

---

# **Software Requirement**

## **Specification**

### **Document**

---

<Aplikasi Manajemen Inventaris Mobil  
Showroom>

<Salmaa Rifhani Rayyan>

Document ID	SRS-SHOWROOMOBIL-V1.0
Version Number	1.0
Issue Date	December 07, 2025
Classification	Internal – Project Akhir PAM

## **Copyright Notice**

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

All Rights Reserved

The information contained in this document is the property of COMPANYNAME. 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 COMPANYNAME. 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 COMPANYNAME. COMPANYNAME reserves the right to make changes to any products or services described in this document at any time without notice. COMPANYNAME 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)
12/01/2025	0.1	Penyusunan SRS dengan room database	Salmaa
04/12/2025	0.2	Revisi Usecase	Salmaa
04/12/2025	0.3	Revisi Flowchart	Salmaa
09/12/2025	1.0	Revisi section 1.6 – Penambahan akronim dan singkatan terkait node.js	Salmaa
09/12/2025	1.1	Revisi section 2.4 – Operating Environment	Salmaa
09/12/2025	1.2	Revisi Section 2.5 - Design Constraints	Salmaa
09/12/2025	1.3	Revisi Section 4.2 - Hardware Interfaces (perubahan SQLite ke MySQL)	Salmaa
09/12/2025	1.4	Revisi Section 4.3 - Software Interfaces (update ke Node.js backend)	Salmaa
09/12/2025	1.5	Revisi Section 4.4 - Communications Interfaces	Salmaa
09/12/2025	1.6	Revisi Section 3 - System Features (penambahan keterangan API di semua REQ)	Salmaa
09/12/2025	1.7	Revisi Section 5.1 - Performance Requirements (penambahan API requirements)	Salmaa
09/12/2025	1.8	Revisi Section 5.3 - Security Requirements (penambahan JWT & HTTPS)	Salmaa
09/12/2025	1.9	Revisi Section 6.1 - Appendix Glossary (penambahan istilah baru)	Salmaa

18/12/2025	1.10	Revisi dokumen SRS menggunakan PHP dan XAMPP (phpMyAdmin)	Salmaa
21/12/2025	1.11	Revisi kamus data – penambahan tabel merk untuk manajemen merk mobil	Salmaa
21/12/2025	1.12	Revisi ERD dan RAT – menambahkan tabel merk	

Reviewed By (Customer)	Signature	Date

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

## Sign Off

Prepared By	Acknowledged By
<Name>	<Salmaa Rifhani Rayyan>
Title: <Position>	Title: <Mahasiswa>
COMPANYNAME	<Universitas Muhammadiyah Yogyakarta>
Date:	Date: 4 Desember 2025

Accepted By	Accepted By
<Name>	<Pascal Pahlevi Pasha>
Title: <Position>	Title: <Instruktur>
<Customer Company Name >	<Universitas Muhammadiyah Yogyakarta>
Date:	Date: 10 Desember 2025

## **Daftar Isi**

<b>1</b>	<b>Pendahuluan.....</b>	<b>10</b>
1.1	Tujuan .....	10
1.2	Konvensi Dokumen.....	10
1.3	Audiens dan Saran Pembacaan .....	10
1.4	Ruang Lingkup Proyek .....	11
1.5	Referensi .....	11
1.6	Akrонim dan Singkatan .....	12
<b>2</b>	<b>Deskripsi Umum .....</b>	<b>12</b>
2.1	Perspektif Produk.....	12
2.2	Fitur Produk .....	13
2.2.1	Autentikasi .....	13
2.2.2	Manajemen Merk Mobil .....	13
2.2.3	Manajemen Data Mobil .....	13
2.2.4	Manajemen Stok .....	13
2.2.5	Validasi Data.....	13
2.3	Kelas dan Karakteristik Pengguna .....	14
2.4	Lingkungan Operasi .....	14
2.4.1	Mobile Application (Android) .....	14
2.4.2	Backend Server .....	15
2.4.3	Database .....	15
2.4.4	Komunikasi & Network .....	15
2.4.5	Arsitektur Sistem .....	15
2.4.6	Development Tools.....	16
2.5	Batasan Desain dan Implementasi .....	16
2.6	Dokumentasi Pengguna.....	17
2.7	Asumsi dan Ketergantungan .....	18
<b>3</b>	<b>System Features.....</b>	<b>18</b>

3.1	Autentikasi - Login .....	18
3.1.1	Deskripsi dan Prioritas .....	18
3.1.2	Stimulus/Response Sequences .....	18
3.1.3	Functional Requirements .....	19
3.2	Autentikasi – Logout.....	19
3.2.1	Deskripsi dan Prioritas .....	19
3.2.2	Stimulus/Response Sequences .....	19
3.2.3	Functional Requirements .....	19
3.3	Manajemen Merk Mobil - Tambah Merk .....	20
3.3.1	Deskripsi dan Prioritas .....	20
3.3.2	Stimulus/Response Sequences .....	20
3.3.3	Functional Requirements .....	20
3.4	Manajemen Merk Mobil - Edit Merk.....	20
3.4.1	Deskripsi dan Prioritas .....	20
3.4.2	Stimulus/Response Sequences .....	21
3.4.3	Functional Requirements .....	21
3.5	Manajemen Merk Mobil - Hapus Merk .....	21
3.5.1	Deskripsi dan Prioritas .....	21
3.5.2	Stimulus/Response Sequences .....	21
3.5.3	Functional Requirements .....	22
3.6	Manajemen Data Mobil - Tambah Mobil .....	22
3.6.1	Deskripsi dan Prioritas .....	22
3.6.2	Stimulus/Response Sequences .....	22
3.6.3	Functional Requirements .....	22
3.7	Manajemen Data Mobil - Lihat Detail Mobil .....	23
3.7.1	Deskripsi dan Prioritas .....	23
3.7.2	Stimulus/Response Sequences .....	23

3.7.3	Functional Requirements .....	23
3.8	Manajemen Data Mobil - Edit Mobil.....	24
3.8.1	Deskripsi dan Prioritas.....	24
3.8.2	Stimulus/Response Sequences .....	24
3.8.3	Functional Requirements .....	24
3.9	Manajemen Data Mobil - Hapus Mobil .....	25
3.9.1	Deskripsi dan Prioritas.....	25
3.9.2	Stimulus/Response Sequences .....	25
3.9.3	Functional Requirements .....	25
3.10	Manajemen Stok - Tambah Stok.....	25
3.10.1	Deskripsi dan Prioritas.....	25
3.10.2	Stimulus/Response Sequences .....	25
3.10.3	Functional Requirements .....	26
3.11	Manajemen Stok - Kurangi Stok.....	26
3.11.1	Deskripsi dan Prioritas.....	26
3.11.2	Stimulus/Response Sequences .....	26
3.11.3	Functional Requirements .....	26
<b>4</b>	<b>Kebutuhan Interface Eksternal.....</b>	<b>27</b>
4.1	User Interfaces .....	27
4.1.1	Halaman Login.....	27
4.1.2	Halaman Dashboard.....	27
4.1.3	Halaman List Mobil .....	28
4.1.4	Halaman Detail Mobil.....	28
4.1.5	Halaman Form Mobil (Tambah/Edit) .....	29
4.1.6	Pop-up Dialog .....	29
4.1.7	Toast Notification .....	29
4.2	Hardware Interfaces .....	29

4.3	Software Interfaces .....	30
4.3.1	MySQL Database.....	30
4.3.2	PHP Backend .....	30
4.3.3	JWT Authentication .....	31
4.3.4	Password Hashing.....	31
4.3.5	Retrofit (Android - HTTP Client).....	31
4.3.6	Android SharedPreferences .....	31
4.3.7	JSON (Data Format) .....	32
4.4	Communications Interfaces.....	32
<b>5</b>	<b>Kebutuhan Non-Fungsional.....</b>	<b>32</b>
5.1	Performance Requirements .....	32
5.1.1	Response Time.....	32
5.1.2	Resource Usage.....	33
5.2	Safety Requirements .....	33
5.2.1	Authentication.....	33
5.2.2	Data Protection.....	33
5.2.3	Access Control .....	33
5.3	Security Requirements .....	33
5.3.1	Usability.....	33
5.3.2	Reliability.....	33
5.3.3	Maintainability.....	34
5.3.4	Portability.....	34
<b>6</b>	<b>Other Requirements .....</b>	<b>34</b>
6.1	Appendix A: Glossary (Glosarium) .....	34
6.2	Appendix B: Analysis Models .....	35
6.3	Appendix C: Issues List .....	54

## 1 Pendahuluan

### 1.1 Tujuan

Dokumen Software Requirement Specification (SRS) ini dibuat untuk mendefinisikan kebutuhan fungsional dan non-fungsional dari Aplikasi Manajemen Inventaris Mobil Showroom. Aplikasi ini dirancang untuk memudahkan admin showroom dalam mengelola data stok mobil secara efisien melalui platform mobile Android dengan backend PHP dan database MySQL.

Dokumen ini mencakup:

- Spesifikasi lengkap fitur aplikasi
- Kebutuhan teknis dan fungsional
- *Interface* pengguna dan sistem
- *Constraint* dan batasan pengembangan

### 1.2 Konvensi Dokumen

Dokumen ini menggunakan konvensi penulisan sebagai berikut:

- **Bold (Tebal)**: Istilah penting atau nama fitur
- *Italic (Miring)*: Istilah teknis atau bahasa asing
- Prioritas fitur: HIGH (Tinggi), MEDIUM (Sedang), LOW (Rendah)

### 1.3 Audiens dan Saran Pembacaan

Dokumen ini ditujukan untuk:

Audiens	Bagian yang Relevan
Dosen, Instruktur, Asisten	Semua bagian untuk evaluasi
Tester	Bagian 3 (Functional Requirements)
Admin Showroom (User)	Bagian 2 (Deskripsi Umum) dan Bagian 4.1 (User Interface)

Saran urutan pembacaan:

1. Baca bagian 1 dan 2 untuk memahami overview sistem
2. Lanjutkan ke bagian 3 untuk detail fitur
3. Baca bagian 4 dan 5 untuk spesifikasi teknis

#### 1.4 Ruang Lingkup Proyek

Aplikasi Manajemen Inventaris Mobil Showroom adalah aplikasi mobile berbasis Android yang menggunakan arsitektur Client-Server. Aplikasi ini terhubung dengan database MySQL melalui RESTful API berbasis PHP untuk pengelolaan data stok mobil secara real-time.

##### **Tujuan Aplikasi:**

- Mendigitalisasi proses pencatatan inventaris mobil
- Memudahkan pengelolaan stok mobil
- Menyediakan sistem autentikasi untuk keamanan data
- Mengurangi kesalahan pencatatan manual
- Mencegah duplikasi data mobil

##### **Manfaat:**

- Efisiensi waktu dalam pengelolaan data mobil
- Akurasi data inventaris yang lebih baik
- Kemudahan akses informasi stok
- Pencegahan duplikasi data
- Pengelolaan stok yang lebih terkontrol dengan sistem tambah/kurangi stok

##### **Batasan:**

- Aplikasi hanya untuk platform Android
- Menggunakan arsitektur Client-Server (Android + PHP + MySQL)
- Memerlukan koneksi internet untuk mengakses API
- Hanya untuk penggunaan single-user (1 admin per device)
- Tidak ada fitur registrasi user baru (admin sudah terdaftar)
- Token autentikasi tidak akan expired secara otomatis (hanya saat logout manual)

#### 1.5 Referensi

- **Entity Relationship Diagram (ERD)**
- **Flowchart Aplikasi**
- **Use Case Diagram**
- **Activity Diagram**
- **Kamus Data**
- **RAT (Relasi Antar Tabel)**

- **Android Developer Documentation:** <https://developer.android.com/>

## 1.6 Akronim dan Singkatan

Akronim	Kepanjangan
SRS	Software Requirement Specification
CRUD	Create, Read, Update, Delete
ERD	Entity Relationship Diagram
RAT	Relasi Antar Tabel
PK	Primary Key
FK	Foreign Key
JWT	JSON Web Token
UI	User Interface
UX	User Experience
API	Application Programming Interface
FAB	Floating Action Button
HTTP/HTTPS	Hypertext Transfer Protocol (Secure)
REST	Representational State Transfer
JSON	JavaScript Object Notation
MySQL	My Structured Query Language
XAMPP	Cross-Platform Apache MySQL PHP Perl
PDO	PHP Data Objects
SHA-256	Secure Hash Algorithm 256-bit

## 2 Deskripsi Umum

### 2.1 Perspektif Produk

Aplikasi Manajemen Inventaris Mobil Showroom adalah aplikasi mobile berbasis Android yang menggunakan arsitektur Client-Server. Aplikasi ini berinteraksi dengan backend server melalui RESTful API.

Arsitektur Sistem:

Android App (Client) → RESTful API (PHP) → Database MySQL

Komponen Sistem:

- Frontend: Aplikasi Android (Kotlin + Jetpack Compose)

- Backend: REST API berbasis PHP
- Database: MySQL (dikelola melalui phpMyAdmin)

## 2.2 Fitur Produk

Aplikasi ini memiliki fitur utama sebagai berikut:

### 2.2.1 Autentikasi

- Login dengan email dan password
- Logout dengan konfirmasi
- Token-based authentication (JWT)
- Token disimpan di database dan SharedPreferences

### 2.2.2 Manajemen Merk Mobil

- Tambah merk mobil baru ke tabel merk (dengan validasi duplikasi)
- Edit nama merk di tabel merk dan update semua mobil terkait (validasi duplikasi)
- Hapus merk dari tabel merk (menghapus semua mobil dengan merk tersebut)
- Lihat daftar merk dari tabel merk dalam bentuk card
- Tambah/edit merk menggunakan pop-up dialog

### 2.2.3 Manajemen Data Mobil

- Tambah mobil baru dengan validasi duplikasi
- Lihat daftar mobil per merk
- Lihat detail lengkap mobil
- Edit data mobil dengan validasi duplikasi
- Hapus mobil (dengan konfirmasi)

### 2.2.4 Manajemen Stok

- Tambah stok mobil (+1) dengan konfirmasi
- Kurangi stok mobil (-1) dengan konfirmasi
- Validasi stok (tombol kurangi disabled jika stok = 0)
- Update stok real-time

### 2.2.5 Validasi Data

- Validasi duplikasi merk
- Validasi duplikasi mobil (nama + merk + warna + tahun)
- Validasi input wajib diisi
- Validasi login

- Validasi tahun mobil (minimal 2000)

### 2.3 Kelas dan Karakteristik Pengguna

Aplikasi ini memiliki satu kelas pengguna:

Karakteristik	Deskripsi
Peran	Mengelola inventaris mobil showroom
Frekuensi Penggunaan	Harian (setiap ada transaksi atau update stok)
Kemampuan Teknis	Dasar - dapat mengoperasikan smartphone Android
Tingkat Pendidikan	Minimal SMA/SMK
Keamanan	Memiliki akun login yang sudah terdaftar di database
Fungsi Utama	CRUD merk mobil, CRUD data mobil, update stok

Skenario Penggunaan Tipikal:

1. Admin login setiap pagi untuk memulai kerja
2. Admin cek stok mobil yang tersedia per merk
3. Admin update stok jika ada mobil baru masuk atau terjual
4. Admin tambah data mobil baru jika ada unit baru
5. Admin logout saat selesai bekerja

### 2.4 Lingkungan Operasi

Aplikasi Manajemen Inventaris Mobil Showroom terdiri dari tiga komponen utama yang saling terintegrasi: Mobile Application (Android), Backend Server (PHP), dan Database (MySQL).

#### 2.4.1 Mobile Application (Android)

Aspek	Spesifikasi
Platform	Android 7.0 (API Level 24) atau lebih baru
Bahasa	Kotlin
UI Framework	Jetpack Compose
Minimum Hardware	2 GB RAM, 50 MB storage
HTTP Client	Retrofit 2.9.0+
Local Storage	SharedPreferences (untuk token)
Koneksi Internet	WAJIB (tidak support offline)

#### **2.4.2 Backend Server**

Aspek	Spesifikasi
Bahasa	PHP7.4+
Server	Apache (XAMPP)
JWT Library	firebase/php-jwt
Authentication	JWT (JSON Web Token)
Password Hashing	Password_hash() dengan bcrypt
API Type	RESTful API (JSON format)
Database Connection	PDO (PHP Data Objects)
Deployment	Local (XAMPP)

#### **2.4.3 Database**

Aspek	Spesifikasi
DBMS	MySQL 8.0+
Port	3306 (default)
Storage Engine	phpMyAdmin (XAMPP)
Deployment	XAMPP (local)

Tabel Database:

- user – data admin
- token – token JWT untuk autentikasi
- merk – data merk mobil
- mobil – data mobil
- stok – data stok mobil

#### **2.4.4 Komunikasi & Network**

Aspek	Spesifikasi
Protocol	HTTP (development), HTTPS (production)
Data Format	JSON
API Base URL	Development: http://10.0.2.2/showroom-backend/api/ (Emulator)
Request Method	GET, POST, PUT, DELETE
Koneksi Internet	Mobile app WAJIB online untuk semua operasi

#### **2.4.5 Arsitektur Sistem**

- a) Komponen Utama:

- Android App (Kotlin + Jetpack Compose + Retrofit)
  - PHP Backend (REST API + JWT Authentication)
  - MySQL Database (phpMyAdmin)
- b) Alur Kerja:
- Admin input data di Android App
  - App kirim request HTTP ke PHP API (dengan token di header)
  - PHP validasi token di database
  - PHP query MySQL (CRUD data)
  - PHP kirim response JSON ke App
  - App update tampilan
- c) Komunikasi:
- Android ↔ PHP: HTTP Request/Response (JSON)
  - PHP ↔ MySQL: SQL Query (PDO)

#### 2.4.6 Development Tools

Komponen	Tools
Mobile Development	Android Studio
Backend Development	Visual Studio Code, Postman (API testing)
Database Management	phpMyAdmin (XAMPP)
Version Control	Git
Backend Server	XAMPP (Apache + MySQL)

#### 2.5 Batasan Desain dan Implementasi

a) **Constraint Teknis:**

- Platform: Android only (min SDK 27)
- Backend: PHP + Apache (XAMPP)
- Database: MySQL 8.0+
- Autentikasi: JWT (library firebase/php-jwt)
- Password: Hashed dengan password\_hash() (bcrypt)
- HTTP Client: Retrofit untuk API calls
- Koneksi: Wajib internet (tidak ada offline mode)
- Data Format: JSON

b) **Constraint Fungsional:**

- Validasi Duplikasi:
  - Merk harus unik
  - Mobil unik berdasarkan: nama + merk\_id + warna + tahun
- Stok:
  - Tidak boleh < 0
  - tombol kurangi disabled jika stok = 0
- Token: Disimpan di database (tabel token) dan SharedPreferences
- No Registration: Tidak ada fitur pendaftaran user baru, admin harus didaftarkan langsung di database
- Login: Tidak ada fitur registrasi, admin sudah terdaftar di database
- Token Expiry: Token hanya hilang saat logout manual (tidak ada auto-expired)

**c) Constraint Bisnis:**

- Target User: Hanya admin yang sudah terdaftar
- Data Storage: Tersimpan di server MySQL (bukan lokal device)
- Security: Admin harus login menggunakan email dan password setiap membuka aplikasi
- Session: Token tersimpan di SharedPreferences, tidak perlu login ulang sampai logout

## 2.6 Dokumentasi Pengguna

Dokumentasi yang akan disediakan:

1. **User Manual** (Panduan Pengguna)
  - Cara login dan logout
  - Cara mengelola merk mobil
  - Cara mengelola data mobil
  - Cara update stok
2. **Technical Documentation**
  - ERD (Entity Relationship Diagram)
  - Flowchart aplikasi
  - Use Case Diagram
  - Activity Diagram
  - RAT (Relasi Antar Tabel)

- Kamus Data

### 3. Installation Guide

- Cara install aplikasi
- Minimum requirements
- Setup XAMPP & database

## 2.7 Asumsi dan Ketergantungan

### Asumsi:

1. Admin sudah memiliki akun yang terdaftar di database (tidak perlu registrasi)
2. Device admin memiliki storage yang cukup untuk database
3. Admin paham cara operasi smartphone Android
4. XAMPP sudah terinstall dan Apache + MySQL aktif
5. Admin memiliki koneksi internet

### Ketergantungan:

1. Android OS: Min Android 7.0 (API 24)
2. XAMPP: Apache dan MySQL harus running
3. Koneksi Internet: Wajib untuk akses API

## 3 System Features

### 3.1 Autentikasi - Login

#### 3.1.1 Deskripsi dan Prioritas

Deskripsi: Admin dapat login ke aplikasi menggunakan email dan password yang sudah terdaftar.

Prioritas: HIGH (Wajib)

#### 3.1.2 Stimulus/Response Sequences

- Admin membuka aplikasi
- Sistem menampilkan halaman login
- Admin input email dan password
- Admin menekan tombol "Login"
- Sistem validasi credentials
- Jika valid: Sistem generate token JWT, simpan ke database, redirect ke dashboard
- Jika tidak valid: Sistem menampilkan pesan error "Email atau password salah"

### 3.1.3 Functional Requirements

**REQ-1.1:** Sistem harus menyediakan form login dengan field email dan password

**REQ-1.2:** Sistem harus memvalidasi bahwa email dan password tidak boleh kosong

**REQ-1.3:** Mobile app kirim HTTP POST ke /api/login.php dengan JSON {email, password}

**REQ-1.4:** Backend PHP query database tabel user untuk cek email dan password

**REQ-1.5:** Jika login berhasil:

- PHP generate token JWT
- PHP simpan token ke tabel token (kolom: user\_id, token, created\_at)
- PHP return response JSON: {success, token, user}
- Android simpan token ke SharedPreferences
- Android redirect ke Dashboard

**REQ-1.6:** Password tidak ditampilkan dalam plain text (di-mask dengan dot/asterisk)

## 3.2 Autentikasi – Logout

### 3.2.1 Deskripsi dan Prioritas

Deskripsi: Admin dapat logout dari aplikasi dengan aman.

Prioritas: HIGH (Wajib)

### 3.2.2 Stimulus/Response Sequences

1. Admin menekan tombol/menu "Logout"
2. Sistem menampilkan dialog konfirmasi: "Yakin ingin logout?"
3. Admin menekan "Ya"
4. Sistem hapus token dari database dan SharedPreferences
5. Sistem redirect ke halaman login

### 3.2.3 Functional Requirements

**REQ-2.1:** Sistem harus menyediakan tombol "Logout" di halaman dashboard

**REQ-2.2:** Sistem harus menampilkan dialog konfirmasi sebelum logout dengan pilihan "Ya" dan "Tidak"

**REQ-2.3:** Jika admin memilih "Tidak", sistem membatalkan proses logout

**REQ-2.4:** Jika "Ya":

- Mobile app kirim HTTP POST ke /api/logout.php dengan token di header
- PHP hapus token dari tabel token berdasarkan token yang dikirim
- Android hapus token dari SharedPreferences

- Android redirect ke Login Page

**REQ-2.5:** Setelah logout, admin harus login kembali untuk mengakses aplikasi

### 3.3 Manajemen Merk Mobil - Tambah Merk

#### 3.3.1 Deskripsi dan Prioritas

Deskripsi: Admin dapat menambahkan merk mobil baru.

Prioritas: HIGH (Wajib)

#### 3.3.2 Stimulus/Response Sequences

1. Admin di Dashboard, klik FAB (+)
2. Sistem menampilkan pop-up dialog input nama merk
3. Admin input nama merk
4. Admin klik "Simpan"
5. Sistem memvalidasi input dan cek duplikasi
6. Jika valid: Sistem simpan ke database dan tampilkan toast sukses
7. Jika duplikat: Sistem tampilkan error "Merk sudah ada!"

#### 3.3.3 Functional Requirements

**REQ-3.1:** Sistem menyediakan FAB (+) di Dashboard

**REQ-3.2:** Sistem tampilkan pop-up dialog dengan field input nama merk

**REQ-3.3:** Sistem validasi nama merk tidak boleh kosong

**REQ-3.4:** Mobile app kirim HTTP POST ke /api/merk/create.php dengan JSON {merk}

**REQ-3.5:** Backend PHP cek duplikasi: query tabel merk untuk cek apakah merk sudah ada (case-insensitive)

**REQ-3.6:** Jika merk sudah ada, return error: "Merk sudah ada!"

**REQ-3.7:** Jika valid:

- Backend INSERT merk ke tabel merk
- Return success, android tampilkan toast "Merk berhasil ditambahkan" dan refresh Dashboard

**REQ-3.8:** Admin dapat batalkan dengan klik "Batal"

### 3.4 Manajemen Merk Mobil - Edit Merk

#### 3.4.1 Deskripsi dan Prioritas

Deskripsi: Admin dapat mengubah nama merk mobil yang sudah ada.

Prioritas: MEDIUM

### 3.4.2 Stimulus/Response Sequences

1. Admin di Dashboard, klik tombol Edit pada card merk
2. Sistem tampilkan pop-up dialog dengan nama merk lama
3. Admin ubah nama merk
4. Admin menekan "Simpan"
5. Sistem validasi dan update database
6. Sistem menampilkan toast sukses

### 3.4.3 Functional Requirements

**REQ-4.1:** Sistem menyediakan tombol Edit pada setiap card merk

**REQ-4.2:** Sistem tampilkan pop-up dialog dengan field nama merk terisi data lama

**REQ-4.3:** Sistem validasi nama merk tidak boleh kosong

**REQ-4.4:** Mobile app kirim HTTP PUT ke /api/merk/update.php dengan JSON {id, merk\_baru}

**REQ-4.5:** Backend cek duplikasi: query tabel merk untuk cek apakah nama merk baru sudah digunakan merk lain

**REQ-4.6:** Jika sudah ada, return error: "Merk sudah ada!"

**REQ-4.7:** Jika valid:

- Backend UPDATE tabel merk: ubah nama\_merk WHERE id = id
- Return success, android tampilkan toast "Merk berhasil diperbarui" dan refresh Dashboard

**REQ-4.8:** Return success, Android tampilkan toast "Merk berhasil diperbarui" dan refresh Dashboard

## 3.5 Manajemen Merk Mobil - Hapus Merk

### 3.5.1 Deskripsi dan Prioritas

Deskripsi: Admin dapat menghapus merk mobil beserta semua mobil yang terkait.

Prioritas: MEDIUM

### 3.5.2 Stimulus/Response Sequences

1. Admin di Dashboard, klik tombol Hapus pada card merk
2. Sistem menampilkan dialog konfirmasi
3. Admin menekan "Ya"
4. Sistem hapus semua mobil dengan merk tersebut
5. Sistem refresh Dashboard

### 3.5.3 Functional Requirements

**REQ-5.1:** Sistem harus menyediakan tombol Hapus pada setiap card merk

**REQ-5.2:** Sistem harus menampilkan dialog konfirmasi: "Yakin hapus merk ini?  
Semua mobil dengan merk ini akan ikut terhapus"

**REQ-5.3:** Jika "Tidak", batalkan proses

**REQ-5.4:** Jika "Ya":

- Mobile app kirim HTTP DELETE ke /api/merk/delete.php dengan parameter id
- Backend hapus merk dari tabel merk WHERE id = id
- CASCADE DELETE otomatis: Semua mobil dengan merk\_id = id akan terhapus otomatis (dan stok terkait juga ikut terhapus)
- Return success, Android tampilkan toast "Merk berhasil dihapus" dan refresh Dashboard

## 3.6 Manajemen Data Mobil - Tambah Mobil

### 3.6.1 Deskripsi dan Prioritas

Deskripsi: Admin dapat menambahkan data mobil baru.

Prioritas: HIGH (Wajib)

### 3.6.2 Stimulus/Response Sequences

1. Admin di List Mobil, klik FAB (+)
2. Sistem tampilkan form input mobil
3. Admin isi: nama mobil, tipe, tahun, harga, warna, stok awal
4. Admin klik "Simpan"
5. Sistem validasi dan cek duplikasi
6. Jika valid: Simpan ke database, tampilkan toast, kembali ke list
7. Jika duplikat: Tampilkan error

### 3.6.3 Functional Requirements

**REQ-6.1:** Sistem harus menyediakan FAB (+) di halaman List Mobil

**REQ-6.2:** Sistem tampilkan form dengan field:

- Nama Mobil (text)
- Tipe (dropdown: Sedan, SUV, MPV, Crossover, Hatchback, Off Road, Sport, Pickup, Electric, Hybrid, LCGC)
- Tahun (number, min: 2000)
- Merk (read-only, menampilkan nama merk bukan ID)

- Harga (number)
- Warna (text)
- Stok Awal (number, default 0)

**REQ-6.3:** Sistem validasi:

- Semua field wajib diisi
- Tahun harus  $\geq 2000$
- Harga harus angka positif
- Stok awal harus  $\geq 0$

**REQ-6.4:** Mobile app kirim HTTP POST ke /api/mobil/create.php dengan JSON data mobil lengkap (termasuk merk\_id yang diambil dari konteks)

**REQ-6.5:** Backend cek duplikasi: query tabel mobil untuk cek apakah sudah ada mobil dengan kombinasi (nama\_mobil + merk\_id + warna + tahun) yang sama

**REQ-6.6:** Jika duplikat, return error: "Mobil sudah ada!"

**REQ-6.7:** Jika valid:

- Backend INSERT ke tabel mobil dengan merk\_id
- Backend JOIN dengan tabel merk untuk return nama\_merk di response
- Backend INSERT stok awal ke tabel stok dengan mobil\_id dari LAST\_INSERT\_ID()
- Return success, android tampilkan toast "Mobil berhasil ditambahkan" dan kembali ke List Mobil

### 3.7 Manajemen Data Mobil - Lihat Detail Mobil

#### 3.7.1 Deskripsi dan Prioritas

Deskripsi: Admin dapat melihat detail lengkap informasi mobil.

Prioritas: MEDIUM

#### 3.7.2 Stimulus/Response Sequences

1. Admin di List Mobil, klik tombol Mata pada mobil
2. Sistem tampilkan halaman detail mobil dengan semua informasi
3. Admin dapat klik "Edit" atau "Kembali"

#### 3.7.3 Functional Requirements

**REQ-7.1:** Sistem menyediakan tombol Mata pada setiap item mobil

**REQ-7.2:** Mobile app kirim HTTP GET ke /api/mobil/read\_detail.php?id={mobil\_id} dengan token di header

**REQ-7.3:** Backend query JOIN tabel mobil, merk, dan stok untuk ambil data lengkap

**REQ-7.4:** Sistem tampilkan detail:

- Nama Mobil
- Merk
- Tipe
- Tahun
- Harga (format Rupiah)
- Warna
- Stok saat ini

**REQ-7.5:** Sistem menyediakan tombol "Edit" dan "Kembali"

### 3.8 Manajemen Data Mobil - Edit Mobil

#### 3.8.1 Deskripsi dan Prioritas

Deskripsi: Admin dapat mengubah data mobil yang sudah ada.

Prioritas: HIGH (Wajib)

#### 3.8.2 Stimulus/Response Sequences

1. Admin di Detail Mobil, klik tombol "Edit"
2. Sistem tampilkan form edit dengan data lama
3. Admin ubah data yang diperlukan
4. Admin klik "Simpan"
5. Sistem validasi dan update database
6. Sistem tampilkan toast sukses

#### 3.8.3 Functional Requirements

**REQ-8.1:** Sistem tampilkan form edit dengan semua field terisi data lama

**REQ-8.2:** Sistem validasi sama seperti tambah mobil (REQ-6.3)

**REQ-8.3:** Mobile app kirim HTTP PUT ke /api/mobil/update.php dengan JSON data mobil yang diupdate

**REQ-8.4:** Backend cek duplikasi: apakah ada mobil lain (selain yang diedit) dengan kombinasi (nama\_mobil + merk\_id + warna + tahun) yang sama

**REQ-8.5:** Jika duplikat dengan mobil lain, return error: "Mobil sudah ada!"

**REQ-8.6:** Jika valid:

- Backend UPDATE tabel mobil
- Backend UPDATE tabel stok jika stok diubah

- Return success, android tampilkan toast "Mobil berhasil diperbarui" dan kembali ke List Mobil

### 3.9 Manajemen Data Mobil - Hapus Mobil

#### 3.9.1 Deskripsi dan Prioritas

Deskripsi: Admin dapat menghapus data mobil dari sistem.

Prioritas: MEDIUM

#### 3.9.2 Stimulus/Response Sequences

1. Admin di List Mobil, klik tombol Hapus pada mobil
2. Sistem tampilkan dialog konfirmasi
3. Admin klik "Ya"
4. Sistem hapus mobil dan stok terkait
5. Sistem refresh list

#### 3.9.3 Functional Requirements

**REQ-9.1:** Sistem menyediakan tombol Hapus pada setiap item mobil

**REQ-9.2:** Sistem tampilkan dialog konfirmasi: "Yakin hapus mobil ini?"

**REQ-9.3:** Jika "Tidak", batalkan proses

**REQ-9.4:** Jika "Ya":

- Mobile app kirim HTTP DELETE ke /api/mobil/delete.php dengan parameter id
- Backend hapus record dari tabel mobil berdasarkan id (CASCADE delete otomatis hapus data di tabel stok)
- Return success, Android tampilkan toast "Mobil berhasil dihapus" dan refresh list

### 3.10 Manajemen Stok - Tambah Stok

#### 3.10.1 Deskripsi dan Prioritas

Deskripsi: Admin dapat menambah stok mobil sebanyak 1 unit.

Prioritas: HIGH (Wajib)

#### 3.10.2 Stimulus/Response Sequences

1. Admin di List Mobil, klik tombol (+) pada mobil
2. Sistem tampilkan dialog konfirmasi
3. Admin klik "Ya"
4. Sistem update stok +1 dan refresh tampilan

### 3.10.3 Functional Requirements

- REQ-10.1:** Sistem menyediakan tombol (+) pada setiap item mobil
- REQ-10.2:** Sistem tampilkan dialog konfirmasi: "Yakin ingin menambah stok?"
- REQ-10.3:** Jika "Tidak", batalkan proses
- REQ-10.4:** Jika "Ya":
- Mobile app kirim HTTP PUT ke /api/stok/tambah.php dengan JSON {mobil\_id}
  - Backend UPDATE tabel stok: jumlah\_stok = jumlah\_stok + 1, updated\_at = NOW()
  - Return success dengan data stok terbaru
  - Android tampilkan toast "Stok berhasil ditambah" dan update angka stok

## 3.11 Manajemen Stok - Kurangi Stok

### 3.11.1 Deskripsi dan Prioritas

Deskripsi: Admin dapat mengurangi stok mobil sebanyak 1 unit dengan validasi.

Prioritas: HIGH (Wajib)

### 3.11.2 Stimulus/Response Sequences

1. Admin di List Mobil
2. Sistem cek stok mobil
3. Jika stok > 0: Tombol (-) aktif
4. Jika stok = 0: Tombol (-) disabled
5. Admin klik tombol (-)
6. Sistem tampilkan dialog konfirmasi
7. Admin klik "Ya"
8. Sistem update stok -1

### 3.11.3 Functional Requirements

- REQ-11.1:** Sistem harus menyediakan tombol (-) pada setiap item mobil
- REQ-11.2:** Sistem cek stok di client side (Android):
- Jika jumlah\_stok = 0: Disable tombol (-), tampil abu-abu, tidak bisa diklik
  - Jika jumlah\_stok > 0: Enable tombol (-), tampil normal, bisa diklik
- REQ-11.3:** Ketika tombol (-) diklik (saat stok > 0), sistem tampilkan dialog konfirmasi: "Yakin ingin mengurangi stok?"
- REQ-11.4:** Jika "Tidak", batalkan proses

**REQ-11.5:** Jika "Ya":

- Mobile app kirim HTTP PUT ke /api/stok/kurangi.php dengan JSON {mobil\_id}
- Backend validasi stok > 0 terlebih dahulu
- Jika stok = 0, return error: "Stok sudah 0, tidak bisa dikurangi"
- Jika stok > 0, backend UPDATE tabel stok: jumlah\_stok = jumlah\_stok - 1, updated\_at = NOW()
- Return success dengan data stok terbaru
- Android tampilkan toast "Stok berhasil dikurangi" dan update angka stok secara real-time
- Jika stok baru = 0, otomatis disable tombol (-)

## 4 Kebutuhan Interface Eksternal

### 4.1 User Interfaces

#### 4.1.1 Halaman Login

##### Elemen UI:

- Logo aplikasi
- Text field: Email (hint "Masukkan email")
- Text field: Password (hint "Masukkan password", input di-mask)
- Button: "Login"
- Text error message (muncul saat login gagal)

##### Behavior:

- Tombol Login disabled jika email atau password kosong
- Tampilkan loading indicator saat proses login
- Tampilkan error "Email atau password salah" jika gagal

#### 4.1.2 Halaman Dashboard

##### Elemen UI:

- Toolbar dengan judul "Dashboard" dan menu Logout
- Grid/List Card Merk Mobil, setiap card berisi:
  - Nama merk
  - Jumlah mobil (contoh: "15 mobil")
  - Tombol Edit

- Tombol Hapus
- FAB (+) di kanan bawah untuk tambah merk
- Empty state jika belum ada merk

**Behavior:**

- Klik card → redirect ke List Mobil
- Klik Edit → pop-up edit merk
- Klik Hapus → dialog konfirmasi
- Klik FAB → pop-up tambah merk

#### 4.1.3 Halaman List Mobil

**Elemen UI:**

- Toolbar dengan judul "Daftar Mobil - [Nama Merk]" dan tombol Back
- List/LazyColumn dengan card mobil:
  - Nama mobil
  - Tipe, Tahun, Warna
  - Harga (format Rupiah)
  - Jumlah stok
  - Tombol: mata (Detail), Hapus, Tambah Stok, Kurangi Stok
- FAB (+) untuk tambah mobil
- Empty state jika belum ada mobil

**Behavior:**

- Tombol Kurangi Stok disabled jika stok = 0
- Klik Detail → redirect ke Detail Mobil
- Klik Tambah Stok/Kurangi Stok → dialog konfirmasi

#### 4.1.4 Halaman Detail Mobil

**Elemen UI:**

- Toolbar dengan judul "Detail Mobil" dan tombol Back
- Card info dengan layout:
  - Nama Mobil
  - Merk, Tipe, Tahun
  - Harga (format Rupiah)
  - Warna
  - Stok

- Button: "Edit Mobil"

#### 4.1.5 Halaman Form Mobil (Tambah/Edit)

##### **Elemen UI:**

- Toolbar dengan judul "Tambah Mobil" atau "Edit Mobil" dan tombol Back
- Text field: Nama Mobil
- Dropdown: Tipe Mobil
- Number field: Tahun (4 digit, min: 2000)
- Text field: Merk (read-only)
- Number field: Harga
- Text field: Warna
- Number field: Stok (hanya untuk tambah)
- Button: "Simpan"
- Text error message (muncul jika ada error)

##### **Behavior:**

- Tombol Simpan disabled jika ada field kosong
- Validasi input secara real-time

#### 4.1.6 Pop-up Dialog

##### **Elemen:**

- Judul dialog
- Message atau form input (untuk tambah/edit merk)
- Button: "Ya"/"Simpan" dan "Tidak"/"Batal"

#### 4.1.7 Toast Notification

##### **Contoh pesan:**

- "Login berhasil"
- "Merk berhasil ditambahkan"
- "Mobil berhasil dihapus"
- "Stok berhasil diperbarui"

## 4.2 Hardware Interfaces

### **Perangkat Keras yang Digunakan:**

1. Touchscreen Display
  - Input: Touch gestures (tap, swipe)
  - Output: Visual feedback

2. Storage Internal
  - Minimum: 50 MB untuk aplikasi (APK + cache)
  - Fungsi: Menyimpan token di SharedPreferences
3. RAM
  - Minimum: 2 GB
  - Fungsi: Running aplikasi dan caching data
4. Processor
  - Minimum: Dual-core
  - Fungsi: UI rendering dan data processing

#### **Tidak Ada Interface Hardware Tambahan:**

- Tidak ada kamera (karena tidak ada fitur foto)
- Tidak ada GPS
- Tidak ada NFC/Bluetooth

#### 4.3 Software Interfaces

##### 4.3.1 MySQL Database

Versi: MySQL 8.0+

Fungsi: Menyimpan data user, token, merk, mobil, dan stok

Interaksi: Backend PHP mengakses database menggunakan PDO

##### 4.3.2 PHP Backend

Versi: PHP 7.4+

Server: Apache (XAMPP)

Fungsi: Menangani request dari mobile app, business logic, dan komunikasi dengan database

Endpoints Utama:

- POST /api/login.php – Login
- POST /api/logout.php – Logout
- POST /api/merk/create.php - Tambah merk
- PUT /api/merk/update.php - Edit merk
- DELETE /api/merk/delete.php - Hapus merk
- GET /api/merk/read.php - Ambil data merk dengan jumlah mobil per merk
- GET /api/mobil/read.php?merk\_id=X - Ambil mobil per merk

- POST /api/mobil/create.php - Tambah mobil
- GET /api/mobil/read\_detail.php?id=X - Detail mobil
- PUT /api/mobil/update.php - Edit mobil
- DELETE /api/mobil/delete.php - Hapus mobil
- PUT /api/stok/tambah.php - Tambah stok
- PUT /api/stok/kurangi.php - Kurangi stok

#### 4.3.3 JWT Authentication

Library: firebase/php-jwt

Fungsi:

- Generate token saat login berhasil
- Validasi token di setiap protected endpoint
- Token berisi: user\_id, email, created\_at

#### 4.3.4 Password Hashing

Fungsi PHP: password\_hash() dan password\_verify()

Algoritma: bcrypt

Fungsi:

- Hash password sebelum disimpan ke database
- Verifikasi password saat login

#### 4.3.5 Retrofit (Android - HTTP Client)

Versi: Retrofit 2.9.0+

Fungsi:

- Kirim HTTP request ke backend API
- Parsing JSON response ke Kotlin data class
- Support Coroutines untuk async operations

Companion Library: Gson untuk JSON parsing

#### 4.3.6 Android Shared Preferences

Fungsi:

- Menyimpan JWT token secara lokal di device
- Mode: MODE\_PRIVATE untuk keamanan
- Token tersimpan sampai user logout

#### 4.3.7 JSON (Data Format)

Fungsi: Format data untuk request dan response antara Android dan PHP

Content-Type: application/json

### 4.4 Communications Interfaces

Protocol:

- Development: HTTP
- Production: HTTPS (wajib)

Data Format: JSON

Base URL:

- Emulator: http://10.0.2.2/showroom-backend/api/

HTTP Methods:

- GET (mengambil data)
- POST (membuat data baru)
- PUT (update data)
- DELETE (hapus data)

Authentication:

- Semua endpoint (kecuali login) wajib kirim token di header
- Format: Authorization: Bearer {token}

Network Requirements:

- Mobile app WAJIB koneksi internet
- Tidak ada offline mode

## 5 Kebutuhan Non-Fungsional

### 5.1 Performance Requirements

#### 5.1.1 Response Time

REQ-NFR-1.1: Aplikasi langsung ke Login Page

REQ-NFR-1.2: Proses login maksimal 3 detik (dengan koneksi internet stabil)

REQ-NFR-1.3: Load data list mobil maksimal 2 detik

REQ-NFR-1.4: Update stok (+/-) maksimal 2 detik

REQ-NFR-1.5: Transisi antar halaman smooth (< 500 ms)

### **5.1.2 Resource Usage**

REQ-NFR-1.6: Ukuran APK maksimal 30 MB

REQ-NFR-1.7: Penggunaan RAM maksimal 200 MB saat aplikasi aktif

## 5.2 Safety Requirements

### **5.2.1 Authentication**

REQ-NFR-2.1: Password di-hash menggunakan password\_hash() (bcrypt) sebelum disimpan ke database

REQ-NFR-2.2: Password tidak ditampilkan dalam plain text di UI (di-mask)

REQ-NFR-2.3: Sistem menggunakan JWT untuk autentikasi

REQ-NFR-2.4: Token disimpan di SharedPreferences (Android) dan tabel token (database)

REQ-NFR-2.5: Token hanya hilang saat logout manual

REQ-NFR-2.6: Setelah logout, token dihapus dari SharedPreferences dan database

### **5.2.2 Data Protection**

REQ-NFR-2.7: Komunikasi production harus menggunakan HTTPS

REQ-NFR-2.8: Backend menggunakan Prepared Statements (PDO) untuk mencegah SQL Injection

REQ-NFR-2.9: Validasi input dilakukan di client-side (Android) dan server-side (PHP)

### **5.2.3 Access Control**

REQ-NFR-2.10: Setiap API request (kecuali login) wajib kirim token di header

REQ-NFR-2.11: Backend validasi token di setiap protected endpoint

REQ-NFR-2.12: Jika token invalid, API return HTTP 401 dan Android redirect ke Login Page

## 5.3 Security Requirements

### **5.3.1 Usability**

REQ-NFR-3.1: UI harus intuitif dan mudah digunakan

REQ-NFR-3.2: Feedback visual harus jelas (loading indicator, toast message)

REQ-NFR-3.3: Navigasi konsisten (tombol back berfungsi dengan benar)

### **5.3.2 Reliability**

REQ-NFR-3.4: Aplikasi tidak boleh crash selama normal operation

REQ-NFR-3.5: Error handling yang jelas untuk setiap kasus (koneksi gagal, data tidak valid, dll)

### **5.3.3 Maintainability**

REQ-NFR-3.6: Kode Android mengikuti MVVM architecture pattern

REQ-NFR-3.7: Kode harus ter-dokumentasi dengan comment yang jelas

### **5.3.4 Portability**

REQ-NFR-3.8: Aplikasi support Android 7.0 ke atas (API 24+)

REQ-NFR-3.9: Aplikasi support berbagai screen size (min 5 inch)

## **6 Other Requirements**

### **6.1 Appendix A: Glossary (Glosarium)**

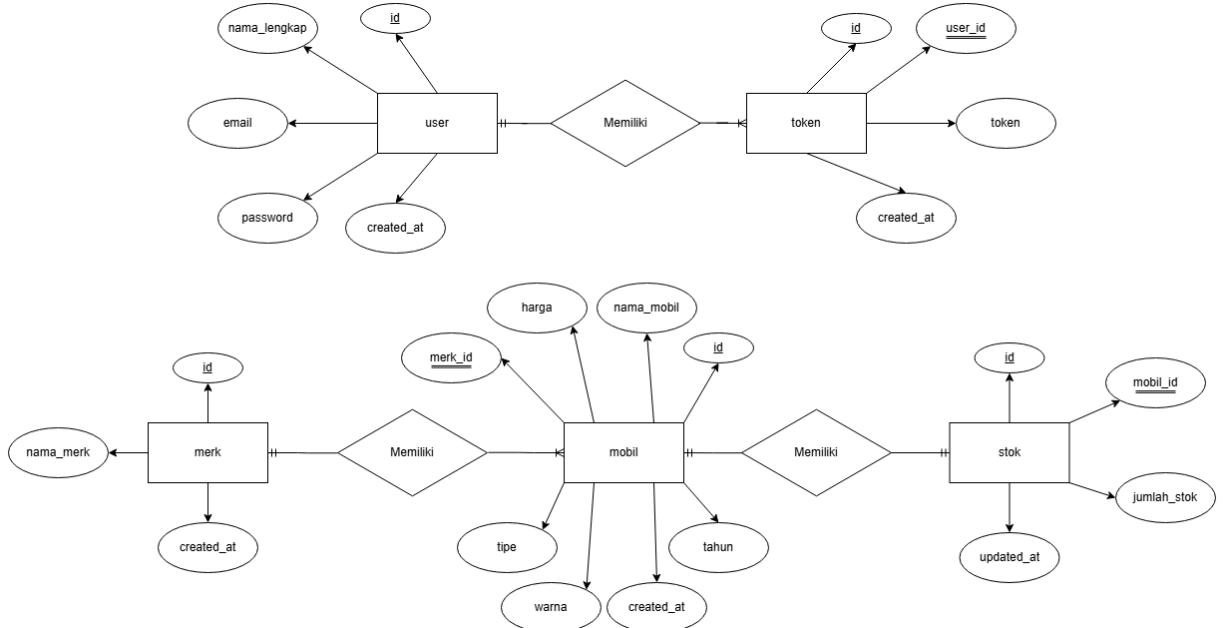
Istilah	Definisi
<b>Admin</b>	Pengguna aplikasi yang memiliki akses penuh untuk mengelola inventaris mobil
<b>API Endpoint</b>	URL spesifik di backend untuk melakukan operasi tertentu
<b>Authentication</b>	Proses verifikasi identitas pengguna melalui email dan password
<b>Backend</b>	Sistem server berbasis PHP yang mengelola logika bisnis, database, dan API
<b>bcrypt</b>	Algoritma hashing password yang aman, digunakan melalui fungsi password_hash() di PHP
<b>Client-Server</b>	Arsitektur aplikasi dimana client (Android) berkomunikasi dengan server (PHP)
<b>CRUD</b>	Create, Read, Update, Delete - Operasi dasar pada database
<b>FAB</b>	Floating Action Button - Tombol melayang untuk primary action
<b>Hash</b>	Proses mengubah password menjadi string acak untuk keamanan
<b>HTTP/HTTPS</b>	Protokol komunikasi antara client dan server (HTTPS terenkripsi)
<b>JSON</b>	Format data ringan untuk pertukaran data antara client dan server
<b>JWT</b>	JSON Web Token - Token untuk autentikasi berbasis JSON

<b>MySQL</b>	Relational Database Management System yang digunakan untuk menyimpan data
<b>PDO</b>	PHP Data Objects - Library PHP untuk koneksi dan query database
<b>phpMyAdmin</b>	Tool berbasis web untuk mengelola database MySQL
<b>Prepared Statement</b>	Teknik query SQL yang aman untuk mencegah SQL Injection
<b>REST API</b>	Arsitektur API yang menggunakan HTTP methods (GET, POST, PUT, DELETE)
<b>Request/Response</b>	Pola komunikasi dimana client mengirim request dan server mengirim response
<b>Retrofit</b>	HTTP client library untuk Android yang digunakan untuk call API
<b>Server</b>	Komputer yang menjalankan backend PHP dan database MySQL (XAMPP)
<b>Session</b>	Periode waktu user tetap login setelah autentikasi berhasil
<b>SharedPreferences</b>	Penyimpanan lokal di Android untuk data kecil seperti token
<b>Stok</b>	Jumlah unit mobil yang tersedia di showroom
<b>Token</b>	String unik yang merepresentasikan session user yang sudah login
<b>Validasi</b>	Proses pengecekan kebenaran data input
<b>XAMPP</b>	Package software yang berisi Apache, MySQL, PHP untuk development local

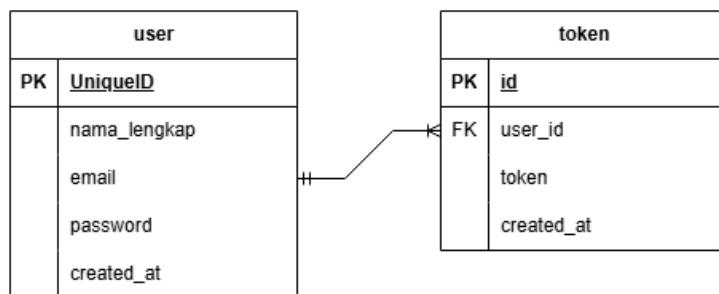
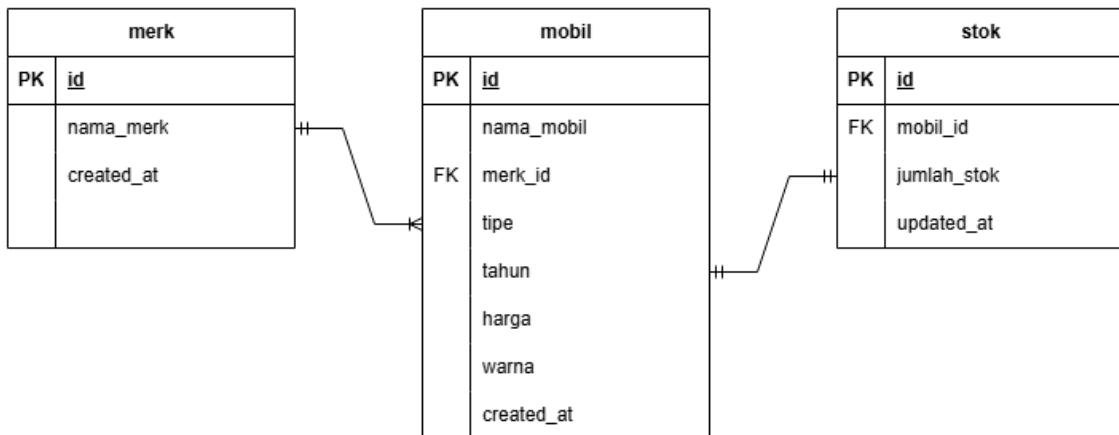
## 6.2 Appendix B: Analysis Models

Lampiran ini berisi diagram dan model analisis:

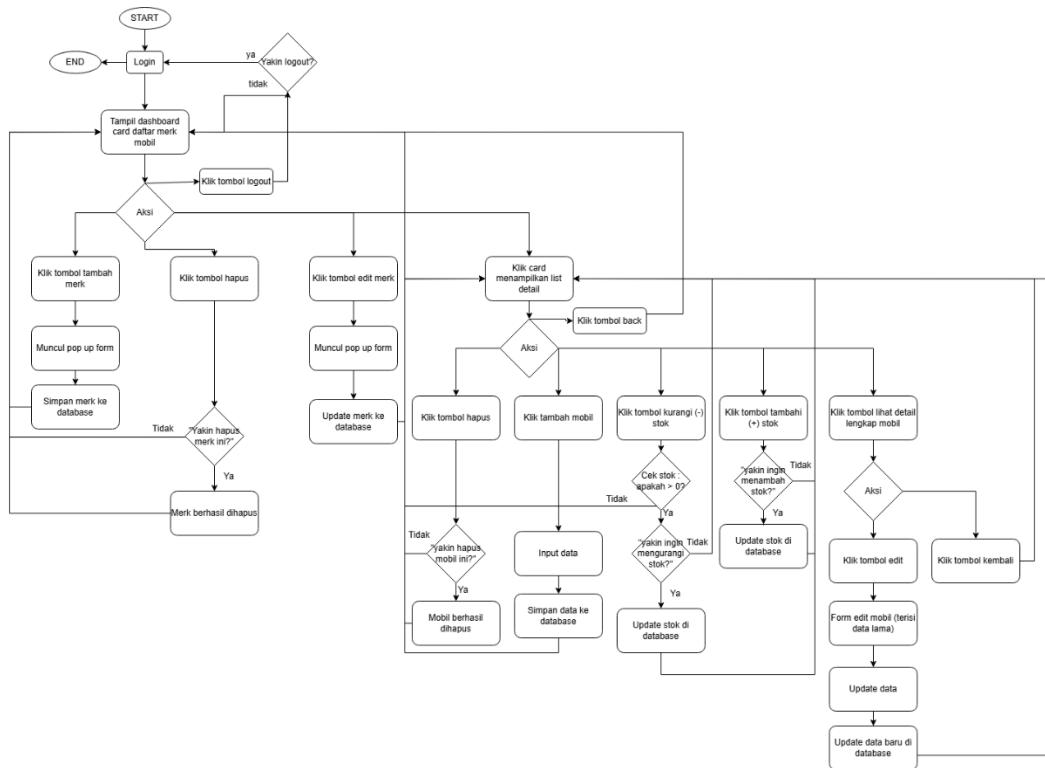
## 1. Entity Relationship Diagram (ERD)



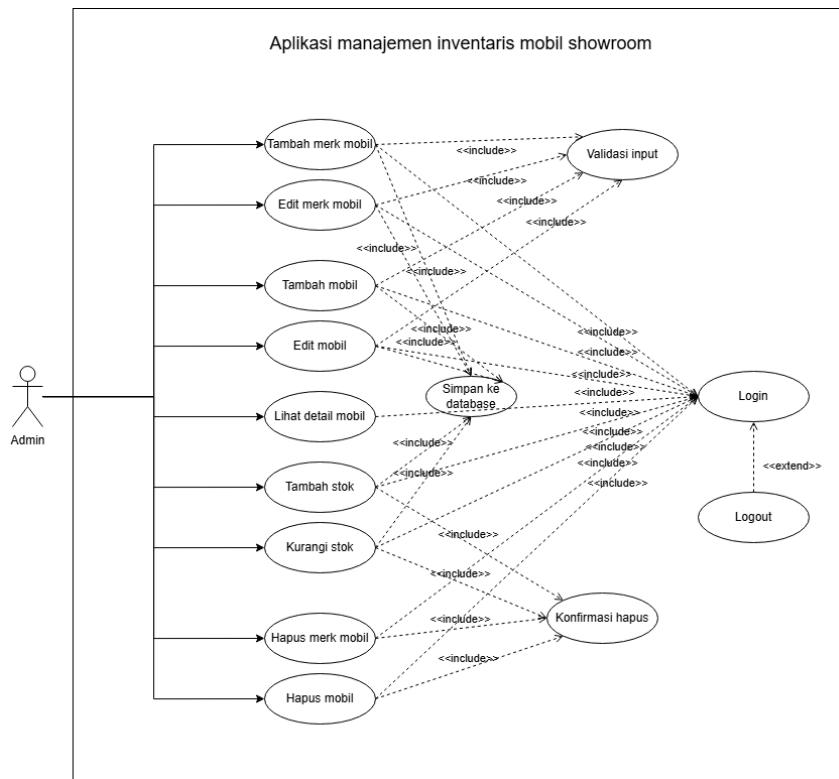
## 2. Relasi Antar Tabel (RAT)



### 3. Flowchart Aplikasi



### 4. Use Case Diagram



## 5. Kamus Data

### 1) Tabel user

#### Deskripsi Tabel

Tabel untuk menyimpan data admin showroom yang dapat login ke aplikasi.

#### Informasi Umum

Atribut	Keterangan
<b>Nama Tabel</b>	user
<b>Primary Key</b>	id
<b>Foreign Key</b>	-
<b>Jumlah Field</b>	5
<b>Relasi</b>	user (1) → token (N) - One to Many

#### Struktur Field

No	Nama Field	Tipe Data	Panjang	Null	Default	Key	Keterangan
1	id	INTEGER	-	NOT NULL	AUTO INCREMENT	PK	ID unik admin
2	nama_lengkap	VARCHAR	100	NOT NULL	-	-	Nama lengkap admin
3	email	VARCHAR	100	NOT NULL	-	UNIQUE	Email untuk login (unik)
4	password	VARCHAR	255	NOT NULL	-	-	Password yang sudah di-hash (bcrypt)
5	created_at	DATETIME	-	NOT NULL	CURRENT_TIMESTAMP	-	Tanggal admin terdaftar

#### Detail Field

1. Id
  - Primary key, auto-increment
  - Contoh: 1, 2, 3
2. nama\_lengkap
  - Nama lengkap admin

- Contoh: "Salmaa Rifhani Rayyan"
3. Email
- Email admin untuk login (harus unik)
  - Format: [nama@domain.com](mailto:nama@domain.com)
  - Contoh: [admin@showroom.com](mailto:admin@showroom.com)
  - Validasi: Harus ada karakter @
4. Password
- Password yang sudah di-hash menggunakan password\_hash() (bcrypt)
  - Panjang: 255 karakter (untuk hash bcrypt)
  - Contoh hash:  
\$2y\$10\$N9qo8uLOickgx2ZMRZoMyeIjZAgcfl7p92ldGxad68LJZdL17lhWy
  - PENTING: Tidak boleh simpan password plain text!
5. created\_at
- Timestamp otomatis saat admin dibuat
  - Format: YYYY-MM-DD HH:MM:SS
  - Contoh: 2025-12-07 10:30:00

### **Contoh Data**

id	nama_lengkap	email	password	created_at
1	Salmaa Rifhani Rayyan	admin@showroom.com	5e884898da28047151d0 e56f8dc 6292773603d0d6aabbdd 62a11ef721d1542d8	2025-12-07 10:30:00

### **Operasi pada Tabel user**

- CREATE: Saat pendaftaran admin baru (password harus di-hash dulu)
- READ: Saat login (cek email dan verify password)
- UPDATE: Update nama atau email (email harus tetap unik)
- DELETE: Hapus admin (CASCADE delete: token admin ikut terhapus)

### **Relasi**

USER (1) → TOKEN (N) - One to Many

- Satu user bisa memiliki banyak token
- CASCADE: Hapus user → semua token terkait ikut terhapus

2) Tabel token

### Deskripsi Tabel

Tabel untuk menyimpan token JWT yang di-generate saat admin berhasil login.

### Informasi Umum

Atribut	Keterangan
<b>Nama Tabel</b>	token
<b>Primary Key</b>	id
<b>Foreign Key</b>	user_id (referensi ke user.id)
<b>Jumlah Field</b>	4
<b>Relasi</b>	user (1) ← token (N) - Many to One

### Struktur Field

No	Nama Field	Tipe Data	Panjang	Null	Default	Key	Keterangan
1	id	INTEGER	-	NOT NULL	AUTO INCREMENT	PK	ID unik token
2	user_id	INTEGER	-	NOT NULL	-	FK	ID admin (referensi ke user.id)
3	token	TEXT	-	NOT NULL	-	-	String JWT token
4	created_at	DATETIME	-	NOT NULL	CURRENT_TIMESTAMP	-	Waktu token dibuat (saat login)

### Detail Field

1. id
  - Primary key, auto-increment
  - Contoh: 1, 2, 3
2. user\_id
  - Foreign key ke tabel user
  - Contoh: 1 (admin dengan id=1)
  - CASCADE: Jika user dihapus, token ikut terhapus

### 3. Field: token

- String JWT yang di-generate saat login

- Format:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoxLCJlbWFpbCI  
6ImFkbWluQHNob3dyb29tLmNvbSJ9.abc123...

- Panjang: ~200-500 karakter

- Berisi: user\_id, email (di-encode)

### 4. created\_at

- Timestamp otomatis saat token dibuat (saat login)

- Format: YYYY-MM-DD HH:MM:SS

- Contoh: 2025-12-07 10:30:00

## Contoh Data

id	User_id	token	created_at
1	1	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoxLCJlbWFpbCI 6ImFkbWluQHNob3dyb29tLmNvbSJ9.abc123...	2025-12-07 10:30:00

## Operasi pada Tabel token

- CREATE: Saat admin login berhasil (generate JWT, simpan ke database)
- READ: Validasi token saat setiap API request (cek apakah token ada di database)
- UPDATE: Tidak ada (token tidak diubah)
- DELETE: Saat logout (hapus token dari database)

## Relasi

USER (1) ← TOKEN (N)

- Satu user bisa punya banyak token (history login)
- CASCADE: Hapus user → token ikut terhapus

### 3) Tabel merk

## Deskripsi Tabel

Tabel untuk menyimpan daftar merk mobil yang tersedia di showroom.

## Informasi Umum

Atribut	Keterangan
Nama Tabel	merk
Primary Key	id
Foreign Key	-
Jumlah Field	3
Relasi	merk (1) → mobil (N) - One to Many

### Struktur Field

No	Nama Field	Tipe Data	Panjang	Null	Default	Key	Keterangan
1	id	INTEGER	-	NOT NULL	AUTO INCREMENT	PK	ID unik merk
2	nama_merk	VARCHAR	50	NOT NULL	-	UNIQUE	Nama merk mobil
3	created_at	DATETIME	-	NOT NULL	CURRENT_TIMESTAMP	-	Tanggal merk ditambahkan

### Detail Field

1. id
  - Primary key, auto-increment
  - Contoh: 1, 2, 3
2. nama\_merk
  - Nama merk mobil (harus unik)
  - Contoh: "Toyota", "Honda", "Suzuki", "Mitsubishi"
  - Maksimal: 50 karakter
  - UNIQUE: Tidak boleh ada duplikat (case-insensitive)
3. created\_at
  - Timestamp otomatis saat merk ditambahkan
  - Format: YYYY-MM-DD HH:MM:SS
  - Contoh: 2025-12-20 10:00:00

### Contoh Data

id	nama_merk	created_at

1	Toyota	2025-12-07 09:00:00
---	--------	---------------------

### Operasi pada Tabel merk

- CREATE: Tambah merk baru (validasi duplikasi nama\_merk)
- READ: Tampil list semua merk
- UPDATE: Edit nama merk (validasi duplikasi)
- DELETE: Hapus merk (delete juga semua mobil dengan merk tersebut)

### Relasi

merk (1) → mobil (N) - One to Many

- Satu merk bisa punya banyak mobil
- ON DELETE CASCADE: Saat hapus merk, semua mobil dengan merk tersebut juga terhapus
- ON UPDATE CASCADE: Saat update merk.id, merk\_id di tabel mobil otomatis terupdate

## 4) Tabel mobil

### Deskripsi Tabel

Tabel untuk menyimpan data mobil yang tersedia di showroom.

### Informasi Umum

Atribut	Keterangan
<b>Nama Tabel</b>	mobil
<b>Primary Key</b>	id
<b>Foreign Key</b>	merk_id (referensi ke merk.id)
<b>Jumlah Field</b>	8
<b>Relasi</b>	merk (1) ← mobil (N) – Many to One mobil (1) → stok (1) - One to One

### Struktur Field

No	Nama Field	Tipe Data	Panjang	Null	Default	Key	Keterangan
1	id	INTEGER	-	NOT NULL	AUTO INCREMENT	PK	ID unik mobil
2	nama_mobil	VARCHAR	100	NOT NULL	-	UNIQUE	Nama lengkap

							mobil + varian
3	Merk_id	INTEGER	-	NOT NULL	-	FK, UNIQUE	ID merk (referensi ke merk.id)
4	tipe	VARCHAR	50	NOT NULL	-	-	Tipe mobil (Sedan, SUV, MPV, dll)
5	tahun	INTEGER	-	NOT NULL	-	UNIQUE	Tahun produksi (4 digit)
6	harga	DECIMAL	15,2	NOT NULL	-	-	Harga mobil (Rupiah)
7	warna	VARCHAR	30	NOT NULL	-	UNIQUE	Warna mobil
8	created_at	DATETIME	-	NOT NULL	CURRENT_TIMESTAMP	-	Tanggal mobil ditambahkan

**Catatan UNIQUE:** Kombinasi (nama\_mobil + merk\_id + warna + tahun) harus unik

untuk mencegah duplikasi mobil yang sama persis.

### Detail Field

1. id
  - Primary key, auto-increment
  - Contoh: 1, 2, 3
2. nama\_mobil
  - Nama lengkap mobil termasuk varian
  - Contoh: "Avanza 1.5 G", "Innova Zenix 2.0 Q", "Brio RS"
  - Maksimal: 100 karakter
3. Merk\_id
  - Foreign key ke tabel merk (merk.id)
  - Tipe: INTEGER

- Contoh: 1, 2, 3
  - CASCADE:
    - Jika merk dihapus dari tabel merk → semua mobil dengan merk\_id tersebut ikut terhapus
    - Jika merk.id diupdate → merk\_id di tabel mobil otomatis terupdate (ON UPDATE CASCADE)
  - PENTING: Merk harus sudah ada di tabel merk (merk.id) sebelum bisa digunakan untuk mobil baru
  - Untuk menampilkan nama merk, harus JOIN dengan tabel merk
4. tipe
    - Kategori mobil
    - Pilihan: Sedan, SUV, MPV, Crossover, Hatchback, Off Road, Sport, Pickup, Electric, Hybrid, LCGC
    - Contoh: "MPV", "SUV", "Sedan"
    - Maksimal: 50 karakter
  5. tahun
    - Tahun produksi mobil (4 digit)
    - Contoh: 2023, 2024, 2025
    - Validasi: tahun  $\geq 2000$
  6. harga
    - Harga jual mobil dalam Rupiah
    - Format: DECIMAL(15,2)
    - Contoh: 250000000.00 (Rp 250 juta)
    - Validasi: harga  $> 0$
  7. warna
    - Warna eksterior mobil
    - Contoh: "Putih", "Hitam", "Silver", "Merah"
    - Maksimal: 30 karakter
  8. created\_at
    - Timestamp otomatis saat mobil ditambahkan
    - Format: YYYY-MM-DD HH:MM:SS
    - Contoh: 2025-12-07 09:00:00

## Contoh Data

<b>id</b>	<b>nama_mobil</b>	<b>Merk_id</b>	<b>tipe</b>	<b>tahun</b>	<b>harga</b>	<b>warna</b>	<b>created_at</b>
1	Avanza 1.5 G	1	MPV	2023	250000000.00	Putih	2025-12-07 09:00:00
2	Avanza 1.5 G	1	MPV	2023	250000000.00	Hitam	2025-12-07 09:00:00
3	Avanza 1.5 G	1	MPV	2024	250000000.00	Putih	2025-12-07 09:00:00

Penjelasan:

- Mobil id=1 dan id=2: Mobil SAMA tapi warna beda = BOLEH (tidak duplikat)
- Mobil id=1 dan id=3: Mobil SAMA tapi tahun beda = BOLEH (tidak duplikat)
- Mobil id=1: Tidak boleh tambah lagi "Avanza 1.5 G - 1 - Putih - 2023" = DUPLIKAT

## Validasi Duplikasi

a) Saat INSERT:

SELECT \* FROM mobil

WHERE nama\_mobil = 'Avanza 1.5 G'

AND merk\_id = 1

AND warna = 'Putih'

AND tahun = 2023

- Jika ada hasil → DUPLIKAT (tidak boleh simpan)
- Jika tidak ada → BOLEH simpan

b) Saat UPDATE:

SELECT \* FROM mobil

WHERE nama\_mobil = 'Avanza 1.5 G'

AND merk\_id = 1

AND warna = 'Putih'

AND tahun = 2023

AND id != 1

- Jika ada hasil → DUPLIKAT (tidak boleh update)
- Jika tidak ada → BOLEH update

## Operasi pada Tabel mobil

- CREATE: Tambah mobil baru (auto-generate id, validasi duplikasi, insert ke tabel stok juga)
- READ: Tampil list mobil per merk, tampil detail mobil
- UPDATE: Edit data mobil (validasi duplikasi kecuali mobil yang diedit)
- DELETE: Hapus mobil (CASCADE: stok ikut terhapus)

### **Relasi**

merk (1) ← mobil (N) - Many to One

- Setiap mobil memiliki satu merk

mobil (1) → stok (1) - One to One

- Setiap mobil memiliki tepat 1 record stok
- CASCADE: Hapus mobil → stok ikut terhapus

### 5) Tabel stok

#### **Deskripsi Tabel**

Tabel untuk menyimpan jumlah stok untuk setiap mobil.

#### **Informasi Umum**

Atribut	Keterangan
<b>Nama Tabel</b>	stok
<b>Primary Key</b>	id
<b>Foreign Key</b>	mobil_id (referensi ke mobil.id)
<b>Jumlah Field</b>	4
<b>Relasi</b>	mobil (1) ← stok (1) - One to One

#### **Struktur Field**

No	Nama Field	Tipe Data	Panjang	Null	Default	Key	Keterangan
1	id	INTEGER	-	NOT NULL	AUTO INCREMENT	PK	ID unik stok
2	mobil_id	INTEGER	-	NOT NULL	-	FK, UNIQUE	ID mobil (referensi ke mobil.id)

3	jumlah_stok	INTEGER	-	NOT NULL	0	-	Jumlah unit mobil tersedia
4	updated_at	DATETIME	-	NOT NULL	CURRENT_TIMESTAMP	-	Waktu terakhir stok diupdate

### Detail Field

1. id
  - Primary key, auto-increment
  - Contoh: 1, 2, 3
2. mobil\_id
  - Foreign key ke tabel mobil (UNIQUE - setiap mobil hanya punya 1 stok)
  - Contoh: 1 (mobil dengan id=1)
  - CASCADE: Jika mobil dihapus, stok ikut terhapus
3. jumlah\_stok
  - Jumlah unit mobil yang tersedia
  - Contoh: 0, 5, 15, 100
  - Default: 0
  - Validasi: jumlah\_stok  $\geq 0$  (tidak boleh negatif)
  - Operasi:
    - Tambah stok: jumlah\_stok = jumlah\_stok + 1
    - Kurangi stok: jumlah\_stok = jumlah\_stok - 1 (jika  $> 0$ )
4. updated\_at
  - Timestamp terakhir kali stok diupdate
  - Format: YYYY-MM-DD HH:MM:SS
  - Contoh: 2025-12-07 14:30:00
  - Auto-update setiap kali stok berubah

### Contoh Data

id	mobil_id	jumlah_stok	updated_at
1	1	15	2025-12-07 14:30:00

### Operasi pada Tabel stok

- CREATE: Saat tambah mobil baru (auto-insert stok dengan jumlah\_stok default 0 atau sesuai input)
- READ: Query stok berdasarkan mobil\_id, JOIN dengan tabel mobil untuk tampilkan nama mobil + stok
- UPDATE:
  - Tambah Stok (+): Admin klik tombol (+) → konfirmasi → stok +1 → update updated\_at
  - Kurangi Stok (-): Admin klik tombol (-) → konfirmasi → stok -1 (jika > 0) → update updated\_at
- DELETE: CASCADE delete saat mobil dihapus

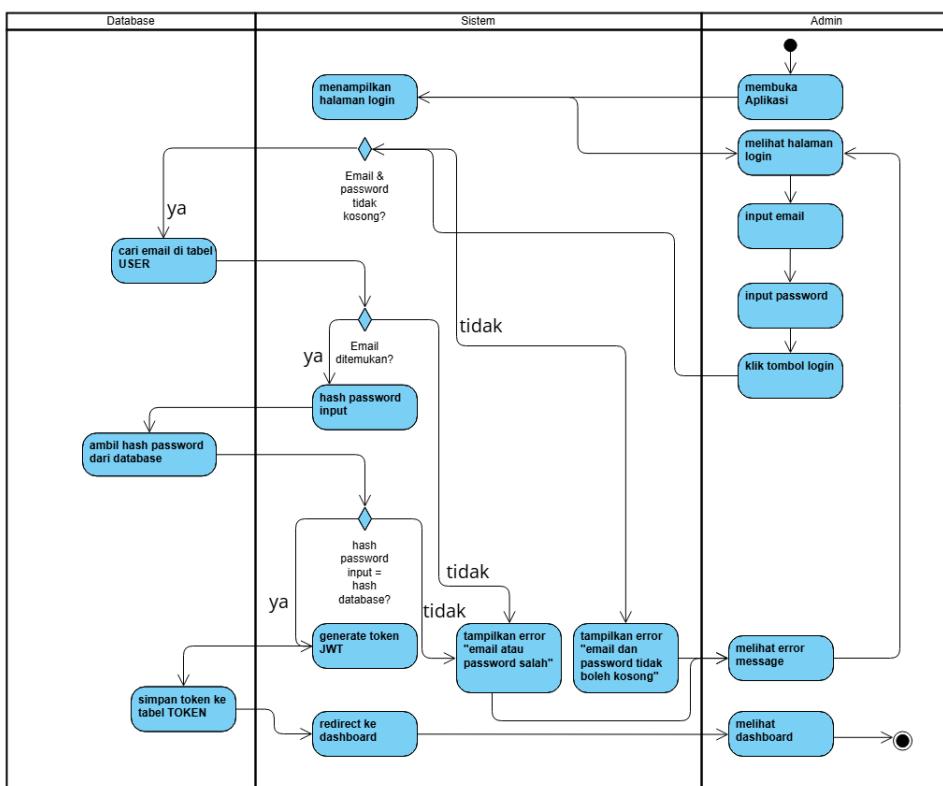
## Relasi

mobil (1) → stok (1) - One to One

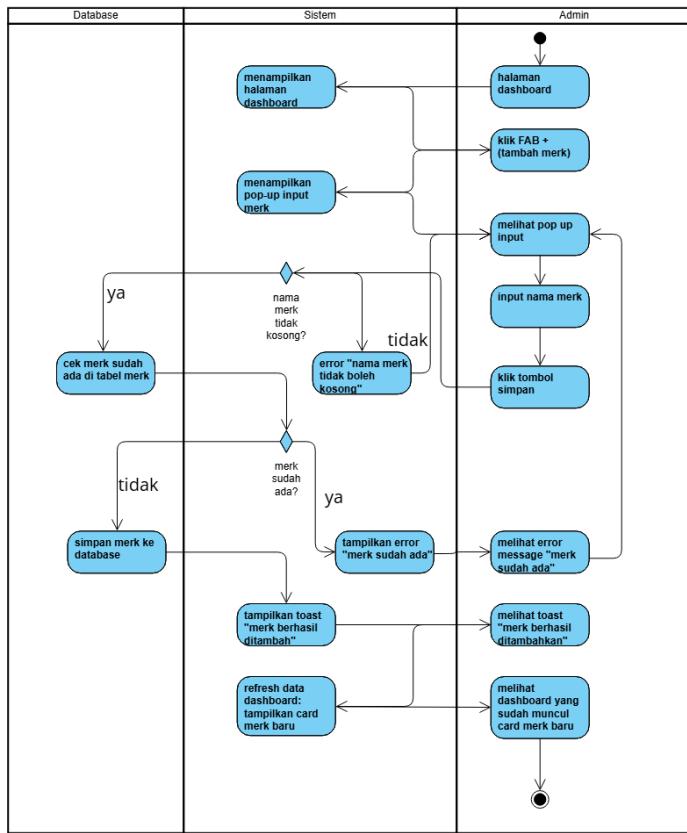
- Setiap mobil memiliki tepat 1 record stok
- CASCADE: Hapus mobil → stok ikut terhapus

## 6. Activity Diagram

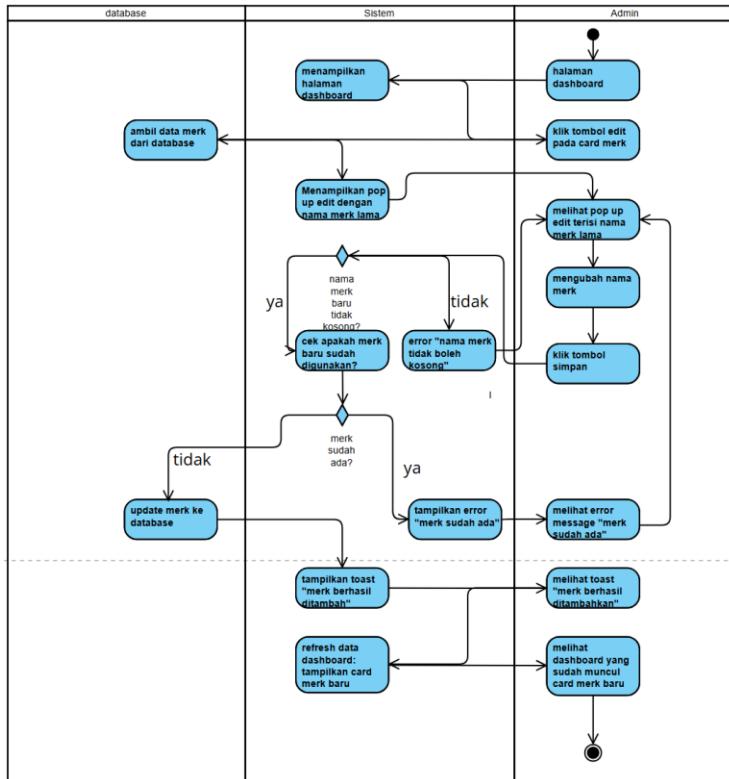
### Activity diagram - login



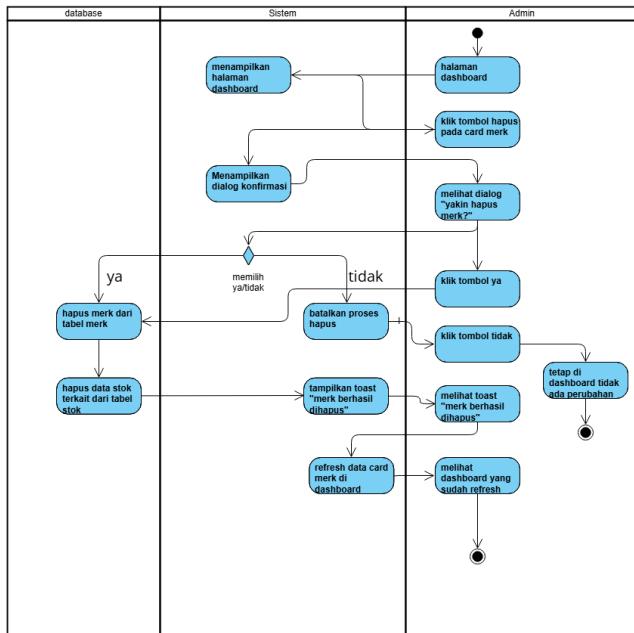
## Activity diagram – create merk



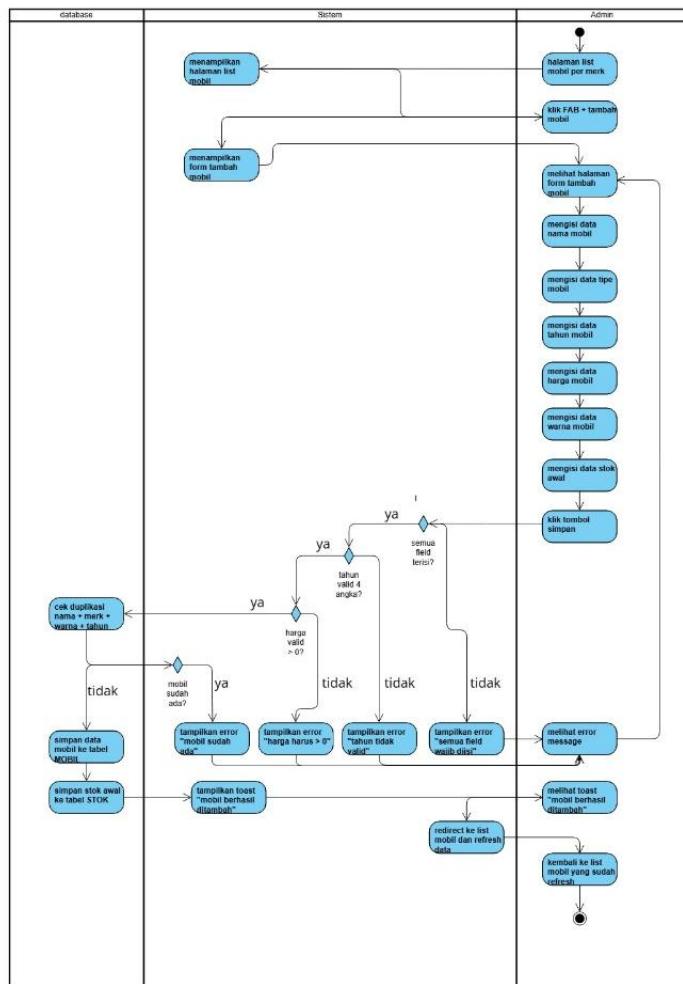
## Activity diagram – edit merk



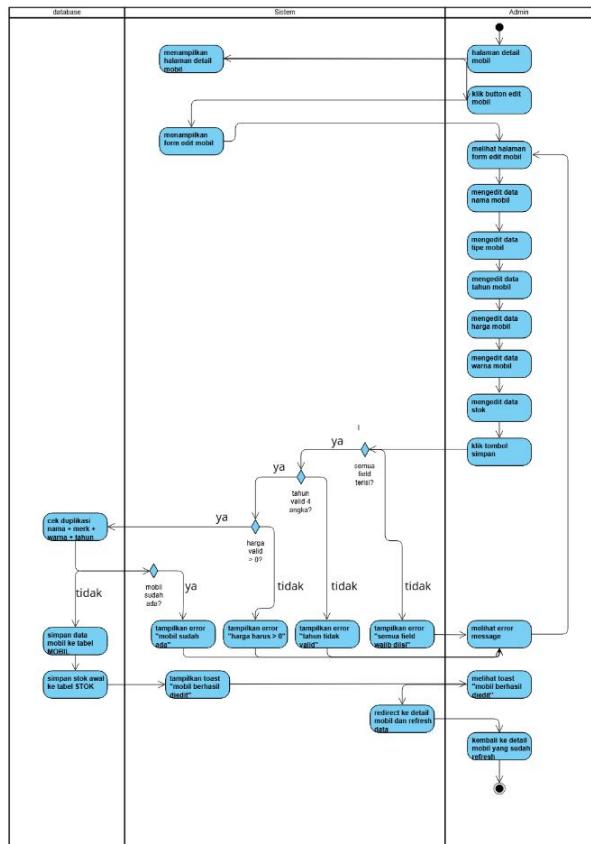
## Activity diagram – delete merk



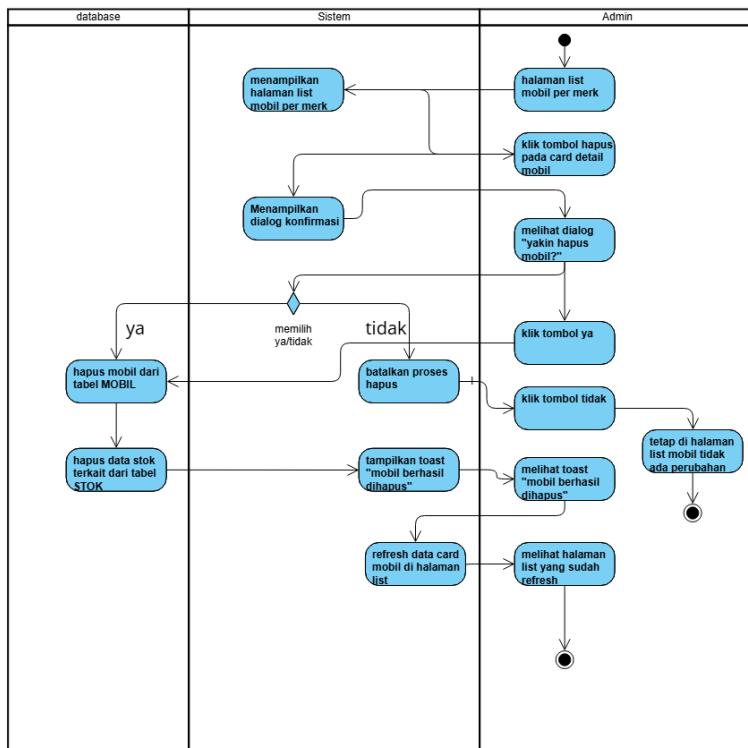
## Activity diagram – create mobil



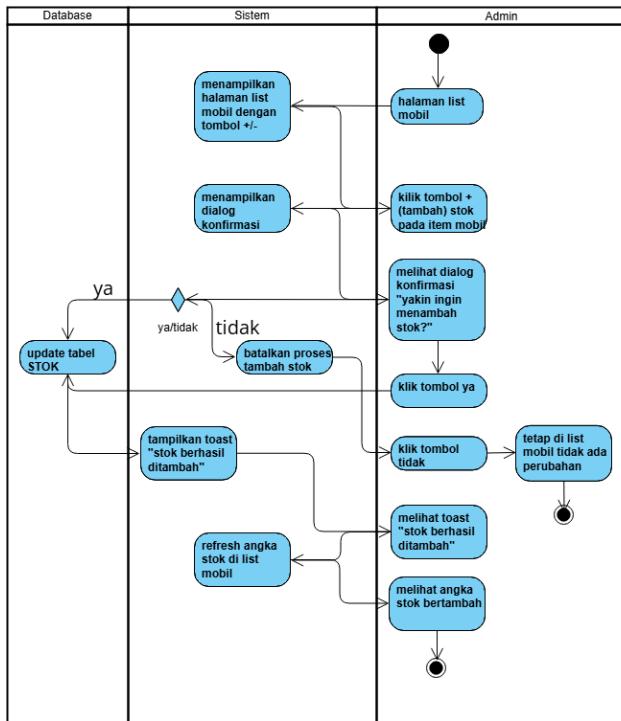
## Activity diagram – edit mobil



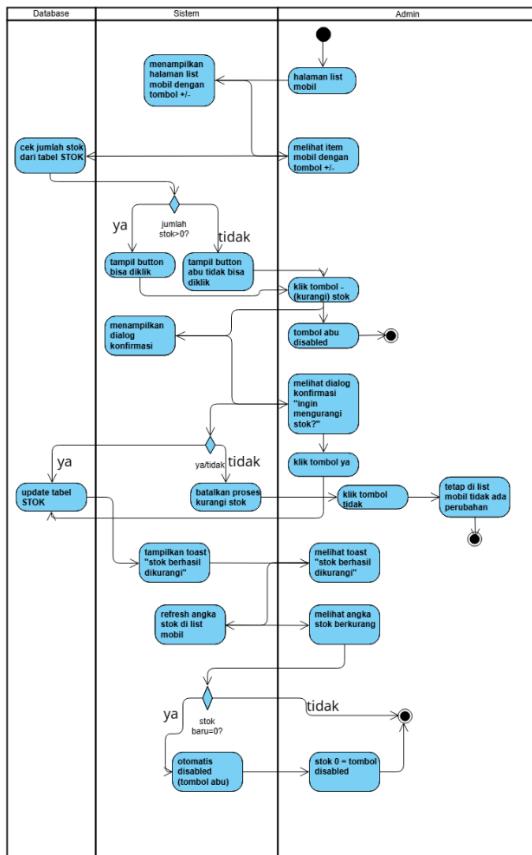
## Activity diagram – hapus mobil



## Activity diagram – tambah stok



## Activity diagram – kurangi stok



### 6.3 Appendix C: Issues List

Daftar isu yang perlu dipertimbangkan untuk pengembangan lebih lanjut:

Issues yang Belum Resolved:

1. Foto Mobil
  - Belum ada fitur upload foto mobil
  - Solusi masa depan: Tambah fitur upload dari galeri/kamera
2. Search & Filter
  - Belum ada fitur pencarian mobil
  - Belum ada filter berdasarkan tahun, harga, atau stok
  - Solusi masa depan: Tambah search bar dan filter options
3. Laporan/Report
  - Belum ada fitur laporan atau statistic
  - Solusi masa depan: Tambah dashboard analytics (total mobil, stok menipis, dll)
4. Backup Database
  - Belum ada automated backup untuk database MySQL
  - Solusi masa depan: Implementasi backup manual atau automated
5. Export Data
  - Belum ada fitur export data ke CSV/Excel
  - Solusi masa depan: Tambah fitur export untuk reporting

Known Limitations:

1. Network Dependency: Aplikasi wajib koneksi internet (tidak ada offline mode)
2. Single User: Belum ada multi-user atau role-based access control
3. No Audit Trail: Belum ada tracking perubahan data (siapa edit apa dan kapan)
4. API Security: Belum ada rate limiting atau advanced security untuk production