*Software Requirements Specification*

for

KANTINKU

Version 1.0 approved

Prepared by

1122140004 – Almaidah Isnaeni

1122140128 – Muhamad Rodiyansah

1122140057 – Irvandy Zainur Firmansyah

11/05/2025

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 1

1.5 Refrensi 1

2. Deskripsi Keseluruhan 2

2.1 Deskripsi Produk 2

2.2 Fungsi Produk 2

2.3 Penggolongan Karakterik Pengguna 2

2.4 Lingkungan Operasi 2

2.5 Batasan Desain dan Implementasi 2

2.6 Dokumentasi Pengguna 3

3. Kebutuhan Antarmuka Eksternal 4

3.1 User Interfaces 4

3.2 Hardware Interface 4

3.3 Software Interface 4

3.4 Communication Interface 4

4. Functional Requirement 5

4.1 Use Case Diagram 5

4.2 Nama Use Case 1 5

4.3 Nama Use Case 2 5

4.4 Class Diagram 6

5. Non Functional Requirements 7

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason For Changes | Version |
|  |  |  |  |
|  |  |  |  |
| Irvandy Zainur | 11/05/2025 | Update Payment Sistem | 1.1 |
|  |  |  |  |

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen Software Requirements Specifications (SRS) disusun untuk memberikan gambaran umum mengenai sistem aplikasi Kantinku, termasuk ruang lingkup, fungsi utama, dan batasan produk. Dokumen ini juga bertujuan sebagai acuan dalam pengembangan sistem, pengujian, dan implementasi aplikasi.

## Audien yang Dituju dan Pembaca yang Disarankan

Dokumen SRS ini ditujukan untuk berbagai pihak yang terlibat dalam proses pengembangan, implementasi, dan penggunaan aplikasi **Kantinku** — sebuah sistem pemesanan dan manajemen kantin digital, di antaranya:

* **Pengembang Perangkat Lunak (Developer)**  
  Bertanggung jawab dalam merancang dan membangun sistem sesuai spesifikasi fungsional dan teknis yang tercantum dalam dokumen ini.
* **Pemilik Kantin atau Penjual (Merchant)**  
  Sebagai pihak yang menyediakan produk makanan/minuman, mereka menggunakan sistem untuk mengelola menu, menerima pesanan, dan melakukan transaksi.
* **Pengguna Akhir (Pelanggan Kantin)**  
  Menggunakan aplikasi ini untuk melakukan pemesanan makanan/minuman, melakukan pembayaran, dan memantau status pesanan.

## Batasan Produk

* Sistem hanya mencakup fitur pemesanan makanan dan minuman secara online, pengelolaan menu, pencatatan transaksi, serta pembayaran digital.
* Sistem tidak mencakup layanan pengiriman (delivery), pengguna harus mengambil pesanan secara langsung di kantin.
* Sistem tidak terintegrasi dengan sistem eksternal lain seperti sistem keuangan institusi, sistem presensi, atau aplikasi logistik pihak ketiga.
* Sistem hanya mendukung metode pembayaran digital melalui integrasi dengan **Tripay** sebagai payment gateway, metode pembayaran lain (tunai, transfer langsung) tidak dicatat dalam sistem.
* Akses sistem hanya tersedia untuk pengguna yang telah terdaftar sebagai pelanggan, penjual (pemilik kantin).
* Sistem tidak menyediakan fitur pelacakan riwayat konsumsi pengguna secara detail (misalnya statistik kalori atau pola makan).
* Tidak tersedia fitur pengingat atau notifikasi melalui SMS atau WhatsApp semua notifikasi hanya ditampilkan di dalam aplikasi.

## Definisi dan Istilah

* **SRS** : *Software Requirements Specification*, dokumen yang menjelaskan kebutuhan fungsional dan non-fungsional dari suatu sistem perangkat lunak.
* **IEEE** : *Institute of Electrical and Electronics Engineers*, organisasi yang menetapkan standar teknis termasuk untuk dokumentasi perangkat lunak.
* **Kantinku** : Nama aplikasi mobile yang dirancang untuk memfasilitasi pemesanan makanan dan minuman secara digital di lingkungan institusi seperti sekolah, kampus, atau kantor.
* **Admin** : Pengguna dengan hak akses untuk mengelola menu, memproses pesanan, memverifikasi pembayaran, dan melihat laporan transaksi.
* **User** : Pengguna akhir (pelanggan) yang menggunakan aplikasi Kantinku untuk melihat menu, memesan makanan/minuman, dan melakukan pembayaran.
* **Tripay** : Penyedia layanan *payment gateway* yang digunakan aplikasi Kantinku untuk memproses transaksi pembayaran secara digital.
* **Payment Gateway** : Sistem yang menghubungkan aplikasi dengan penyedia layanan pembayaran (seperti e-wallet atau virtual account) untuk memproses transaksi secara online.
* **Firebase**  : Platform dari Google yang digunakan dalam aplikasi ini untuk autentikasi pengguna, penyimpanan data (database), dan notifikasi real-time.
* **Cloud**-**based** : Sistem penyimpanan dan pengolahan data yang dilakukan secara online melalui server cloud, bukan secara lokal di perangkat.

## Referensi

Dokumen ini disusun berdasarkan hasil observasi langsung terhadap kebutuhan pengguna, serta berbagai sumber referensi daring dan aplikasi pembanding yang digunakan selama proses perancangan dan pengembangan aplikasi Kantinku. Referensi tersebut meliputi :

* **Aplikasi Tomoro Coffee**

Digunakan sebagai acuan (benchmark) dalam hal desain antarmuka, kemudahan navigasi, dan alur pemesanan yang sederhana namun efektif.

* **Firebase Documentation – Authentication & Realtime Database**

Referensi utama untuk pengelolaan autentikasi pengguna dan penyimpanan data berbasis cloud. <https://firebase.google.com/docs>

* **Flutter Documentation**

Panduan resmi dalam pengembangan aplikasi mobile menggunakan Flutter dan bahasa Dart. <https://docs.flutter.dev>

* **Material Design 3 Guidelines**

Digunakan sebagai pedoman dalam perancangan antarmuka pengguna (UI) agar konsisten dan sesuai standar Google. <https://m3.material.io>

* **Forum Komunitas (Stack Overflow, Github, dll.)**

Dimanfaatkan sebagai sumber pemecahan masalah teknis selama proses pengembangan aplikasi.

* **Catatan Internal dan Hasil Diskusi Tim**

Termasuk brainstorming, pertemuan tim, serta dokumentasi kebutuhan pengguna yang dikumpulkan selama tahap analisis dan desain sistem.

|  |
| --- |
|  |

# Deskripsi Keseluruhan

## Deskripsi Produk

**Kantinku** adalah sebuah aplikasi berbasis mobile yang dirancang untuk mempermudah proses pemesanan makanan dan minuman di lingkungan kantin, baik di sekolah, kampus, maupun kantor. Aplikasi ini memungkinkan pengguna (pelanggan) untuk melihat menu yang tersedia, melakukan pemesanan secara langsung melalui perangkat mereka, serta melakukan pembayaran secara digital tanpa harus mengantre secara fisik.

Berdasarkan masalah-masalah tersebut, dibuatlah aplikasi Kantinku sebagai solusi digital untuk:

* Mengelola penjualan secara otomatis.
* Menyediakan laporan dan grafik analisis penjualan.
* Memberikan kemudahan bagi pelanggan dan pengelola dalam melakukan transaksi.

Selain itu, Kantinku juga menyediakan antarmuka khusus untuk penjual (pemilik kantin) guna mengelola daftar menu, stok makanan, pesanan masuk, serta laporan penjualan harian. Dengan sistem ini, transaksi menjadi lebih efisien, cepat, dan transparan bagi kedua belah pihak. Produk ini bertujuan untuk mengurangi antrian panjang, meningkatkan kenyamanan pelanggan, serta mendukung digitalisasi layanan kantin secara menyeluruh.

## Fungsi Produk

* Menampilkan daftar menu makanan dan minuman yang tersedia di kantin.
* Memungkinkan pengguna melakukan pemesanan makanan/minuman secara online.
* Menyediakan metode pembayaran digital (e-wallet, transfer, dsb).
* Mengelola data pesanan masuk secara real-time untuk penjual.
* Menyediakan laporan transaksi dan riwayat pembelian bagi pengguna dan penjual.
* Memberikan notifikasi status pesanan (diproses, siap, selesai).

## Penggolongan Karakterik Pengguna

Tabel 1 Karakteristik Pengguna

|  |  |  |  |
| --- | --- | --- | --- |
| Kategori Pengguna | Tugas | Hak Akses ke Aplikasi | Kemampuan yang harus dimiliki |
| ADMIN (Penjual) | Mengelola menu, memproses pesanan, dan mencatat transaksi | Insert, Update, Delete Data | Menginput dan mengelola data menu serta transaksi |
| USER (Customer) | Melakukan pemesanan dan melihat status transaksi. | Insert, Update Data | Melakukan pemesanan, ubah data profil |

## Lingkungan Operasi

Aplikasi **Kantinku** dirancang untuk beroperasi dalam lingkungan sebagai berikut:

* Platform :

Aplikasi berbasis mobile, dikembangkan khusus untuk perangkat **Android**

* Perangkat Keras (Hardware) :

Smartphone Android dengan spesifikasi minimum:

* RAM minimal 2 GB
* Koneksi internet yang stabil (Wi-Fi atau data seluler)
* Sistem Operasi (Operating System) :
* Android OS versi 11 (Android R) atau yang lebih baru
* Mendukung API Level 30 ke atas
* Komponen Perangkat Lunak Pendukung (Software Components) :
* **Android SDK** (API Level 30 atau lebih tinggi) untuk proses pengembangan dan kompilasi aplikasi
* **Firebase** untuk layanan:
  + - * Autentikasi pengguna
      * Penyimpanan data (Realtime Database)
      * Notifikasi real-time

## Batasan Desain dan Implementasi

Berikut adalah beberapa batasan yang perlu diperhatikan selama proses desain dan implementasi aplikasi **Kantinku**:

1. **Batasan Sistem Operasi**

Aplikasi hanya dapat berjalan pada perangkat dengan Android 11 (API Level 30) atau yang lebih baru, karena Flutter memerlukan dukungan dari versi Android tersebut untuk menjalankan fitur-fitur terbaru.

1. **Keterbatasan Perangkat Keras**

Aplikasi memerlukan perangkat dengan minimal 2 GB RAM untuk memastikan performa yang optimal, karena Flutter mengharuskan sumber daya memadai untuk rendering antarmuka.

1. **Koneksi Internet**

Aplikasi ini bergantung pada koneksi internet yang stabil karena tidak menggunakan penyimpanan lokal. Semua data disimpan di cloud, dan setiap transaksi memerlukan koneksi aktif untuk berfungsi dengan baik.

1. **Penyimpanan Cloud**

Data pengguna, transaksi, dan menu disimpan secara cloud-based menggunakan Firebase, sehingga aplikasi tidak mengandalkan penyimpanan lokal atau database internal perangkat.

1. **Batasan Keamanan**

Semua komunikasi antara aplikasi dan server harus dilakukan melalui protokol HTTPS untuk menjamin keamanan data pengguna, khususnya dalam transaksi pembayaran dan pengiriman data sensitif.

1. **Keterbatasan Teknologi**

Aplikasi dikembangkan menggunakan Flutter dengan Dart sebagai bahasa pemrograman utama. Aplikasi ini ditujukan untuk platform Android, sehingga tidak ada dukungan untuk pengembangan di iOS atau web pada tahap ini.

1. **Standar Pemrograman dan Pengembangan**

* Pengembangan aplikasi menggunakan Flutter harus mengikuti best practices Flutter, termasuk penggunaan state management (seperti Provider) untuk pengelolaan status aplikasi dan memastikan kode tetap bersih dan mudah dipelihara.
* UI/UX harus mengikuti pedoman desain Android dan Material Design dari Google untuk menjaga konsistensi dan kemudahan penggunaan.

1. **Keterbatasan Anggaran dan Waktu**

Karena keterbatasan anggaran, pengembangan aplikasi difokuskan pada fitur-fitur inti terlebih dahulu, tanpa menambah fitur atau layanan tambahan yang memerlukan biaya lebih. Pengujian aplikasi dilakukan dengan perangkat Android standar tanpa perlu membeli perangkat tambahan atau layanan berbayar untuk tahap pengembangan awal.

## Dokumentasi Pengguna

Dokumentasi pengguna untuk aplikasi **Kantinku** akan mencakup berbagai komponen untuk membantu pengguna memahami cara menggunakan aplikasi dengan efektif. Komponen dokumentasi yang akan disediakan adalah sebagai berikut:

1. **User Manual (Panduan Pengguna)**

* Dokumen panduan lengkap yang menjelaskan alur penggunaan aplikasi, mulai dari proses pendaftaran, pemesanan menu, hingga pembayaran dan pelacakan status pesanan.
* Panduan ini akan tersedia dalam format digital (PDF) dan dapat diakses melalui aplikasi atau situs resmi Kantinku. **Link ke User Manual**

1. **Online Help (Bantuan Online)**

* Fitur *Frequently Asked Questions (FAQ)* di dalam aplikasi, mencakup pertanyaan dan solusi umum terkait kendala yang sering dialami pengguna, seperti kesalahan login, transaksi gagal, atau pembatalan pesanan.
* Bantuan online disajikan dalam bentuk teks yang mudah diakses kapan pun oleh pengguna.

1. **Tutorial Interaktif**

* Panduan interaktif yang memandu pengguna baru melalui langkah-langkah dasar seperti memilih menu, menambahkan pesanan, melakukan pembayaran, dan melacak status pesanan.
* Tutorial ini akan muncul saat pertama kali aplikasi digunakan, dan juga dapat diakses kembali melalui menu bantuan.

1. **Video Tutorial**

* Video demonstrasi yang menjelaskan penggunaan aplikasi secara praktis dan visual, mulai dari login hingga pengambilan pesanan.
* Video akan diunggah di platform seperti YouTube atau situs resmi dan ditautkan langsung dalam aplikasi.

1. **Support Contact** **(Kontak Dukungan)**

* Informasi kontak layanan pelanggan (*Customer Support*) akan tersedia dalam aplikasi, termasuk email, atau nomor telepon.
* Waktu respon dan jam operasional akan dicantumkan dengan jelas agar pengguna mengetahui kapan mereka dapat menghubungi tim bantuan.

# Kebutuhan Antarmuka Eksternal

## User Interfaces

Antarmuka pengguna (*User Interface/ UI*) aplikasi **Kantinku** dirancang untuk memberikan pengalaman penggunaan yang intuitif, efisien, dan konsisten. Berikut adalah karakteristik logis dari masing-masing antarmuka pengguna:

1. **Tampilan Halaman Utama**

* **Komponen** :
* Menu navigasi utama (seperti : Beranda, Pesanan, Profil)
* Banner atau slide yang menampilkan promosi atau penawaran khusus
* Daftar kategori menu yang dapat dipilih pengguna
* **Fungsi** :
* Menyediakan navigasi mudah antar halaman
* Menampilkan rekomendasi dan produk popular

1. **Halaman Pemesanan**

* **Komponen** :
  + - * Daftar menu makanan, minuman, atau snack
      * Tombol untuk menambahkan item ke keranjang
      * Subtotal dan total harga yang diperbarui otomatis
  + **Fungsi** :
* Memungkinkan pengguna memilih dan menyesuaikan item pesanan
* Memberikan estimasi harga secara real-time

1. **Halaman Pembayaran**

* **Komponen** :
* Formulir metode pembayaran (e-wallet, transfer bank, dsb.)
* Tombol konfirmasi pembayaran
* Rincian pesanan yang dipesan
  + **Fungsi** :
* Memfasilitasi proses pembayaran digital secara aman

1. **Tampilan Notifikasi**
   * + - **Komponen :**
   * Pesan pop-up atau banner untuk memberi tahu status pesanan (misalnya, "Pesanan Anda Sedang Diproses").
   * Tombol untuk menutup notifikasi

* **Fungsi** :
* Memberikan pembaruan status secara real-time kepada pengguna tentang pesanan mereka

1. **Pengaturan Akun**

* **Komponen** :
* Formulir edit profil (seperti nama, email, password)
* Tombol untuk keluar dari akun
* **Fungsi** :
* Mengelola data pribadi dan preferensi pengguna

1. **Standar Desain UI**

* **Panduan Desain**: Mengikuti pedoman desain **Material Design** dari Google untuk memastikan konsistensi UI/UX di seluruh aplikasi.
* **Tombol & Fungsi Umum**:
* Setiap layar memiliki tombol standar seperti Batal, Simpan, dan Kembali
* Semua layar akan memiliki akses ke **Bantuan** melalui tombol ikon atau menu di bagian atas atau bawah layar.

1. **Standar Pesan Kesalahan**

* **Pesan Kesalahan** :
* Ditampilkan dalam format jelas dan mudah dipahami
* Memberikan opsi untuk memperbaiki atau melanjutkan
* **Contoh**: "Gagal masuk. Periksa kembali email dan kata sandi Anda."

1. **Tampilan Responsif**

* Desain antarmuka disesuaikan dengan berbagai ukuran layar perangkat Android untuk menjaga konsistensi pengalaman pengguna di semua tipe ponsel.

1. **Contoh Tampilan Layar (Mockup)**

Mockup atau desain antarmuka pengguna aplikasi **Kantinku** disediakan untuk memberikan gambaran visual mengenai layout dan interaksi pengguna. Mockup dapat diakses melalui tautan berikut :

[https://www.figma.com/design/PlQH4oNz0QsO1v5J7OatGr/KantinKu?node-id=106-191&t=WBgDzPF6bYLQFvTU-1](file:///C:\Users\Almaids\Documents\GitHub\globalSabtuGenap2425\Dokumen%20SRS.docx)

## Hardware Interface

Bagian ini menjelaskan karakteristik logis dan fisik dari antarmuka antara perangkat lunak **Kantinku** dan komponen perangkat keras (hardware) yang digunakan oleh pengguna. Aplikasi **Kantinku** tidak memerlukan perangkat tambahan khusus dan berjalan sepenuhnya di perangkat mobile. Berikut adalah detail antarmuka perangkat keras yang didukung :

1. **Perangkat yang Didukung**

* Jenis Perangkat : Smartphone Android
* Spesifikasi Minimum :
* RAM : 2 GB
* Penyimpanan Internal : Minimal 100 MB ruang kosong
* Layar : Ukuran minimum 5.0 inci (untuk kenyamanan penggunaan antarmuka)
* Koneksi Internet : Wi-Fi atau jaringan seluler (4G atau lebih disarankan)

1. **Interaksi Perangkat Lunak dan Perangkat Keras**

* **Layar Sentuh (Touchscreen)**  
  Aplikasi menggunakan input melalui layar sentuh untuk navigasi menu, pemesanan, dan interaksi data
* **Modul Internet (Wi-Fi / Data Seluler)**  
  Diperlukan untuk komunikasi real-time antara aplikasi dan layanan Firebase berbasis cloud
* **Speaker / Getar (Opsional)**  
  Digunakan untuk pemberitahuan status pesanan dalam bentuk suara atau getaran, tergantung pada pengaturan perangkat pengguna
* **Penyimpanan Internal**  
  Digunakan untuk menyimpan cache ringan aplikasi guna meningkatkan performa dan waktu pemuatan

1. **Kompatibilitas Perangkat**

* Aplikasi dirancang dan dioptimalkan untuk perangkat Android dengan sistem operasi minimal **versi 11 (API Level 30)**
* Aplikasi tidak memerlukan akses ke kamera, GPS, Bluetooth, atau sensor perangkat keras lainnya

1. **Protokol Komunikasi**

* Komunikasi antara aplikasi dan server dilakukan melalui koneksi internet yang aman, menggunakan protokol HTTPS untuk menjamin keamanan data pengguna

## Software Interface

Bagian ini menjelaskan antarmuka perangkat lunak (*software interface*) yang digunakan dalam pengoperasian aplikasi **Kantinku**, termasuk integrasi dengan sistem operasi, database, pustaka pihak ketiga, serta format data dan protokol komunikasi.

* **Sistem Operasi**

Android OS versi 11 (API Level 30) atau lebih baru

* **Framework & Bahasa Pemrograman**
* Flutter SDK
* Bahasa pemrograman: Dart
* **Backend & Database**
* **Firebase Realtime Database** – digunakan untuk menyimpan dan membaca data seperti menu, pesanan, dan informasi pengguna
* **Firebase Authentication** – digunakan untuk proses login, registrasi, dan manajemen akun pengguna
* **Library Pihak Ketiga**
* firebase\_core, firebase\_auth, firebase\_database – untuk konektivitas Firebase
* provider – untuk *state management*
* fluttertoast – untuk menampilkan pesan pop-up notifikasi kepada pengguna
* **Format Data**

Seluruh pertukaran data antar komponen dilakukan dalam format **JSON**

* **Protokol Komunikasi**

Menggunakan protokol **HTTPS** untuk memastikan komunikasi yang aman dan terenkripsi antara aplikasi dan layanan cloud (Firebase)

## Communication Interface

Bagian ini menjelaskan antarmuka komunikasi jaringan yang digunakan oleh aplikasi **Kantinku** dalam mengirim dan menerima data secara online. Komunikasi ini memastikan sinkronisasi real-time, keamanan transaksi, dan efisiensi akses data.

* **Tipe Komunikasi**

Komunikasi dilakukan secara online melalui koneksi internet

* **Protokol yang Digunakan**

**HTTPS** sebagai protokol utama, memastikan keamanan data menggunakan enkripsi SSL/TLS

* **Layanan yang Digunakan**
* **Firebase Realtime Database** – untuk sinkronisasi data pesanan, menu, dan transaksi secara langsung
* **Firebase Authentication** – untuk proses autentikasi login dan pendaftaran akun pengguna secara aman
* **Format Pesan**

Seluruh data dikirim dan diterima dalam format **JSON**, yang ringan dan mudah dibaca oleh sistem

* **Keamanan Komunikasi**

Seluruh komunikasi dilindungi dengan enkripsi **SSL/TLS** melalui protokol **HTTPS**

* **Koneksi yang Dibutuhkan**

Jaringan internet minimal 3G, dengan rekomendasi koneksi **4G atau lebih** untuk performa optimal

# Functional Requirement

<Area ini menggambarkan pengorganisasian persyaratan fungsional untuk produk dengan fitur sistem, layanan utama yang disediakan oleh produk>

<Tulis Kebutuhan Fungsional / Functional Requirement disini>

Diawali dengan membuat daftar kebutuhan fungsional P/L, lengkap dengan ID dan penjelasan jika perlu. Bisa dibuat dalam bentuk tabel.

| **ID** | **Kebutuhan Fungsional** | **Penjelasan** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Use Case Diagram

## Melakukan Pemesanan dan Pembayaran

4.1.1 Deskripsi Use Case

User login ke sistem, melihat menu, membuat dan membayar pesanan, kemudian mengambil pesanan yang telah diverifikasi oleh admin.

4.1.2 Stimulus and Respon

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1. User melakukan login ke sistem |  |
|  | 1. Sistem memverifikasi akun dan menampilkan halaman utama |
| 1. User melihat menu |  |
|  | 1. Sistem menampilkan daftar menu dari KantinKu |
| 5. User membuat pesanan |  |
|  | 6. Sistem menyimpan pesanan dan menampilkan rincian pesanan |
| 7. User membayar pesanan |  |
|  | 8. Sistem menyimpan bukti pembayaran dan verifikasi |
| 1. User mengambil pesanan di kantin |  |
|  | 1. Sistem menandai pesanan sebagai telah diambil |

## Mengelola Pemesanan dan Pembelian

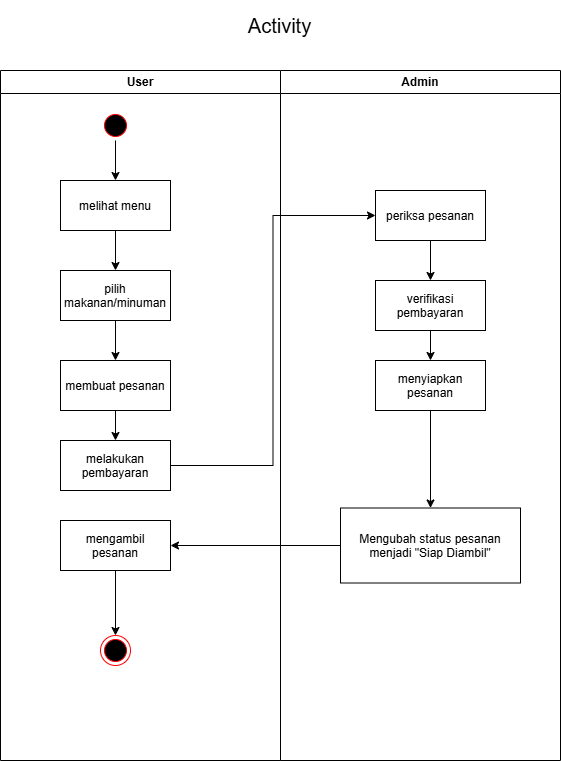
* + 1. Deskripi Use Case

Admin melakukan login ke sistem, memeriksa status pesanan, memverifikasi pembayaran yang dilakukan oleh mahasiswa, dan mengubah status pesanan setelah diproses.

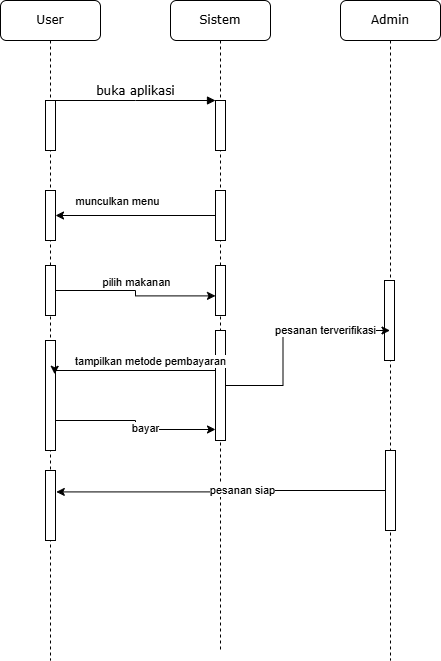
* + 1. Stimulus and Respon

|  |  |
| --- | --- |
| *Action by user* | *Response from system* |
| 1. Admin melakukan login ke sistem |  |
|  | 1. Sistem memverifikasi login dan menampilkan halaman utama admin |
| 1. Admin memeriksa status pesanan User |  |
|  | 1. Sistem menampilkan daftar pesanan dan status pembayaran |
| 1. Admin memverifikasi pembayaran yang diunggah User |  |
|  | 1. Sistem mengubah status pembayaran menjadi “Telah di bayar” |
| 1. Admin mengubah status pesanan sesuai progresnya |  |
|  | 1. Sistem memperbarui status pesanan (diproses → siap diambil → selesai) |

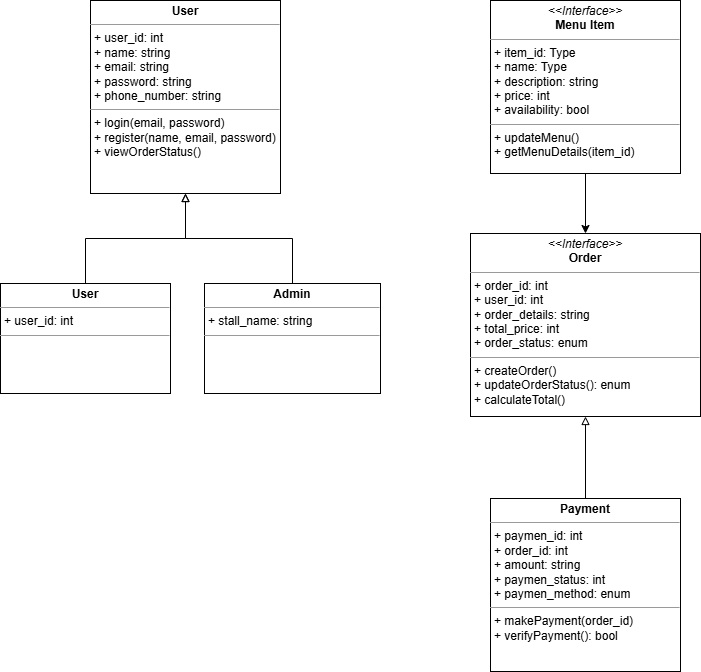
## Activity Diagram



## Squence 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*