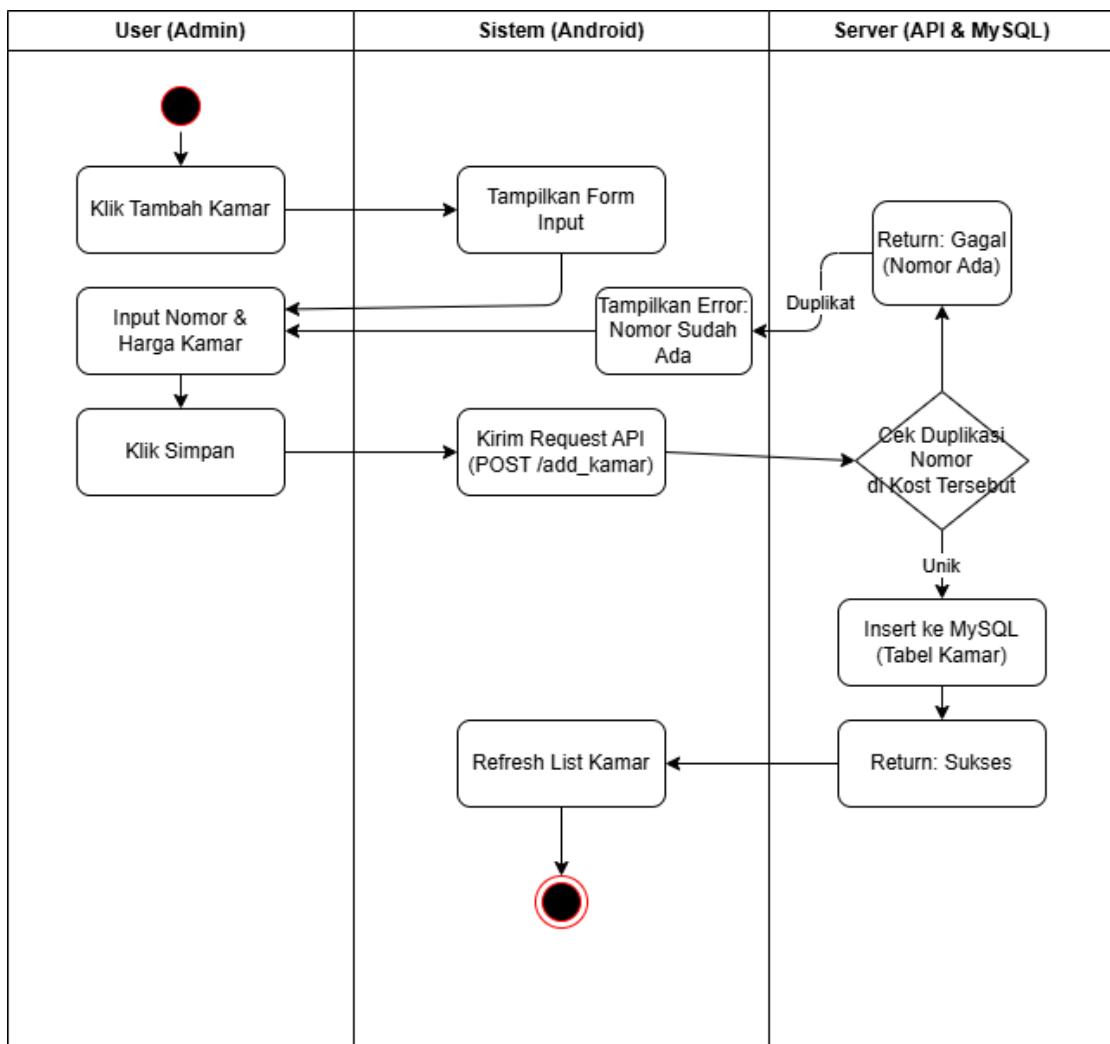
Hapus Kost



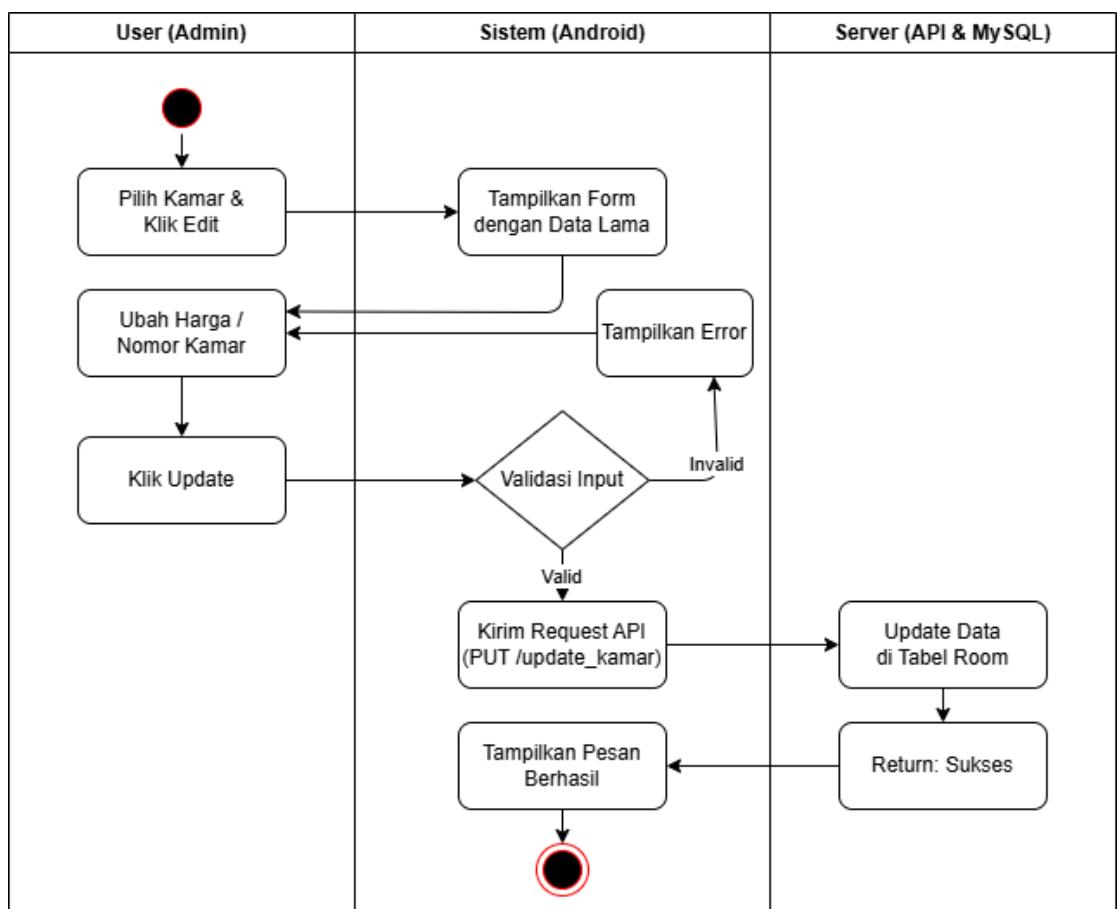
Pilih Kost



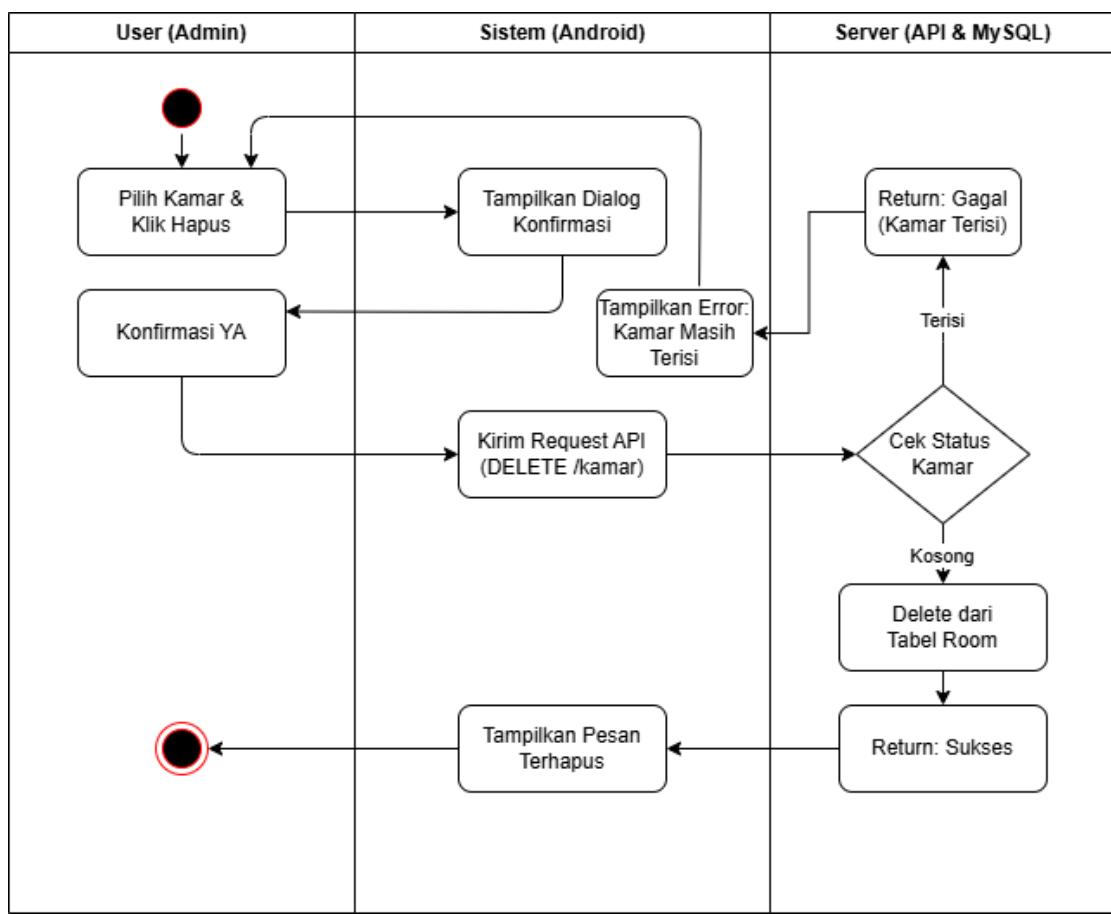
Tambah kamar



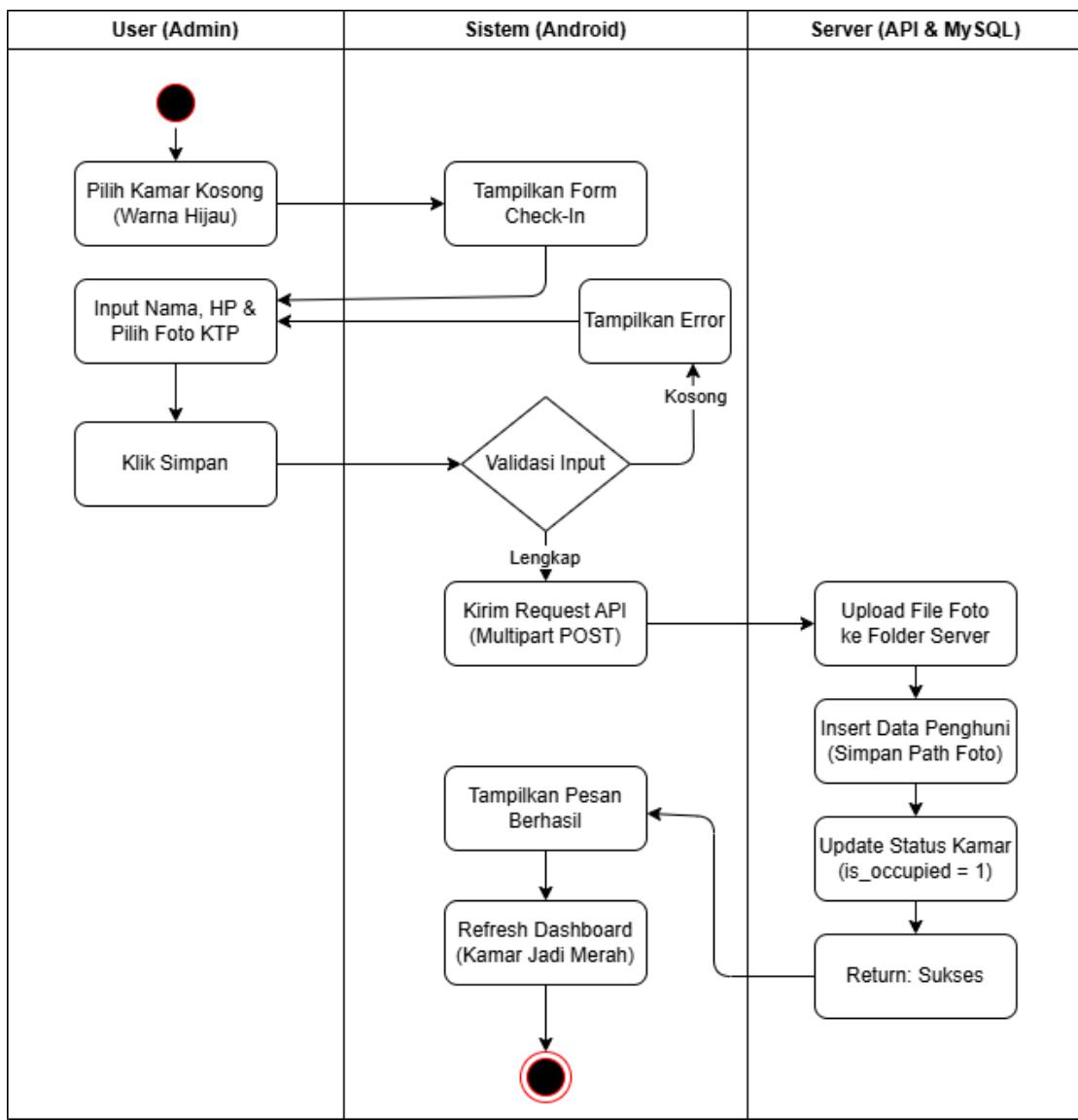
Edit Kamar



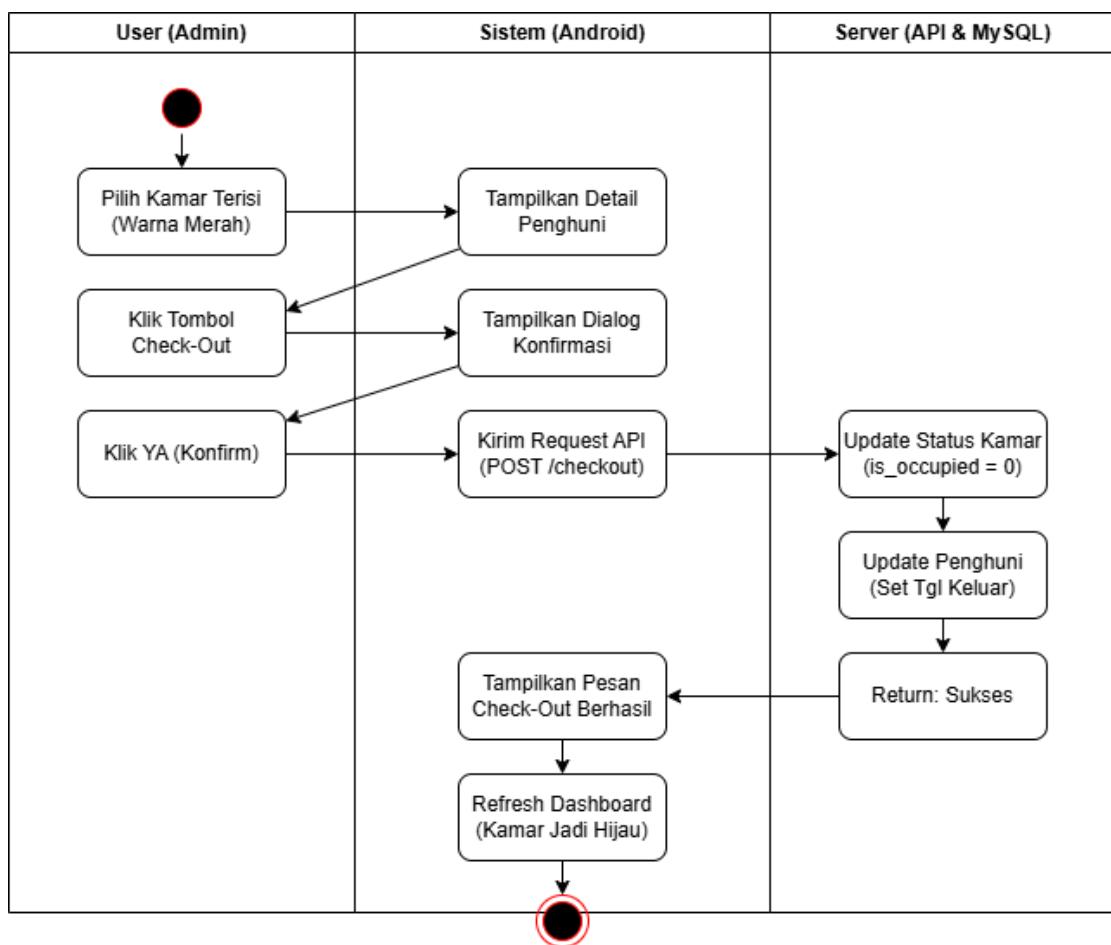
Delete Kamar



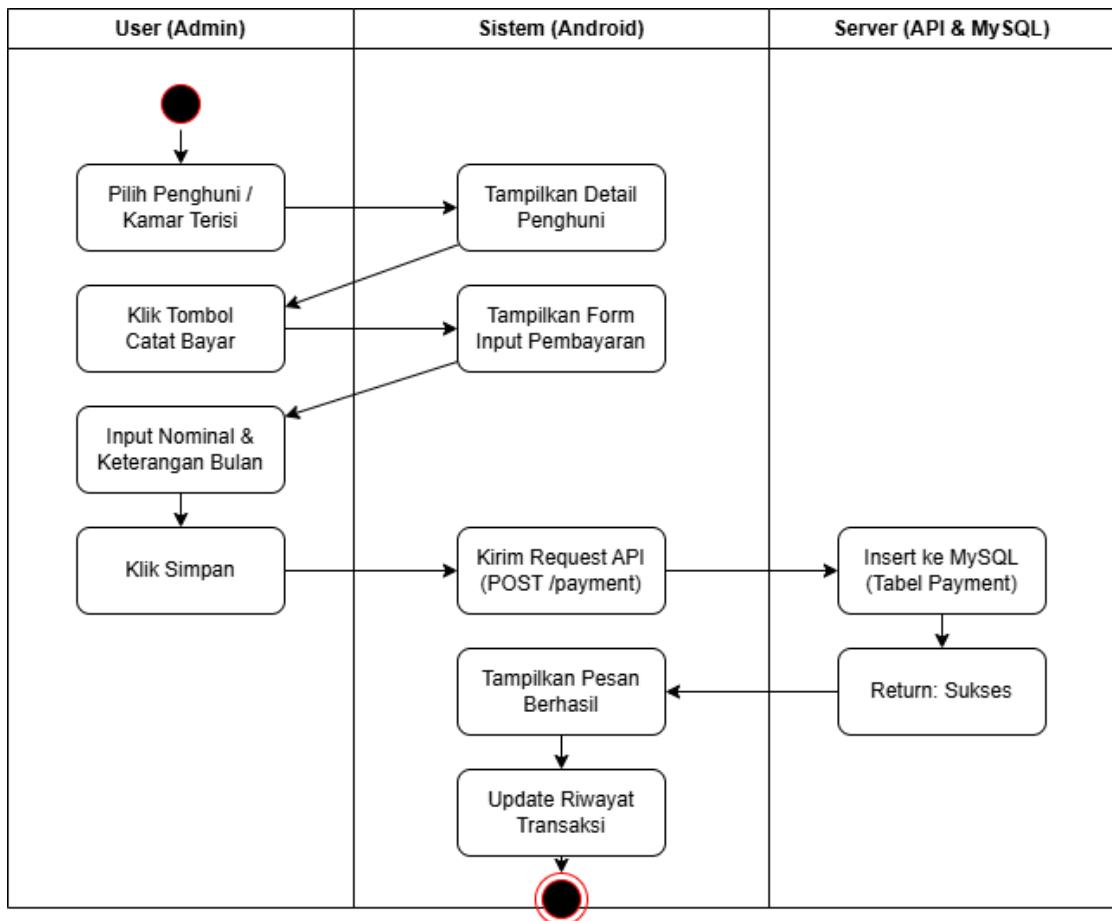
Check-In



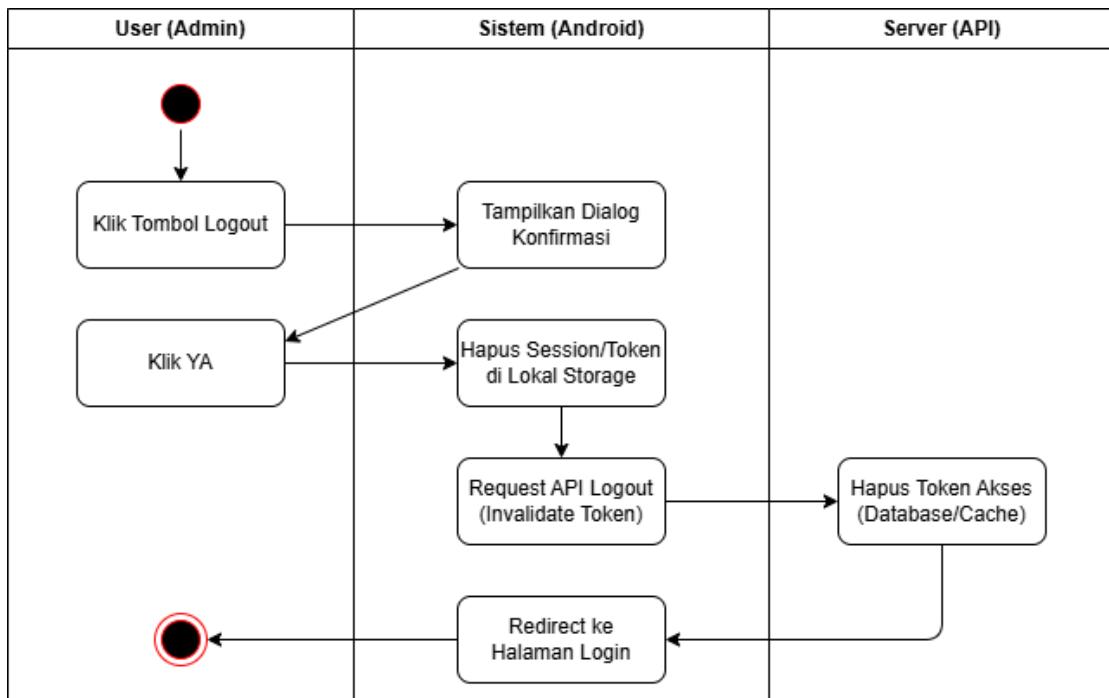
Check-Out



Payment



Logout



6. Kamus Data

User

Nama Atribut	Tipe Data	Ukuran	Constraint/Keterangan	Kunci
id_user	INT	11	AUTO_INCREMENT, NOT NULL	PK
username	VARCHAR	50	NOT NULL, UNIQUE	
password	CHAR	64	NOT NULL, Disimpan dalam bentuk Hash (SHA-256)	

- **id_user:** Kolom identitas unik untuk admin. Menggunakan AUTO_INCREMENT yang berarti nilainya bertambah otomatis setiap kali ada admin baru (jika dikembangkan menjadi multi-admin).
- **username:** Nama pengguna yang digunakan untuk *login*. Harus bersifat UNIQUE (tidak boleh ada username kembar).
- **password:** Kode keamanan untuk masuk sistem. Disimpan dalam format *hashing* (bukan teks biasa) demi keamanan data.

Kost

Nama Atribut	Tipe Data	Ukuran	Keterangan / Constraint	Key
id_kost	INT	11	AUTO_INCREMENT, Not Null	PK
id_user	INT	11	Merujuk ke Tabel User	FK
nama_kost	VARCHAR	100	Not Null (Contoh: "Kost Mawar")	
alamat	TEXT	-	Not Null	

- id_kost: ID unik (Primary Key) untuk mengidentifikasi setiap cabang kost yang terdaftar dalam sistem.
- id_user: Foreign Key yang menghubungkan data kost dengan tabel User (Admin). Kolom ini berfungsi untuk menandai siapa pemilik sah dari cabang kost tersebut.
- nama_kost: Nama identitas bangunan kost (Contoh: "Kost Putri Mawar", "Kost Putra Petir") untuk membedakan antar cabang.
- alamat: Informasi lengkap mengenai lokasi fisik bangunan kost.

Room (Kamar)

Nama Atribut	Tipe Data	Ukuran	Keterangan / Constraint	Key
id_room	INT	11	AUTO_INCREMENT, Not Null	PK
id_kost	INT	11	Merujuk ke Tabel Kost	FK
nomor_kamar	VARCHAR	10	Not Null (Contoh: "A1")	
harga	INT	11	Not Null	
is_occupied	TINYINT	1	Default 0 (0=Kosong, 1=Terisi)	

- id_kamar: ID unik untuk setiap kamar.
- id_kost: Foreign Key yang merujuk ke tabel **Kost**. Kolom ini sangat krusial dalam sistem *Multi-Kost* karena berfungsi untuk mengelompokkan kamar agar sistem tahu kamar ini berada di gedung/cabang yang mana.
- nomor_kamar: Label nomor atau nama kamar. Bersifat UNIQUE untuk mencegah duplikasi nomor kamar.

- harga: Harga sewa per bulan dalam Rupiah. Disimpan sebagai Integer.
- is_occupied: Penanda status hunian. Nilai 0 (False) berarti Kosong (Warna Hijau di Dashboard), dan nilai 1 (True) berarti Terisi (Warna Merah di Dashboard).

Penghuni

Nama Atribut	Tipe Data	Ukuran	Constraint/Keterangan	Kunci
id_penghuni	INT	11	AUTO_INCREMENT, NOT NULL	PK
id_kamar	INT	11	NOT NULL, Merujuk ke Tabel ROOM, ON DELETE CASCADE	FK
nama_lengkap	VARCHAR	100	NOT NULL	
no_hp	VARCHAR	15	NOT NULL	
foto_ktp_path	TEXT	-	NOT NULL, Menyimpan Alamat File (URI)	
tanggal_masuk	DATETIME	-	NOT NULL, Default CURRENT_TIMESTAMP	

- id_penghuni: ID unik untuk setiap data penghuni.
- id_kamar: *Foreign Key* yang menghubungkan penghuni dengan kamar yang ditempatinya.
- nama_lengkap: Nama penyewa sesuai KTP.
- foto_ktp_path: Menyimpan path atau URL file foto KTP yang tersimpan di server.
- tanggal_masuk: Mencatat kapan penghuni tersebut mulai *Check-In*.

Payment

Nama Atribut	Tipe Data	Ukuran	Constraint/Keterangan	Kunci
id_payment	INT	11	AUTO_INCREMENT, NOT NULL	PK
id_penghuni	INT	11	NOT NULL, Merujuk ke Tabel TENANT	FK
tanggal_bayar	DATETIME	-	NOT NULL, Tanggal Transaksi	
nominal	INT	11	NOT NULL, Nilai > 0	
keterangan	TEXT	-	Opsional (Contoh: "Lunas Januari")	

- id_payment: ID unik untuk setiap transaksi pembayaran.
- id_penghuni: *Foreign Key* yang menunjukkan siapa penghuni yang melakukan pembayaran tersebut.
- tanggal_bayar: Waktu transaksi dicatat.
- nominal: Jumlah uang yang dibayarkan.
- keterangan: Catatan tambahan terkait pembayaran (misalnya periode bulan yang dibayar).

Appendix C: Issues List

Berikut adalah daftar fitur atau kemampuan sistem yang dikecualikan (Out of Scope) dalam pengembangan versi 1.0 proyek Tugas Akhir ini:

1. Ketergantungan Jaringan Karena menggunakan arsitektur Client-Server, aplikasi memerlukan koneksi ke server XAMPP agar dapat berfungsi normal (Login, Simpan Data). Jika server mati atau jaringan terputus, aplikasi tidak dapat memproses transaksi baru.
2. Fitur Sinkronisasi (Keunggulan) Berbeda dengan versi *offline*, sistem ini mendukung multi-device. Admin A bisa menginput pembayaran di HP 1, dan Admin B bisa melihat datanya di HP 2 secara instan.
3. Keamanan Data Jika perangkat HP hilang atau aplikasi terhapus, data tetap aman tersimpan di database server (MySQL).