

# SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK (SKPL)

**Nama Proyek:** Photobooth Management System (Local Network)

**Versi:** 1.0

**Status:** Draft Awal

## 1. Pendahuluan

### 1.1 Tujuan

Dokumen ini menjelaskan spesifikasi teknis dan fungsional untuk pembangunan sistem manajemen bisnis Photobooth yang beroperasi secara offline menggunakan jaringan lokal (LAN). Sistem ini bertujuan untuk memisahkan antrean pembayaran (Kasir) dan antrean layanan (Foto) secara real-time.

### 1.2 Lingkup Masalah

- Sistem harus berjalan tanpa ketergantungan internet (Offline First).
- Sistem menggunakan arsitektur *Single Codebase* (Flutter) dengan dua mode operasi: **Mode Server/Kasir** (Windows) dan **Mode Client/Verifikator** (Android).
- Komunikasi data menggunakan HTTP (REST) untuk transaksi dan WebSocket untuk pembaruan data real-time.

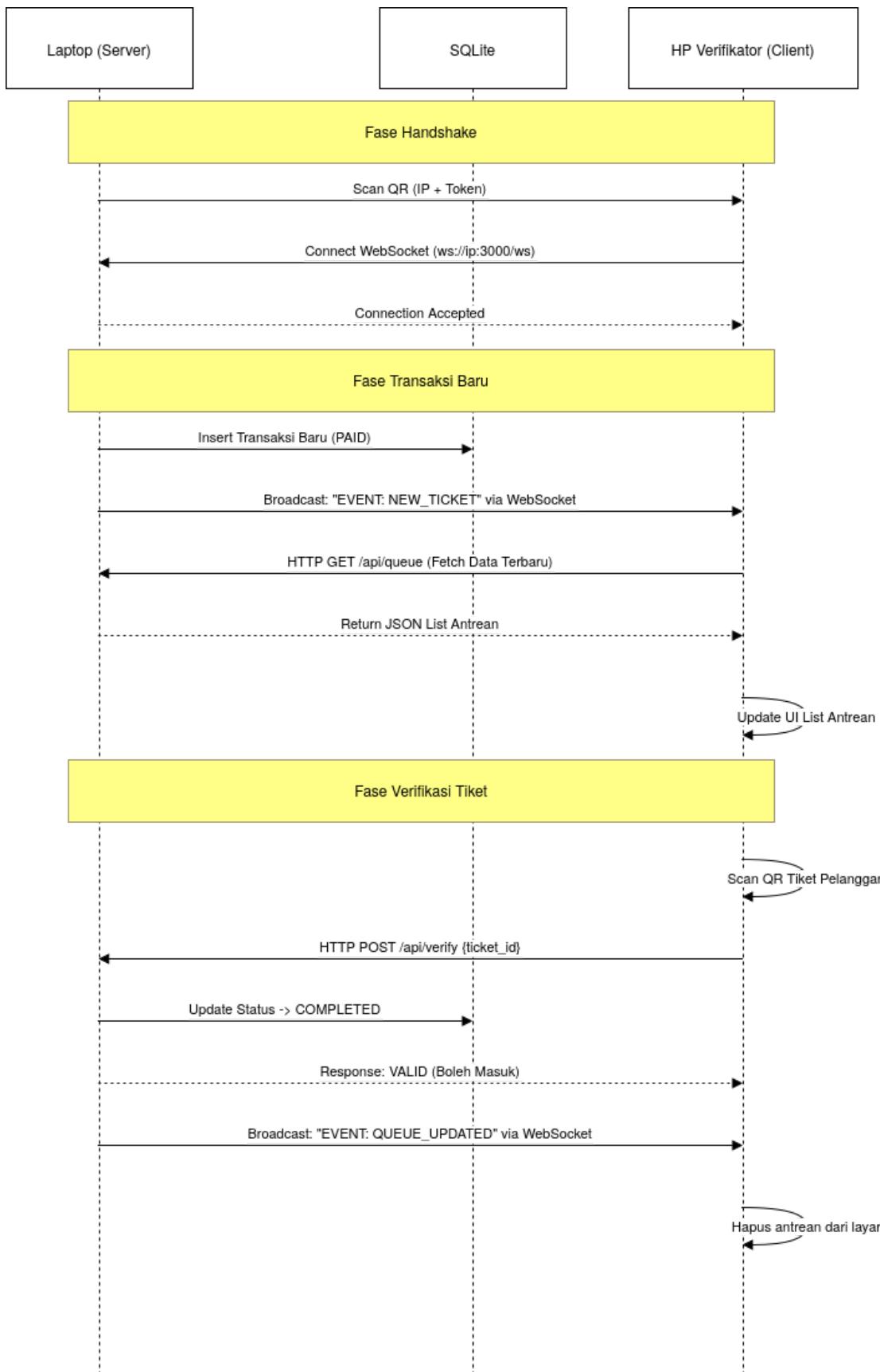
## 2. Arsitektur Sistem

### 2.1 Topologi

Sistem menggunakan topologi **Peer-to-Peer (P2P) via Wi-Fi**, dimana Laptop Kasir bertindak sebagai *Central Node (Server)*.

- **Node A (Laptop Kasir):**
  - Menjalankan Flutter Desktop (Windows).
  - Hosting HTTP Server (Alfred) pada port 3000.
  - Hosting WebSocket Server pada endpoint /ws.
  - Menyimpan Database Utama (SQLite FFI).
- **Node B (HP Verifikator):**
  - Menjalankan Flutter Mobile (Android).
  - Bertindak sebagai Client yang terhubung ke IP Node A.

### 2.2 Diagram Alur Data (WebSocket)



### **3. Kebutuhan Fungsional**

#### **3.1 Modul Kasir (Server Role)**

Modul ini aktif jika aplikasi dijalankan di Windows/Linux.

ID	Fitur	Deskripsi
K-01	<b>Embedded Server</b>	Aplikasi otomatis menjalankan server HTTP & WebSocket saat startup di background.
K-02	<b>Kelola Paket</b>	CRUD sederhana untuk daftar harga dan nama paket (disimpan di SQLite).
K-03	<b>Transaksi Baru</b>	Input pesanan, pilih paket, terima pembayaran, dan simpan status PAID.
K-04	<b>Cetak Struk</b>	Generate QR Code unik (UUID) dan cetak ke Thermal Printer via USB/Bluetooth.
K-05	<b>Pairing QR</b>	Menampilkan QR Code berisi IP Address & Secret Token untuk pairing dengan HP Verifikator.
K-06	<b>Broadcast</b>	Mengirim sinyal ke WebSocket setiap kali ada perubahan data (Insert/Update).

#### **3.2 Modul Verifikator (Client Role)**

Modul ini aktif jika aplikasi dijalankan di Android/iOS.

ID	Fitur	Deskripsi
V-01	<b>Handshake</b>	Fitur scan QR Server untuk mendapatkan konfigurasi IP

		Address & Token otentifikasi.
V-02	<b>Live Queue</b>	Menampilkan daftar antrian yang statusnya PAID secara real-time (mendengarkan trigger WebSocket).
V-03	<b>Scan Tiket</b>	Membuka kamera untuk scan QR Code pada struk pelanggan.
V-04	<b>Validasi</b>	Mengirim kode tiket ke server untuk divalidasi. Jika sukses, menampilkan detail paket.

## 4. Desain API & Protokol Komunikasi

### 4.1 Struktur Data JSON (Tiket)

```
{
  "id": "uuid-v4",
  "customer_name": "Budi",
  "package_name": "Photo + Print",
  "status": "PAID", // PAID, COMPLETED, CANCELLED
  "created_at": "2023-10-27 10:00:00"
}
```

### 4.2 WebSocket Events

Komunikasi via WebSocket hanya mengirimkan **Signal (Pemicu)**, bukan data besar, untuk menjaga performa ringan. Client yang menerima sinyal wajib melakukan *fetch* ulang via HTTP.

Event Name	Direction	Trigger Condition	Aksi Client
REFRESH_QUEUE	Server -> Client	Ada transaksi baru masuk.	GET /api/queue
TICKET_REDEEMED	Server -> Client	Ada tiket yang baru saja discan/selesai.	GET /api/queue (Hapus dari list)

## 4.3 HTTP Endpoints (REST)

### 1. Cek Status Server

- GET /health
- Response: 200 OK

### 2. Ambil Antrean (Polling/Fetch)

- GET /api/queue
- Response: [List of Transaction Objects where status = PAID]

### 3. Verifikasi Tiket

- POST /api/verify
- Body: {"ticket\_code": "...”}
- Response:
  - 200 OK: {"valid": true, "data": {...}}
  - 400 Bad Request: {"valid": false, "message": "Tiket sudah dipakai"}
  - 404 Not Found: {"valid": false, "message": "Tiket tidak ditemukan"}

## 5. Model Database (SQLite Schema)

Hanya ada 2 tabel utama untuk menjaga kesederhanaan.

### Tabel: products

- id (INTEGER, PK, Auto Increment)
- name (TEXT) - e.g., "Self Photo 15 Menit"
- price (INTEGER)

### Tabel: transactions

- uuid (TEXT, PK) - ID unik untuk QR Code
- customer\_name (TEXT, Nullable)
- product\_name (TEXT) - Snapshot nama produk saat transaksi
- product\_price (INTEGER)
- status (TEXT) - Default: 'PAID'
- created\_at (TEXT/DATETIME)
- redeemed\_at (TEXT/DATETIME, Nullable) - Diisi saat scan berhasil

## 6. Kebutuhan Non-Fungsional

### 1. Security:

- Akses ke API Server dibatasi menggunakan Authorization Header yang tokennya didapat saat scan QR Handshake.
- Aplikasi Windows harus diizinkan pada Firewall (Private Network).

### 2. Performance:

- Delay WebSocket maksimal 1 detik dalam jaringan LAN stabil.
- Scan QR Code harus diproses dalam < 2 detik.

### 3. Reliability:

- Jika koneksi WebSocket terputus (HP mati layar), Client harus memiliki mekanisme *Auto-Reconnect* saat layar menyala kembali.