*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 | 2.1.00 |
|  |  |  |  |
|  |  |  |  |

# Pendahuluan

## Tujuan Penulisan Dokumen

<Tulis tujuan penulisan dokumen disini.>

## Audien yang Dituju dan Pembaca yang Disarankan

<Jelaskan berbagai jenis pembaca bahwa dokumen ini ditujukan untuk, seperti pengembang, manajer proyek, staf pemasaran, pengguna, penguji, dan lainnya>

## Batasan Produk

<Berikan penjelasan singkat dari perangkat lunak yang ditentukan dan tujuannya, termasuk manfaat yang relevan, tujuan, dan sasaran. Hubungkan perangkat lunak untuk tujuan perusahaan atau strategi bisnis>

## Definisi dan Istilah

<tulis istilah dan definisikan jika ada>

* SRS : *Software Requirements Specification*, atau

Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

* IEEE : *Institute of Electrical and Electronics Engineering*

Standar internasional untuk pengembangan dan perancangan produk.

## Refrensi

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

# 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.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.
* Mengelola stok menu serta ketersediaan produk oleh pihak kantin.
* Memberikan notifikasi status pesanan (diproses, siap, selesai).

## Penggolongan Karakterik Pengguna

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

* Platform :
* Aplikasi mobile (Android)
* Hardware :
* Smartphone Android dengan minimal RAM 2 GB dan koneksi internet stabil
* Sistem Operasi :
* Android OS versi 11 (Android R) atau yang lebih baru
* Komponen Perangkat Lunak :
* Android SDK (API Level 30 atau lebih tinggi)
* Firebase (untuk autentikasi, Database, dan notifikasi real-time)

## Batasan Desain dan Implementasi

Berikut adalah beberapa batasan yang perlu dipertimbangkan 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.
2. **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.
3. **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.
4. **Penyimpanan Cloud:**
   * Data pengguna, transaksi, dan menu disimpan secara cloud-based menggunakan Firebase, sehingga aplikasi tidak mengandalkan penyimpanan lokal atau database internal perangkat.
5. **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.
6. **Pembayaran Digital:**
   * Aplikasi akan mengintegrasikan sistem pembayaran digital melalui gateway pihak ketiga (misalnya, Midtrans, DANA, atau OVO). Integrasi ini harus mematuhi persyaratan teknis dan keamanan dari penyedia pembayaran tersebut.
7. **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.
8. **State Management:**
   * Pengelolaan status aplikasi akan menggunakan Provider, yang merupakan metode state management yang sederhana dan banyak digunakan di Flutter. Hal ini akan memudahkan pemeliharaan aplikasi seiring berjalannya waktu, menjaga agar data tetap konsisten di seluruh aplikasi.
9. **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.
10. **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)**:
   * Panduan pengguna lengkap yang menjelaskan cara menggunakan aplikasi **Kantinku**, mulai dari pendaftaran, pemesanan menu, hingga pembayaran dan pelacakan status pesanan.
   * **Link ke User Manual** (Jika dokumen dalam format PDF atau platform lain).
2. **Online Help (Bantuan Online)**:
   * Fitur **FAQ (Frequently Asked Questions)** di dalam aplikasi yang mencakup pertanyaan umum dan solusi terkait masalah yang sering dihadapi pengguna, seperti masalah login, kesalahan transaksi, atau cara mengubah pesanan.
   * Tersedia dalam bentuk teks yang dapat diakses kapan saja melalui aplikasi.
3. **Tutorial Interaktif**:
   * Tutorial singkat yang memandu pengguna baru melalui langkah-langkah dasar penggunaan aplikasi, seperti cara memilih menu, melakukan pembayaran, dan melacak pesanan.
   * Tutorial ini dapat diakses pada pertama kali pengguna membuka aplikasi atau melalui menu bantuan di dalam aplikasi

.

1. **Video Tutorial**:
   * Video yang memberikan demonstrasi langsung tentang bagaimana menggunakan aplikasi dengan langkah-langkah praktis, dari mulai pemesanan hingga menerima pesanan.
   * Video akan diunggah dan bisa diakses melalui link di dalam aplikasi.
2. **Support Contact**:
   * Pengguna dapat menghubungi **Customer Support** untuk mendapatkan bantuan lebih lanjut melalui email, nomor telepon, atau chat langsung di aplikasi.
   * Waktu respon dan jam operasional akan dicantumkan dalam dokumentasi untuk memastikan pengguna tahu kapan mereka dapat menghubungi tim support.

# Kebutuhan Antarmuka Eksternal

## User Interfaces

Antarmuka pengguna (User Interface, UI) aplikasi **KantinKu** dirancang untuk memastikan pengalaman pengguna yang intuitif dan efisien. Berikut adalah deskripsi logis tentang karakteristik setiap antarmuka antara produk perangkat lunak dan pengguna:

1. **Tampilan Halaman Utama**:
   * **Komponen**:
     + Menu navigasi utama (seperti **Home**, **Menu**, **Pesanan**, **Akun**)
     + Banner atau slide yang menampilkan promosi atau penawaran khusus
     + Daftar kategori menu yang dapat dipilih pengguna
   * **Fungsi**:
     + Pengguna dapat dengan mudah menavigasi antar halaman.
     + Menampilkan rekomendasi atau item terpopuler yang memudahkan pengguna memilih makanan.
2. **Halaman Pemesanan**:
   * **Komponen**:
     + Daftar menu yang dapat dipilih
     + Tombol untuk menambahkan item ke keranjang
     + Penampilan subtotal dan total harga yang terus diperbarui
   * **Fungsi**:
     + Pengguna dapat memilih item makanan, menambahkannya ke keranjang, dan melihat total harga yang dihitung otomatis.
     + Pilihan untuk mengubah jumlah atau menghapus item di keranjang.
3. **Halaman Pembayaran**:
   * **Komponen**:
     + Formulir untuk memasukkan informasi pembayaran (seperti metode pembayaran digital, dll.)
     + Tombol untuk mengonfirmasi pembayaran
     + Rincian pesanan
   * **Fungsi**:
     + Pengguna dapat memilih metode pembayaran, dan menyelesaikan transaksi.
4. **Tampilan Notifikasi**:
   * **Komponen**:
     + Pesan pop-up atau banner untuk memberi tahu status pesanan (misalnya, "Pesanan Anda Sedang Diproses").
     + Tombol **Tutup** untuk menutup notifikasi
   * **Fungsi**:
     + Memberikan pembaruan status secara real-time kepada pengguna tentang pesanan mereka.
5. **Pengaturan Akun**:
   * **Komponen**:
     + Formulir untuk memperbarui informasi akun (seperti nama, email, kata sandi)
     + Tombol untuk keluar dari akun
   * **Fungsi**:
     + Pengguna dapat mengelola informasi pribadi dan preferensi mereka.
6. **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 aplikasi akan menampilkan tombol standar seperti **Batal**, **Simpan**, dan **Kembali** di posisi yang konsisten.
     + Semua layar akan memiliki akses ke **Bantuan** melalui tombol ikon atau menu di bagian atas atau bawah layar.
7. **Standar Pesan Kesalahan**:
   * **Pesan Kesalahan**: Setiap kesalahan (misalnya, saat pengguna gagal login atau kesalahan pembayaran) akan ditampilkan dalam format pesan kesalahan yang jelas dan mudah dipahami, dengan opsi untuk memperbaiki masalah atau melanjutkan.
   * **Contoh**: "Gagal masuk. Periksa kembali email dan kata sandi Anda."
8. **Tampilan Responsif**:
   * **Desain Responsif**: UI akan disesuaikan untuk berbagai ukuran layar perangkat, memastikan pengalaman pengguna yang konsisten di semua jenis ponsel Android.
9. **Contoh Tampilan Layar (Mockup)**:
   * Screenshots atau mockup antarmuka pengguna akan disertakan dalam dokumen terpisah untuk memberikan gambaran yang lebih jelas tentang layout dan desain UI aplikasi.

## Hardware Interface

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## Software Interface

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communication Interface

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# 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

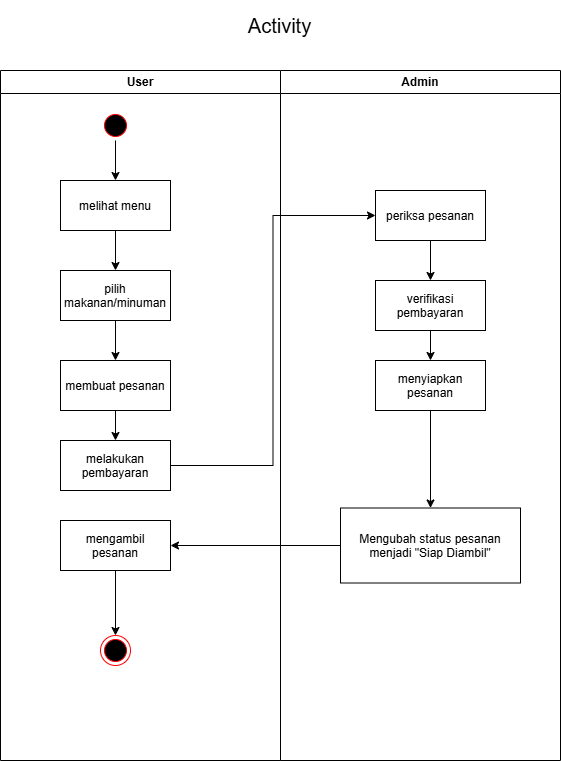
* + 1. Deskripi Use Case

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

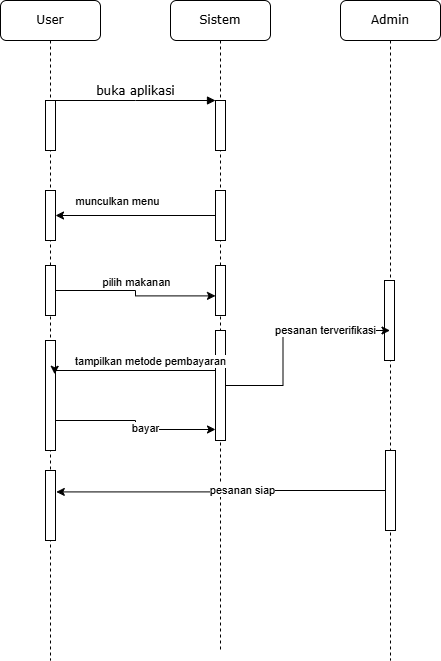
* + 1. Stimulus and Respon

|  |  |
| --- | --- |
| *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) |

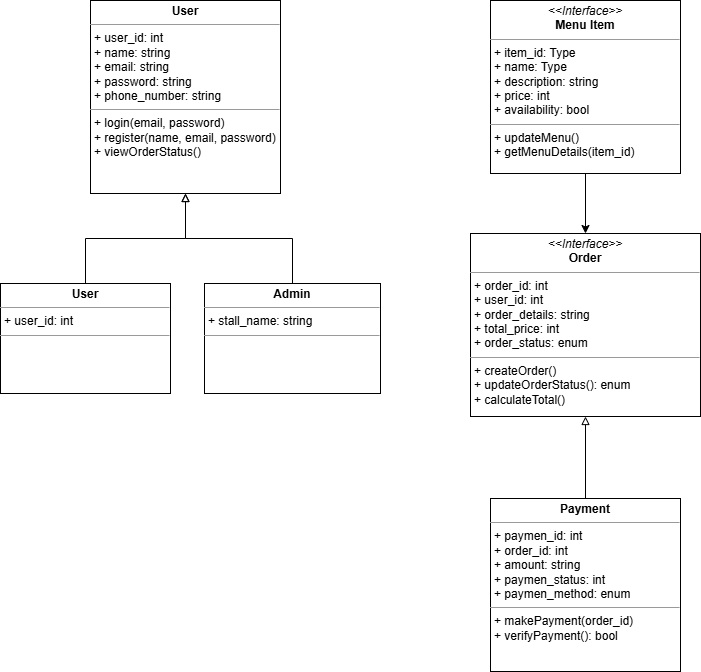
## Activity Diagram



## 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*