*Software Requirements Specification*

for

*Qurbanku*

Version 1.0 approved

Prepared by

1122140056 - Agung Rizki Saputra

1122140107 – Gusnaldi Luthfi

1122140037 – Wahyu Arliansyah

<date created>

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 disusun untuk menggambarkan secara detail kebutuhan fungsional dan non-fungsional dari aplikasi Qurban. Dokumen ini bertujuan untuk menjadi acuan utama bagi tim pengembang, pemangku kepentingan, dan pihak terkait lainnya dalam proses analisis, perancangan, pengujian, dan implementasi aplikasi.

Tujuan Utama :

* Mendefinisikan fitur dan fungsi aplikasi Qurban secara jelas.
* Memberikan pemahaman yang sama antara pengembang dan pengguna.
* Menjadi landasan dalam proses pengujian dan validasi sistem.

## Audien yang Dituju dan Pembaca yang Disarankan

* Admin : Memahami kebutuhan pengguna, mengelola pengguna, dan transaksi dalam aplikasi.
* User : Sebagai pengguna akhir yang mencari, memilih, dan memesan hewan qurban melalui aplikasi.

## Batasan Produk

* **Platform Terbatas**: Aplikasi hanya tersedia di platform **Android** untuk tahap awal. Dukungan iOS akan direncanakan pada fase pengembangan selanjutnya.
* **Koneksi Internet Aktif:** Aplikasi tidak mendukung penggunaan secara offline. Seluruh fitur memerlukan koneksi internet yang stabil untuk sinkronisasi data secara real-time dengan Firebase.
* **Metode Pembayaran Terbatas**: Pembayaran hanya dapat dilakukan melalui **metode online** (transfer, e-wallet), dan **cetak bukti pembayaran fisik** disediakan oleh admin secara manual.
* **Bahasa Antarmuka**: Antarmuka aplikasi hanya mendukung **Bahasa Indonesia**, tanpa opsi bahasa lain.

## 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.

* Qurban : Proses penyembelihan hewan sebagai bentuk ibadah yang dilakukan

umat Islam pada Idul Adha.

* User : Pengguna aplikasi dari kalangan masyarakat umum yang

ingin memesan hewan qurban.

* Admin : Pengelola sistem yang bertanggung jawab terhadap data pengguna,

stok hewan, dan transaksi.

* Pemesanan : Proses pembelian hewan qurban melalui aplikasi,

baik secara langsung maupun titip.

## Refrensi

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

# Deskripsi Keseluruhan

## Deskripsi Produk

Aplikasi Qurban Online hadir sebagai solusi modern untuk memudahkan umat Muslim dalam melaksanakan ibadah qurban. Dengan memanfaatkan teknologi digital, aplikasi ini memungkinkan pengguna untuk memilih, memesan, dan membayar hewan qurban secara online tanpa harus datang langsung ke lokasi penjual. Proses pemilihan hewan, **pembayaran(Model bisnis )**, hingga laporan penyembelihan dapat dipantau langsung melalui aplikasi. Kemudahan, keamanan, dan transparansi menjadi komitmen utama kami dalam mendukung pelaksanaan ibadah qurban yang lebih baik dan efisien.

Note:

1. keluarga ada yang bisnis kurban
2. Sudah kenal dengan peternak yang ada / wilayah.
3. Kebutuhan dari penjulan , kebutuhan pembeli

## Fungsi Produk

1. **Manajemen Hewan Qurban**

* Admin dapat menambahkan, mengedit, dan menghapus data hewan qurban (jenis, berat, harga, foto, deskripsi).
* Sistem secara otomatis akan mengurangi stok saat terjadi transaksi pembelian.

1. **Pencarian dan Pemesanan Hewan Qurban**

* Pengguna dapat menelusuri katalog hewan qurban berdasarkan filter (jenis, harga, berat).
* Pengguna dapat memilih dan memesan hewan qurban langsung dari aplikasi.

1. **Manajemen Transaksi**

* Pencatatan transaksi pembelian secara digital.
* Perhitungan total harga termasuk potongan (diskon) jika berlaku.
* Menyediakan bukti transaksi (nota digital) dan fitur cetak struk oleh admin.

1. **Autentikasi Pengguna**

* Sistem login/logout untuk admin dan user menggunakan Firebase Authentication (email & password).

1. **Histori Transaksi**

* Pengguna dapat melihat riwayat pembelian sebelumnya secara rinci.
* Admin dapat mengakses laporan penjualan harian.

1. **Pelacakan Proses Qurban**

* Pengguna dapat memantau status hewan yang dipesan, dari pemesanan hingga penyembelihan (opsional/fase lanjut).

## Penggolongan Karakterik Pengguna

Aplikasi akan digunakan oleh dua golongan pengguna utama, yaitu admin dan user/masyarakat `:

Tabel 1 Karakteristik Pengguna

| **Kategori Pengguna** | **Tugas** | **Hak Akses ke aplikasi** | **Kemampuan yang harus dimiliki** |
| --- | --- | --- | --- |
| Admin | Manajemen stok & mencari sapi terbaik | CRUD Data | Basic smartphone & admin app usage |
| User/Masyarakat | Mencari hewan qurban | Beli Hewan, liat history pembelian | Pemakaian aplikasi |

## Lingkungan Operasi

Aplikasi akan beroperasi dalam lingkungan sebagai berikut:

1. Platform Aplikasi:
2. Android
3. iOS
4. Framework Frontend: Flutter (latest stable release)
5. Backend & Database: Firebase Firestore, Firebase Authentication
6. Perangkat Keras:
7. Smartphone minimal RAM 4 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. **Bahasa**: Aplikasi hanya mendukung **Bahasa Indonesia** sebagai bahasa antarmuka utama.
2. **Desain UI**: Antarmuka pengguna dirancang dengan gaya **family-friendly**, sehingga mudah dipahami dan nyaman digunakan oleh semua kalangan usia.
3. **Penyimpanan Data**: Seluruh data aplikasi disimpan secara **real-time di Firebase Database**, termasuk data pengguna dan transaksi.
4. **Koneksi Internet**: Aplikasi **tidak mendukung mode offline** dan membutuhkan **koneksi internet aktif** untuk seluruh fungsionalitasnya.
5. **Autentikasi**: Sistem login terbatas pada metode **Firebase Email & Password Authentication.**
6. **Peran Pengguna**: Aplikasi hanya mendukung **dua level akses pengguna**, yaitu **admin** dan user / masyarakat**.**
7. **Visual Branding**: Belum tersedia elemen visual resmi (seperti logo dan palet warna), sehingga **desain UI/UX bersifat minimalis dan difokuskan pada fungsionalitas** terlebih dahulu.

## 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

User Interface sementara : <https://www.figma.com/design/Pb6H6GwnJpii4D6Z6dSINw/UI-PROJECT-3?node-id=0-1&p=f&t=ZLqmogCnMFitJHFG-0>

## Hardware Interface

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

1. Perangkat yang Didukung:
2. Smartphone Android
3. Smartphone iOS
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

## Software Interface

Aplikasi Qurbanku ini akan berinteraksi dengan beberapa komponen perangkat lunak, sebagai berikut:

1. Firebase Firestore (Cloud-hosted NoSQL Database)
2. Digunakan untuk menyimpan data transaksi, produk, 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 admin dan user.
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).

# 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

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

## 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** |
| --- | --- | --- |
|  | 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*