

Business Requirements Document (BRD) - Solo Inventory Manager

1	Project Overview	2
2	Tujuan Proyek	2
3	Fitur Utama (Mobile App - Flutter)	2
4	Persyaratan Fungsional	3
4.1	Autentikasi	3
4.2	Manajemen Produk	3
4.3	Scan Barcode	3
4.4	Notifikasi	3
4.5	Laporan	3
5	Persyaratan Non-Fungsional	4
6	Stack Teknologi	4
6.1	Keuntungan Penggunaan GetX	4
7	Arsitektur Aplikasi	4
8	Integrasi dengan Supabase	5
8.1	HTTP Request dengan Supabase	5
9	Desain Database (Supabase)	6
9.1	Tabel Users	6
9.2	Tabel Products	6
9.3	Tabel Transactions	6
10	Manajemen Risiko	7

1 Project Overview

- **Nama Proyek:** Solo Inventory Manager
- **Platform:** Mobile (Flutter 3.27) dengan fokus pada iOS
- **Target Pengguna:** UMKM dengan 10-100 produk
- **Tim Pengembang:** Gheraldy Moses Tarigan
- **Periode Pengembangan:** 1 Minggu

2 Tujuan Proyek

- Memantau stok barang secara real-time
- Mencatat transaksi masuk/keluar barang
- Memberi notifikasi otomatis saat stok minimum
- Membuat laporan bulanan sederhana

3 Fitur Utama (Mobile App - Flutter)

1. Autentikasi

- Login dengan username/password
- Register akun baru
- Reset password

2. Manajemen Produk

- Tambah, edit, lihat, dan hapus produk
- Filter produk berdasarkan kategori

3. Scan Barcode

- Scan barcode untuk identifikasi produk cepat
- Input stok via kamera

4. Notifikasi

- Peringatan stok rendah

5. Laporan

- Laporan bulanan sederhana
- Statistik inventaris dasar

4 Persyaratan Fungsional

4.1 Autentikasi

- Sistem harus memungkinkan pengguna membuat akun baru dengan username dan password.
- Sistem harus memungkinkan pengguna login dengan kredensial yang sudah terdaftar.
- Sistem harus memungkinkan pengguna reset password jika lupa.
- Sistem harus menjaga sesi login pengguna.

4.2 Manajemen Produk

- Sistem harus memungkinkan pengguna menambah produk baru dengan nama, kode, harga, stok, dan kategori.
- Sistem harus memungkinkan pengguna mengedit produk yang ada.
- Sistem harus memungkinkan pengguna menghapus produk.
- Sistem harus menampilkan daftar produk dengan opsi filter dan pencarian.

4.3 Scan Barcode

- Sistem harus memungkinkan pengguna melakukan scan barcode produk.
- Sistem harus dapat mencari produk berdasarkan hasil scan barcode.
- Sistem harus memungkinkan penambahan stok dengan memindai barcode.

4.4 Notifikasi

- Sistem harus mengirim notifikasi saat stok produk mencapai batas minimum.
- Sistem harus menampilkan daftar produk dengan stok rendah.

4.5 Laporan

- Sistem harus menghasilkan laporan bulanan tentang pergerakan inventaris.
- Sistem harus menampilkan statistik dasar seperti produk terlaris dan produk dengan perputaran lambat.

5 Persyaratan Non-Fungsional

- Optimasi untuk iOS dan macOS development.
- Waktu respons aplikasi < 1 detik.
- Keamanan: Enkripsi data pengguna.
- Koneksi internet dibutuhkan karena full menggunakan Supabase.

6 Stack Teknologi

- **Frontend:** Flutter dengan komponen UI library (misalnya, Flutter UI Kit, GetX UI).
- **State Management:** GetX.
- **Database:** Supabase (cloud).
- **API:** GetX HTTP client atau Dio untuk HTTP requests ke Supabase.
- **Autentikasi:** Supabase Auth.
- **Lainnya:** Camera plugin, Barcode Scanner.

6.1 Keuntungan Penggunaan GetX

GetX sangat berguna untuk proyek ini karena:

- State management yang lebih mudah dipelajari dibandingkan BLoC.
- Menyediakan dependency injection yang ringan.
- Memiliki navigasi dan route management terintegrasi.
- Memiliki HTTP client bawaan.
- Mengurangi boilerplate code.
- Membantu membuat UI reaktif dengan lebih mudah.

7 Arsitektur Aplikasi

Aplikasi akan menggunakan arsitektur yang disarankan GetX dengan struktur folder berikut:

```
lib/  
| --- main.dart  
|-- app/  
|   '-- app.dart  
|-- core/  
|   |-- constants/
```

```

|   |-- theme/
|   |-- utils/
|   '-- values/
|-- data/
|   |-- models/
|   |-- providers/
|   |   '-- supabase_provider.dart
|   '-- services/
|       |-- auth_service.dart
|       |-- product_service.dart
|       '-- report_service.dart
|-- modules/
|   |-- auth/
|   |   |-- controllers/
|   |   |-- views/
|   |   '-- bindings/
|   |-- home/
|   |-- product/
|   |-- scanner/
|   '-- report/
'-- routes/
    '-- app_pages.dart

```

8 Integrasi dengan Supabase

Supabase akan digunakan untuk semua fungsionalitas berikut:

- **Auth:** Menggunakan Supabase Auth untuk manajemen autentikasi.
- **Database:** Penyimpanan data produk, transaksi, dan user di Supabase PostgreSQL.
- **API:** REST API untuk komunikasi antara aplikasi dan database.
- **Realtime:** Menggunakan fitur realtime subscription untuk notifikasi.

8.1 HTTP Request dengan Supabase

Aplikasi akan menjalankan HTTP request melalui Supabase SDK untuk Flutter untuk:

- Autentikasi (login, register).
- CRUD operasi pada tabel (produk, transaksi).
- Query data untuk laporan dan statistik.

9 Desain Database (Supabase)

9.1 Tabel Users

- **id:** UUID, primary key.
- **username:** text, unique.
- **email:** text, unique.
- **created_at:** timestamp.

9.2 Tabel Products

- **id:** UUID, primary key.
- **name:** text.
- **barcode:** text.
- **category:** text.
- **price:** numeric.
- **stock:** integer.
- **min_stock:** integer.
- **user_id:** UUID, foreign key.
- **created_at:** timestamp.
- **updated_at:** timestamp.

9.3 Tabel Transactions

- **id:** UUID, primary key.
- **product_id:** UUID, foreign key.
- **type:** text — “in” atau “out”.
- **quantity:** integer.
- **notes:** text.
- **user_id:** UUID, foreign key.
- **created_at:** timestamp.

10 Manajemen Risiko

- Keterlambatan karena single developer.
- Ketergantungan pada koneksi internet (Supabase).
- Kompatibilitas library kamera pada iOS.
- Kurva pembelajaran GetX.