*Software Requirements Specification*

for

<Sistem Informasi Penerima Zakat Fitrah>

Version 1.0 approved

Prepared by

<1122140082 – Yudi Meiza>

<Nim 2 – Aceva Arie Sadewa>

<Nim 3 – Dany Decamdo>

<21/05/2025>

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** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini menyajikan spesifikasi kebutuhan fungsional dan non-fungsional untuk sistem SiPZF. Tujuan utamanya adalah menjadi acuan bagi tim pengembang, penguji, dan pemangku kepentingan dalam merancang, mengimplementasikan, dan memverifikasi aplikasi mobile Flutter yang mendukung pencatatan dan administrasi penerima zakat fitrah.

## Audien yang Dituju dan Pembaca yang Disarankan

* **Pengembang/Developer Flutter** (menerjemahkan kebutuhan ke kode)
* **Tim QA/Pengujian** (menguji fungsi aplikasi dan validasi data)
* **Stakeholder/Admin Zakat** (pihak yang menggunakan aplikasi dan memverifikasi hasil penyaluran)

## Batasan Produk

Aplikasi melayani pencatatan penerima dan penyaluran **zakat fitrah** serta mendukung **perhitungan dan penyaluran zakat maal** berdasarkan nisab yang ditentukan. Fitur meliputi:

* Pencatatan master data penerima dan muzakki untuk kedua jenis zakat.
* Definisi paket zakat fitrah (beras/uang) dan nisab zakat maal (persentase harta).
* Transaksi penyaluran untuk fitrah dan maal, termasuk kalkulasi jumlah yang wajib dibayar.
* Laporan ringkasan penyaluran fitrah dan maal.

## Definisi dan Istilah

* **SiPZF**: Singkatan dari “Sistem Informasi Penerima Zakat Fitrah”.
* **Penerima**: Orang yang berhak menerima zakat fitrah.
* **Muzakki**: Orang yang menyalurkan zakat fitrah.
* **Jenis Zakat**: Paket zakat (beras atau uang) sesuai ketentuan.
* **Penyaluran**: Transaksi pemberian zakat kepada penerima.

## Refrensi

* Dokumen IEEE SRS Template
* Dokumentasi Firebase Firestore

# Deskripsi Keseluruhan

## Deskripsi Produk

SiPZF adalah aplikasi mobile berbasis Flutter yang memungkinkan admin (operator) mengelola data penerima, muzakki, jenis zakat, serta mencatat penyaluran secara realtime menggunakan Firebase Firestore.

## Fungsi Produk

* Manajemen data Master:
  1. **Penerima**: tambah, ubah, hapus, tampilkan data.
  2. **Muzakki**: tambah, ubah, hapus, tampilkan data.
  3. **Jenis Zakat**: definisi paket zakat, tambah, ubah, hapus.
* Transaksi Penyaluran:
  1. Form input penyaluran: pilih penerima, muzakki, jenis zakat, kuantitas.
  2. Simpan riwayat penyaluran dengan tanggal.
* Laporan dan tampilan ringkasan penyaluran.

## Penggolongan Karakterik Pengguna

<Identifikasi berbagai golongan pengguna yang terkait dengan produk yang dikembangkan>

Tabel 1 Karakteristik Pengguna

| **Kategori Pengguna** | **Tugas** | **Hak Akses ke aplikasi** | **Kemampuan yang harus dimiliki** |
| --- | --- | --- | --- |
| Admin/Operator | Mengelola master data dan mencatat penyaluran | CRUD penerima, muzakki, jenis\_zakat, penyaluran | Admin/Operator |
| Auditor | Melihat laporan dan riwayat penyaluran | Hanya Read-Only | Auditor |
|  |  |  |  |
|  |  |  |  |

## Lingkungan Operasi

* **Platform**: Android 7.0+ dan iOS 12.0+
* **Database**: Firebase Firestore
* **Pengembangan**: Flutter SDK ≥2.18, Dart SDK ≥2.18

## Batasan Desain dan Implementasi

* UI mengikuti panduan Material Design.
* Autentikasi minimal (opsional login Google) – tidak wajib untuk MVP.
* Semua data tersimpan secara realtime di Firestore.

## Dokumentasi Pengguna

* Panduan instalasi Flutter dan konfigurasi Firebase.
* User manual singkat di dalam aplikasi (on-boarding/tutorial).

## Struktur Database

Desain database Firestore mencakup koleksi master dan transaksi berikut:

1. **penerima** (Master Penerima)
   * id (string, auto)
   * nama (string)
   * nik (string)
   * alamat (string)
   * jumlah\_keluarga (number)
2. **muzakki** (Master Muzakki)
   * id (string, auto)
   * nama (string)
   * email (string)
   * telepon (string)
   * alamat (string)
3. **jenis\_zakat** (Master Jenis Zakat)
   * id (string, auto)
   * nama\_paket (string)
   * satuan (string)
   * jumlah\_per\_orang (number)
   * kategori (string) // "fitrah" atau "maal"
   * nisab (number) // hanya untuk kategori "maal": ambang harta minimal
   * rate\_persen (number) // hanya untuk kategori "maal": persentase zakat maal
4. **penyaluran** (Transaksi Penyaluran)
   * id (string, auto)
   * tanggal (timestamp)
   * penerimaId (reference ke penerima/{id})
   * muzakkiId (reference ke muzakki/{id})
   * jenisZakatId (reference ke jenis\_zakat/{id})
   * kuantitas (number)
   * total\_nilai (number) // dihitung otomatis: untuk fitrah = kuantitas \* jumlah\_per\_orang; untuk maal = total\_harta \* rate\_persen/100
   * keterangan (string)

# Kebutuhan Antarmuka Eksternal

## User Interfaces

* **Login Screen** (opsional): Input email/password atau Google Sign-In.
* **Home Dashboard**: Menu navigasi bottom bar (Master Data, Penyaluran, Laporan).
* **List Screen** untuk: Penerima, Muzakki, Jenis Zakat (ListView + FAB tambah data).
* **Form Screen** untuk tambah/edit data dengan TextField, Dropdown.
* **Detail Screen** untuk melihat riwayat penyaluran per penerima.

## Hardware Interface

* Menggunakan kamera/perangkat penyimpanan lokal (opsional untuk upload bukti penyaluran).

## Software Interface

* **Firestore SDK**: koneksi ke koleksi penerima, muzakki, jenis\_zakat, penyaluran.
* **Authentication**: Google Sign-In plugin (opsional).

## 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.>

* Protokol HTTPS untuk komunikasi dengan Firebase.

# 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** |
| --- | --- | --- |
| F1 | CRUD Penerima | Admin dapat Tambah/Ubah/Hapus dan tampilkan daftar penerima. |
| F2 | CRUD Muzakki | Admin dapat Tambah/Ubah/Hapus dan tampilkan daftar muzakki. |
| F3 | CRUD Jenis Zakat | Admin dapat Tambah/Ubah/Hapus definisi paket zakat. |
| F4 | Form Penyaluran | Admin memilih penerima, muzakki, jenis zakat, dan kuantitas. |
| F5 | Simpan Penyaluran dengan Timestamp | Sistem mencatat tanggal dan waktu penyaluran. |
| F6 | Laporan Penyaluran | Menampilkan ringkasan penyaluran berdasarkan tanggal. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Use Case Diagram

<*Gambarkan use case diagramnya dari functional requirement yang didapatkan*>

<digambarkan di dokumen terpisah>

## 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*>

# 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** |
| --- | --- | --- |
| N1 | Performance | Aplikasi memuat data master dan daftar penyaluran dalam < 3 detik pada koneksi 4G. |
| N2 | Reliability | Data tersimpan dan sinkron ke Firestore dengan akurasi 100% per penyaluran. |
| N3 | Usability | UI mengikuti standar Material Design, mudah dipahami oleh admin pemula. |
| N4 | Portability | Mendukung Android 7.0+ dan iOS 12.0+. |
| N5 | Security | Komunikasi dengan Firestore melalui HTTPS, akses data diatur melalui aturan Firestore Rules. |
| N6 | Maintainability | Kode bersifat modular (pattern MVC/MVVM), dokumentasi inline, dan struktur folder rapi. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

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*