*Software Requirements Specification*

for

*Restaurant Point of Sale*

Version 1.0 approved

Prepared by

1907156130 - Muhammad Fazlie

10th September 2022

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 3

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 Error! Bookmark not defined.

4.1 Use Case Diagram 6

4.2 Nama Use Case 1 6

4.3 Nama Use Case 2 6

4.4 Class Diagram 6

5. Non Functional Requirements Error! Bookmark not defined.

Revision History

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

# Pendahuluan

## Tujuan Penulisan Dokumen

Tujuan pembuatan dokumen ini untuk mendeskripsikan spesifikasi website Point of Sale (POS) Resto.

## Audien yang Dituju dan Pembaca yang Disarankan

Dokumen ini ditujukan untuk developer.

## Batasan Produk

Sistem yang dibangun dapat melakukan operasi CRUD untuk pesanan pelanggan dan daftar menu. Sistem juga dapat menampilkan grafik penjualan.

## 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.

## Referensi

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

# Deskripsi Keseluruhan

## Deskripsi Produk

Sistem yang akan dibuat merupakan sebuah aplikasi point of sale untuk rumah makan dimana nantinya aplikasi tersebut dapat digunakan oleh kasir dan pelayan untuk mencatat pesanan pelanggan serta pemilik untuk melihat data penjualan.

Sistem ini nantinya dapat memudahkan proses yang terjadi di suatu rumah makan dengan mengotomasisasi proses-proses yang harus dikerjakan secara manual.

## Fungsi Produk

Untuk kasir dan pelayan, sistem ini akan berfungsi dalam penctatatan pesanan pelanggan. Sistem ini juga akan memudahkan pemilik dalam melihat total penjualan dalam satu hari, satu bulan, ataupun satu tahun.

## Penggolongan Karakterik Pengguna

Tabel 1 Karakteristik Pengguna

| **Kategori Pengguna** | **Tugas** | **Hak Akses ke aplikasi** | **Kemampuan yang harus dimiliki** |
| --- | --- | --- | --- |
| Kasir | Mencatat transaksi penjualan harian, Menginput Pesanan Pelanggan | Insert, Update Data | Entry Data Transaksi Penjualan |
| Pelayan | Menginput Pesanan Pelanggan | Insert, Update Data | Entry Data Pesanan |
| Pemilik | Memanipulasi Data Menu, Pesanan | Insert, Update, Delete, Data | Manupulasi Data |

## Lingkungan Operasi

Aplikasi yang dibuat meupakan aplikasi berbasis web yang dapat digunakan menggunakan perangkat desktop, laptop, ios, android, dan setiap system operasi yang memiliki web browser.

## Batasan Desain dan Implementasi

Sistem yang dibuat merupakan aplikasi berbasis web yang dibangu dengan menggunakan Bahasa pemrograman JavaScript dengan menggunakan framework/library React.

## Dokumentasi Pengguna

<Daftar komponen dokumentasi pengguna (seperti user manual, on-line help, dan tutorial) yang akan disampaikan bersama dengan perangkat lunak yang akan dikirim>

# Kebutuhan Antarmuka Eksternal

## User Interfaces

Aplikasi ini akan memiliki sebuah panel navigasi yang berada pada sebelah kiri layar. Pada sidebar navigation, pengguna dapat pergi ke dashboard untuk melihat grafik penjualan, menu order untuk melihat data pesanan serta menambahkan data pesanan baru, menu ‘menu’ untuk melihat daftar menu makanan dan minuman.

## Hardware Interface

Sistem ini merupakan sebuah aplikasi berbasis website, artinya pengguna dapat mengakses website melalui perangkat apapun yang memiliki web browser. Dikarenakan aplikasi yang dibuat berbasis web, pengguna perlu koneksi internet untuk mengakses aplikasi ini.

## Software Interface

Tampilan front end aplikasi ini dibuat dengan menggunakan framework/library React. React adalah sebuah library JavaScript yang digunakan untuk membangun user interface yang interaktif. Tidak hanya React, website ini juga dibangun dengan menggunakan framework dari CSS, tailwind. dengan database firestore dari firebase.

## 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.>*

# Kebutuhan Fungsional

| **ID** | **Kebutuhan Fungsional** | **Penjelasan** |
| --- | --- | --- |
| 1 | Authentication | Pengguna dapat login ke dalam system |
| 2 | Authorization | Terdapat role untuk setiap pengguna |
| 3 | Melihat Grafik Penjualan | Pengguna dapat melihat grafik penjualan |
| 4 | *Create* data pesanan | Pengguna aplikasi dapat melakukan pembuatan/input data pesanan |
| 5 | *Read* data pesanan | Pengguna aplikasi dapat melihat data pesanan |
| 6 | *Update* data pesanan | Pengguna aplikasi dapat mengubah data pesanan yang telah diinputkan |
| 7 | *Delete* data pesanan | Pengguna aplikasi dapat menghapus data pesanan |
| 8 | *Create* data menu | Pengguna aplikasi (pemilik/admin) dapat menambah daftar menu |
| 9 | *Read* data menu | Pengguna aplikasi dapat melihat daftar menu |
| 10 | *Update* data menu | Pengguna aplikasi dapat mengubah data menu yang telah diinputkan |
| 11 | *Delete* data menu | Pengguna aplikasi dapat menghapus data menu. |

## Use Case Diagram

## Nama Use Case 1

4.1.1 Deskripsi Use Case

<desripsikan / jabarkan mengenai use case ini >

4.1.2 Stimulus and Respon

<menyediakan daftar aksi yang dilakukan oleh user dan respon dari sistem.>

|  |  |
| --- | --- |
| Action by user | Response from system |
| 1 |  |
|  | 2 |
| 3 |  |
|  | 4 .. |

4.1.4 *Activity Diagram*

## Nama Use Case 2

<Sama seperti di atas, dan seterusnya sesuai jumlah use case yang didapatkan>

## Class Diagram

<*identifikasi kelas yang terkait dan hubungannya pada sistem yang dikembangkan*>

# Kebutuhan Non Fungsional

| **ID** | **Parameter** | **Kebutuhan** |
| --- | --- | --- |
| 1 | Availability | Aplikasi harus dapat beroperasi 24 jam perhari |
| 2 | Reliability | Aplikasi harus |
| 3 | Ergonomy | Aplikasi harus mudah dioperasikan untuk pengguna biasa |
| 4 | Portability | Aplikasi harus bisa digunakan di semua device yang memiliki browser |
| 5 | Memory |  |
| 6 | Response time | Aplikasi harus memiliki response time yang cepat |
|  | Safety | N/A |
|  | Security |  |
|  |  |  |
|  |  |  |

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*