*Software Requirements Specification*

for

Sistem Kasir dan Manajemen Stok Produk Bakso Djatigiri

Version 1.0 approved

Prepared by

1122140093 – Fachrio Raditya

1122140109 – M. Juan Adi Pratama

1122140090 – Fandi Fadillah

Fakultas Teknik Informatika/Software Engineering

16 Mei 2025

Table of Contents

1. Pendahuluan 1

1.1 Tujuan Penulisan Dokumen 1

1.2 Audien yang Dituju dan Pembaca yang Disarankan 1

1.3 Batasan Produk 1

1.4 Definisi dan Istilah 2

1.5 Refrensi 2

2. Deskripsi Keseluruhan 3

2.1 Deskripsi Produk 3

2.2 Fungsi Produk 3

2.3 Penggolongan Karakterik Pengguna 3

2.4 Lingkungan Operasi 4

2.5 Batasan Desain dan Implementasi 4

2.6 Dokumentasi Pengguna 5

3. Kebutuhan Antarmuka Eksternal 6

3.1 User Interfaces 6

3.2 Hardware Interface 6

3.3 Software Interface 6

3.4 Communication Interface 7

4. Functional Requirement 9

4.1 Use Case Diagram 11

4.2 Use Case: Login 11

4.3 Use Case: Transaksi Penjualan (Kasir) 13

4.4 Use Case: History Transaksi 15

4.5 Use Case: Stok Bahan (Owner) 16

4.6 Use Case: Profile 17

4.7 Use Case: CRUD Menu (Owner) 18

4.8 Class Diagram 21

5. Non Functional Requirements 22

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini disusun untuk mendefinisikan kebutuhan fungsional dan non‑fungsional dari aplikasi mobile Bakso Djatigiri, yang akan diimplementasikan menggunakan Flutter sebagai framework frontend dan **Firebase** sebagai backend. Tujuan utamanya adalah memberikan panduan yang jelas bagi tim pengembang, tester, dan pemangku kepentingan (stakeholders) dalam merancang, membangun, dan memelihara aplikasi sesuai dengan kebutuhan bisnis klien.

## Audien yang Dituju dan Pembaca yang Disarankan

Dokumen ini ditujukan untuk:

1. Tim Pengembang (Developers): Untuk memahami secara detail fitur MVP (kasir & manajemen stok) dan rencana fase selanjutnya.
2. Klien / Pemilik Usaha (Bakso Djatigiri): Sebagai acuan verifikasi bahwa kebutuhan bisnis telah tertangani.
3. Staf Operasional / Admin: Untuk memahami alur sistem kasir dan manajemen stok yang akan mereka gunakan.

## Batasan Produk

Aplikasi Bakso Djatigiri versi awal (MVP) mencakup modul:

1. Sistem Kasir: Input pesanan, kalkulasi total, pencatatan transaksi (history).
2. Manajemen Stok Produk: Penambahan, pengurangan, dan pelacakan level stok.

Batasan‑batasan:

1. Platform: Android dan iOS (melalui Flutter).
2. Integrasi: Saat MVP belum mendukung pembayaran online maupun integrasi layanan pihak ketiga (mis. ojek online).
3. Mode Offline: Aplikasi memerlukan koneksi internet untuk berfungsi (Firebase Realtime Database).
4. Keamanan: Autentikasi hanya menggunakan email + password (tanpa OAuth/Social Login).

Selanjutnya (fase berikutnya), akan ditambahkan:

1. Transaksi Online (order & pembayaran via aplikasi).
2. Analitik Penjualan & Pembelian (dashboard statistik).

## Definisi dan Istilah

* SRS : *Software Requirements Specification*, atau

Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

* IEEE : *Institute of Electrical and Electronics Engineering*

Standar internasional untuk pengembangan dan perancangan produk.

* MVP : Minimum Viable Product, versi awal produk dengan fitur inti untuk

validasi pasar.

* Firebase : Platform Backend‑as‑a‑Service (BaaS) oleh Google, menyediakan

database realtime, autentikasi, dan hosting*.*

* Flutter : UI toolkit open‑source oleh Google untuk membangun aplikasi natively

compiled pada mobile, web, dan desktop.

* Stok Produk : Jumlah ketersediaan bahan baku atau menu siap saji yang

tercatat di sistem.

* Transaksi : Proses penjualan, termasuk input pesanan, perhitungan jumlah, dan

pencatatan ke database.

## Refrensi

<Tulis daftar dokumen lain atau alamat web yang menjadi acuan SRS ini.>

# Deskripsi Keseluruhan

## Deskripsi Produk

Aplikasi mobile Bakso Djatigiri adalah sistem digital berbasis Flutter dengan backend Firebase yang dirancang untuk mendukung proses operasional bisnis UMKM kuliner, khususnya warung mie ayam dan bakso. Produk ini dirancang sebagai solusi pencatatan transaksi penjualan serta pengelolaan stok bahan dan produk secara real-time, guna menggantikan pencatatan manual dan meningkatkan efisiensi operasional. Aplikasi ini ditargetkan untuk digunakan langsung oleh pemilik usaha atau staf kasir, dengan fokus utama pada Minimum Viable Product (MVP) berupa sistem kasir dan manajemen stok. Seiring pertumbuhan bisnis, aplikasi akan dikembangkan lebih lanjut untuk mendukung transaksi online dan fitur analitik data keuangan dan inventori.

## Fungsi Produk

Fungsi utama yang akan tersedia dalam aplikasi meliputi:

1. Pencatatan transaksi penjualan secara digital.
2. Perhitungan total pembayaran termasuk diskon atau pajak jika ada.Manajemen stok produk: input, update, dan pengurangan stok saat penjualan terjadi.
3. Histori transaksi penjualan harian.
4. Autentikasi pengguna (login/logout sederhana).
5. Notifikasi stok menipis & rekomendasi pembelian ulang.

## Penggolongan Karakterik Pengguna

Aplikasi akan digunakan oleh dua golongan pengguna utama, yaitu staf kasir dan pemilik usaha (owner/admin):

Tabel 1 Karakteristik Pengguna

| **Kategori Pengguna** | **Tugas** | **Hak Akses ke aplikasi** | **Kemampuan yang harus dimiliki** |
| --- | --- | --- | --- |
| Kasir | Mencatat transaksi penjualan harian | Insert Data | Entry data transaksi, memilih produk, input jumlah, menyelesaikan transaksi |
| Owner | Memeriksa laporan, mengelola stok, dan memperbaiki data transaksi jika diperlukan | Insert, Update, Delete Data | Memahami dasar laporan penjualan, mengelola data stok, validasi data transaksi |

## Lingkungan Operasi

Aplikasi akan beroperasi dalam lingkungan sebagai berikut:

1. Platform Aplikasi:
2. Android OS (versi minimal 8.0/Oreo)
3. iOS (versi minimal iOS 13)
4. Framework Frontend: Flutter (latest stable release)
5. Backend & Database: Firebase Firestore, Firebase Authentication
6. Perangkat Keras:
7. Smartphone minimal RAM 3 GB dan penyimpanan 32 GB
8. Koneksi internet diperlukan untuk sinkronisasi data secara real-time
9. Tools Pengembangan:
10. Flutter SDK
11. Firebase Console
12. VSCode / Android Studio

## Batasan Desain dan Implementasi

Berikut batasan teknis dan kebijakan dalam pengembangan:

1. Teknologi: Harus menggunakan Flutter dan Firebase sesuai kesepakatan dengan klien.
2. Integrasi Pembayaran: Tidak tersedia pada MVP, akan dimasukkan di fase pengembangan berikutnya.
3. Offline Mode: Tidak didukung – aplikasi memerlukan koneksi internet aktif.
4. Multi User Support: Akan disediakan hanya untuk dua level akses (kasir dan owner).
5. Keamanan: Sistem login terbatas pada Firebase Email & Password Auth (Authentikasi).
6. Visual Branding: Belum tersedia, maka UI/UX akan bersifat minimal dan fungsional dulu.

## Dokumentasi Pengguna

Dokumentasi pengguna yang akan disediakan saat pengiriman aplikasi versi MVP:

1. Tutorial Onboarding (in-app): Penjelasan ringkas pertama kali membuka aplikasi.
2. FAQ Online: Daftar pertanyaan dan jawaban umum (jika diperlukan).
3. Video Tutorial (opsional): Untuk edukasi pemilik usaha dalam penggunaan aplikasi.

# Kebutuhan Antarmuka Eksternal

## User Interfaces

User interface untuk sementara berada di figma: <https://www.figma.com/design/UpDHzglt1AReeBChhv3SmE/Bakso-Djatigiri?node-id=0-1&t=tV4yjUtoNu8cd8T8-1>

## Hardware Interface

Aplikasi akan dijalankan pada perangkat smartphone yang digunakan oleh pelaku UMKM (kasir atau pemilik usaha). Karakteristik antarmuka perangkat keras adalah sebagai berikut:

1. Perangkat yang Didukung:
2. Smartphone Android (minimal OS Android 8.0/Oreo)
3. Smartphone iOS (minimal iOS 13)
4. Interaksi Perangkat Keras:
5. Touchscreen: Input data transaksi & navigasi antarmuka
6. Koneksi Internet: Wi-Fi atau data seluler diperlukan untuk sinkronisasi data ke Firebase
7. Tidak Ada Perangkat Tambahan (printer kasir, barcode scanner, dll. belum dibutuhkan di fase MVP)

## Software Interface

Aplikasi Bakso Djatigiri akan berinteraksi dengan beberapa komponen perangkat lunak, sebagai berikut:

1. Firebase Firestore (Cloud-hosted NoSQL Database)
2. Digunakan untuk menyimpan data transaksi, produk, stok, dan user.
3. Menggunakan struktur koleksi–dokumen yang fleksibel dan mendukung kueri kompleks.
4. Komunikasi berlangsung via protokol HTTPS menggunakan format JSON.
5. Mendukung sinkronisasi offline secara otomatis untuk penggunaan dalam kondisi sinyal terbatas.
6. Firebase Authentication
7. Sistem otentikasi berbasis email & password.
8. Digunakan untuk login kasir dan owner.
9. Tidak menggunakan autentikasi sosial atau SSO di tahap MVP.
10. Flutter SDK (Latest Stable)
11. Digunakan sebagai framework utama UI/UX dan logic aplikasi.
12. Dependency utama (melalui pub.dev):

firebase\_core, cloud\_firestore, firebase\_auth, flutterfire\_ui, provider

(opsional: flutter\_hooks, intl, uuid, dll. jika dibutuhkan)

1. Platform Operating System
2. Android dan iOS, dengan build output dari Flutter engine ke masing-masing native platform.
3. Tidak membutuhkan sistem operasi tambahan atau software eksternal lainnya.
4. Data Sharing
5. Seluruh data disimpan di Firestore dan dapat dibagikan antar perangkat yang login dengan role sesuai.
6. Firestore Rules akan digunakan untuk memfilter hak akses berdasarkan UID pengguna dan role mereka.

## Communication Interface

Komunikasi antar komponen frontend dan backend terjadi melalui koneksi internet, dengan karakteristik sebagai berikut:

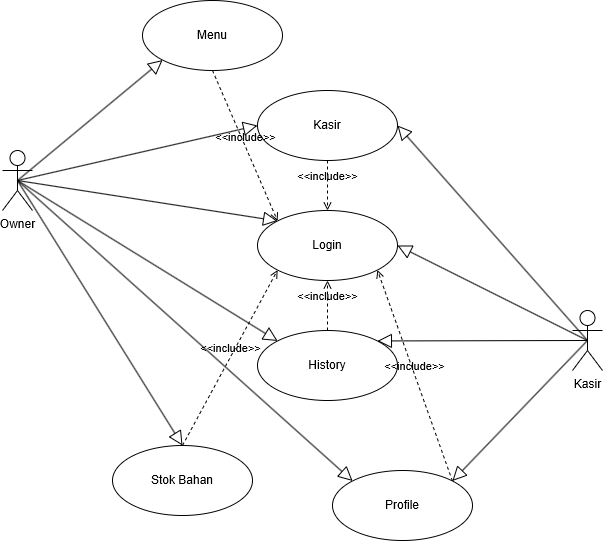
1. Protokol Komunikasi
2. HTTPS (Secure REST API): Semua komunikasi dengan Firebase Firestore dan Authentication berlangsung melalui HTTPS.
3. Firestore juga menggunakan gRPC di bawah kap, tetapi ini dikelola otomatis oleh SDK Firebase dan tidak perlu diatur manual.
4. Standar Komunikasi
5. Firebase SDK menjamin enkripsi komunikasi, otorisasi per pengguna, dan audit access control melalui Firestore Security Rules.
6. Tidak menggunakan komunikasi dengan server atau API pihak ketiga di MVP.
7. Tidak ada dependency pada protokol lain seperti FTP, SMTP, atau SOAP.
8. Keamanan dan Enkripsi
9. Seluruh komunikasi dilindungi TLS 1.2 atau lebih tinggi.
10. Firebase Authentication menyediakan token identitas berbasis JWT yang divalidasi secara otomatis oleh Firestore.
11. Tidak ada penyimpanan data sensitif di sisi client (hanya token akses yang sifatnya sementara).
12. Sinkronisasi & Offline Mode
13. Firestore mendukung sync otomatis: data lokal tersimpan secara sementara, dan akan otomatis tersinkronisasi saat koneksi internet kembali tersedia.
14. Cocok untuk lingkungan UMKM dengan koneksi tidak stabil.
15. Tidak memerlukan sistem pull/push manual oleh pengguna.
16. Data Throughput
17. Beban transfer data kecil karena Firestore hanya mengirimkan dokumen yang berubah (delta).
18. Cocok untuk aplikasi ringan dengan skala transaksi harian menengah ke bawah.

# Functional Requirement

Area ini menjelaskan kebutuhan fungsional utama berdasarkan fitur dan alur bisnis yang telah didefinisikan.

| **ID** | **Kebutuhan Fungsional** | **Penjelasan** |
| --- | --- | --- |
| FR-01 | Autentikasi Pengguna | User dapat login menggunakan email & password melalui Firebase Auth. |
| FR-02 | Validasi Role dan Status | Sistem mengecek status dan role user saat login, hanya active yang bisa masuk. |
| FR-03 | Navigasi Berdasarkan Role | Sistem menampilkan UI dan fitur sesuai role (owner atau kasir). |
| FR-04 | Manajemen Pengguna (Owner) | Owner dapat melihat daftar user, menonaktifkan/mengaktifkan user, dan mengatur role. |
| FR-05 | CRUD Bahan Baku | Owner dapat membuat, melihat, mengedit, dan menghapus data ingredients. |
| FR-06 | CRUD Menu | Owner dapat mengelola menu, termasuk mengatur bahan baku (menu\_requirements). |
| FR-07 | Validasi Bahan Saat Simpan Menu | Tidak bisa menyimpan menu jika bahan tidak tersedia dalam database. |
| FR-08 | Validasi Stok Bahan Sebelum Checkout | Sistem mengecek apakah stok bahan cukup sebelum transaksi dilakukan. |
| FR-09 | Transaksi Penjualan (Kasir) | Kasir dapat menambahkan menu ke keranjang, menghitung total, dan input pembayaran. |
| FR-10 | Perhitungan Kembalian Otomatis | Sistem menghitung dan menampilkan kembalian jika pembayaran cukup. |
| FR-11 | Gagal Checkout Jika Pembayaran Kurang | Sistem memblokir checkout jika uang pelanggan kurang dari total tagihan. |
| FR-12 | Pengurangan Stok Bahan Setelah Transaksi | Setelah transaksi berhasil, stok bahan di ingredients dikurangi otomatis. |
| FR-13 | Penyimpanan Riwayat Transaksi & Detail Item | Data transaksi dan detail item disimpan di koleksi transactions dan transaction\_items. |
| FR-14 | Tampilan Riwayat Transaksi | Owner/kasir dapat melihat daftar transaksi harian dan detail transaksi. |

## Use Case Diagram



## Use Case: Login

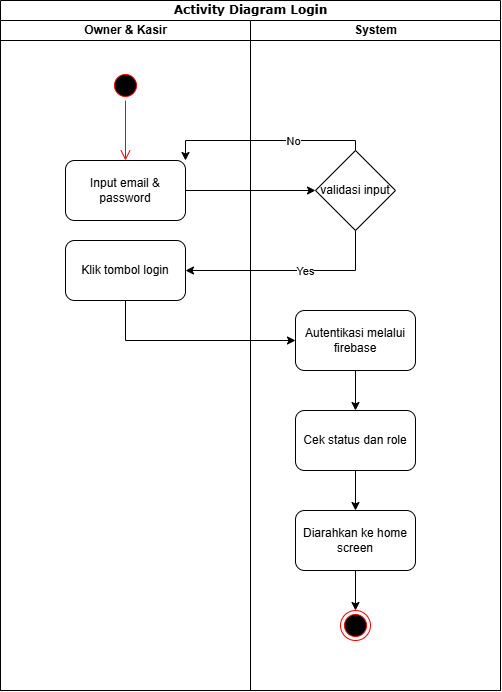
4.2.1 Deskripsi Use Case

Pengguna (kasir atau owner) melakukan autentikasi menggunakan email dan password. Sistem akan memverifikasi data dan hanya mengizinkan pengguna aktif untuk login. Setelah berhasil, sistem akan mengarahkan pengguna ke halaman sesuai dengan perannya.

4.2.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Input email dan password |  |
|  | 2. Sistem validasi format input |
| 3. Klik tombol login |  |
|  | 4. Sistem autentikasi melalui Firebase Auth |
|  | 5. Sistem cek status dan role user |
|  | 6. Di arahkan ke home screen |

4.2.3 Activity Diagram

**

## Use Case: Transaksi Penjualan (Kasir)

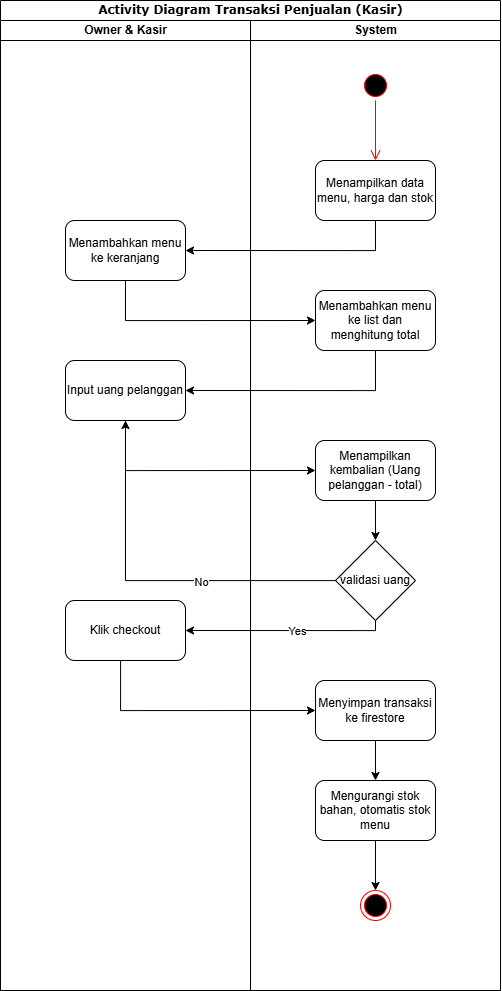
4.3.1 Deskripsi Use Case

Kasir melakukan proses transaksi penjualan mulai dari memilih menu, input jumlah, melakukan checkout, dan mencatat pembayaran. Sistem akan mengecek ketersediaan bahan sebelum mengizinkan transaksi.

4.3.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
|  | 1. Sistem menampilkan data menu. harga dan stok |
| 2. Kasir menambahkan menu ke keranjang |  |
|  | 3. Sistem menambahkan menu ke list dan menghitung total |
| 4. Kasir input uang pelanggan |  |
|  | 5. Sistem menampilkan kembalian (uang pelanggan – total) |
| 6. Kasir klik checkout |  |
|  | 7. Sistem menyimpan transaksi ke firestore |
|  | 8. Sistem mengurangi stok bahan, otomatis stok menu |

* + 1. Activity Diagram



## Use Case: History Transaksi

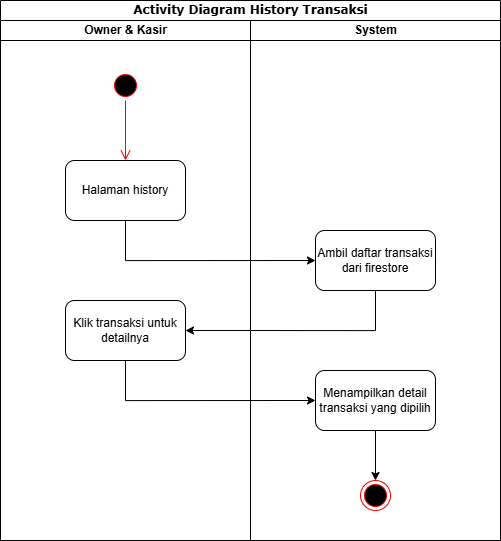
4.4.1 Deskripsi Use Case

Pengguna dapat melihat daftar transaksi yang telah dilakukan, difilter berdasarkan tanggal, serta melihat detail isi transaksi.

4.4.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Buka halaman history |  |
|  | 2. Sistem ambil daftar transaksi dari firestore |
| 3. Klik transaksi untuk detailnya |  |
|  | 4. Menampilkan detail transaksi yang dipilih |

4.4.3 Activity Diagram



## Use Case: Stok Bahan (Owner)

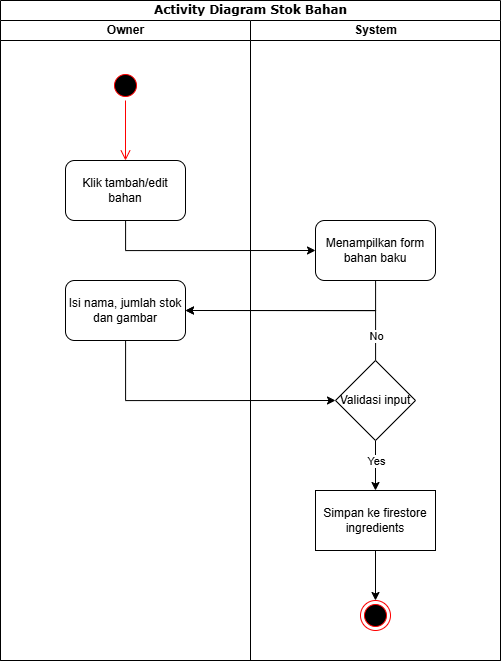
4.5.1 Deskripsi Use Case

Owner dapat mengelola bahan baku seperti menambah, memperbarui, dan menghapus data bahan sesuai kebutuhan.

4.5.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Klik tambah/edit bahan |  |
|  | 2. Sistem tampilkan form bahan baku |
| 3. Isi nama, jumlah stok dan gambar |  |
|  | 4. Sistem validasi dan simpan ke firestore ingredients |
| 5. Klik hapus bahan |  |
|  | 6. Sistem konfirmasi dan hapus data firestore |

4.5.3 Activity Diagram



## Use Case: Profile

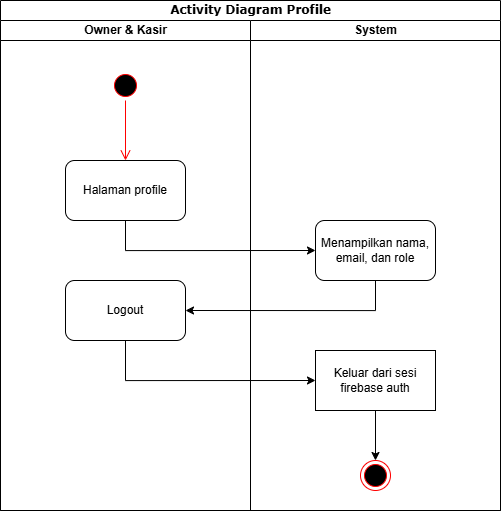
4.6.1 Deskripsi Use Case

Pengguna dapat melihat informasi akun sendiri seperti nama dan email, serta logout dari sistem.

4.6.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Buka halaman profile |  |
|  | 2. Sistem tampilkan nama, email dan role |
| 3. Klik logout |  |
|  | 4. Sistem keluar dari sesi firebase auth |

4.6.3 Activity Diagram



## Use Case: CRUD Menu (Owner)

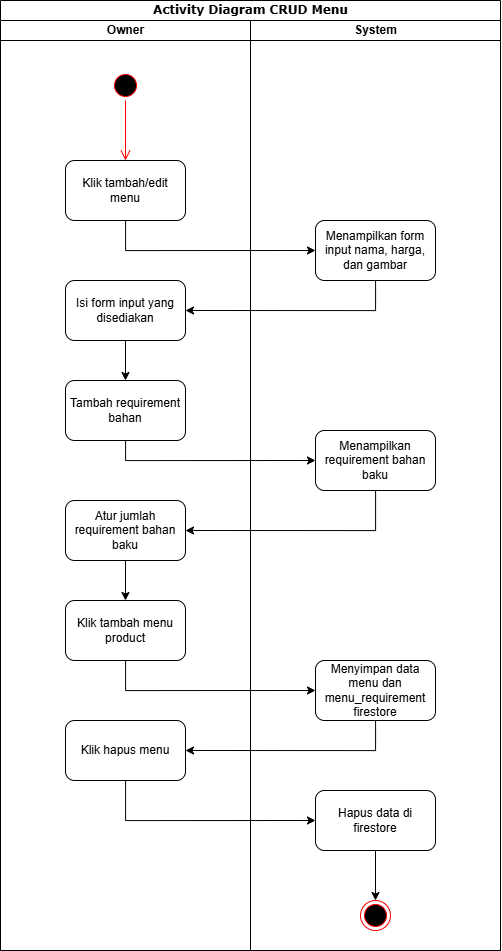
4.7.1 Deskripsi Use Case

Owner dapat mengelola menu makanan, termasuk menentukan harga dan kebutuhan bahan untuk tiap menu.

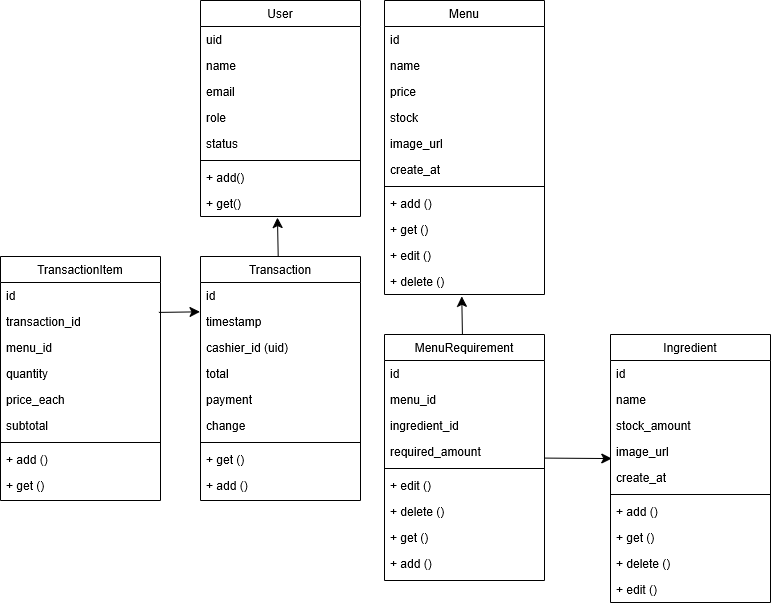
4.7.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. Klik tambah/edit menu |  |
|  | 2. Sistem menampilkan form input nama, harga, dan gambar |
| 3. Isi form input yang sudah disediakan |  |
| 4. Tambah requirement bahan |  |
|  | 5. Sistem tampilkan requirement bahan baku |
| 6. Atur jumlah requirement bahan baku |  |
| 7. Klik tambah menu product |  |
|  | 8. Sistem menyimpan data menu dan menu\_requirement di firestore |
| 9. Klik hapus menu |  |
|  | 10. Sistem hapus data di firestore. |

4.7.3 Activity Diagram



## Class Diagram



# Non Functional Requirements

<*Uraikan dengan ringkas kebutuhan non fungsional dalam tabel sebagai berikut. Isilah Kolom Kebutuhan dengan kalimat yang jelas dan kelak dapat ditest untuk dipenuhi. ID adalah nomor kebutuhan yang harus ditelusuri pada saat test. Tuliskan N/A bila Not Applicable>*

| **ID** | **Parameter** | **Kebutuhan** |
| --- | --- | --- |
|  | Availability |  |
|  | Reliability |  |
|  | Ergonomy |  |
|  | Portability |  |
|  | Memory |  |
|  | Response time |  |
|  | Safety | N/A |
|  | Security |  |
|  |  |  |
|  | Others 1: Bahasa komunikasi | Misalnya : semua tanya jawab harus dalam bahasa Indonesia |
|  |  | Setiap layar harus mengandung logo PT Pos Indonesia |
|  |  |  |

Catatan :

*Availability : ketersediaan aplikasi, misalnya harus terus menerus beroperasi 7 hari perminggu, 24 jam per haritanpa gagal*

*Reliability : keandalan, misalnya tidak pernah boleh gagal(atau kegagalan yang ditolerir adalah …%) sehingga harus dipikirkan fault tolerant architecture. Biasanya hanya perlu untuk Critical Application yang jika gagal akan berakibat fatal.*

*Ergonomy : kenyamanan pakai bagi pengguna*

*Portability : kemudahan untuk dibawa dan dioperasikan ke mesin/sistem operasi/platform yang lain*

*Memory : jika perhitungan kapasitas memori internal kritis (misalnya untuk SW yang harus dijadikan CHIPS dan ukurannya harus kecil*

*Response time : Batasan waktu yang harus dipenuhi. Sangat penting untuk aplikasi Real Time. Contoh: “Aaplikasi harus mampu menampilkan hasil dalam 4 detik”, atau “ATM harus menarik kembali kartu yang tidak diambil dalam waktu 3 menit”*

*Safety: yang menyangkut keselamatan manusia, misalnya untuk SW yang dipakai pada sistem kontrol di pabrik*

*Security : aspek keamanan yang harus dipenuhi*