# Software Design Document (SDD)

**Project Name:** Smart Financial Planner

**Module:** System Architecture & API Design

**Version:** 4.0 (Final Comprehensive)

**Date:** 2024-05-24

## 1. Pendahuluan

### 1.1 Tujuan

Dokumen ini merinci arsitektur teknis, skema database, algoritma logika bisnis, dan spesifikasi API Endpoint untuk implementasi sistem.

### 1.2 Stack Teknologi

* **Frontend:** Next.js (React Framework) + Tailwind CSS.
* **Backend:** Python (FastAPI).
* **Database:** PostgreSQL (Relational DB).
* **ORM:** SQLAlchemy (Async).
* **Scheduler:** APScheduler (Background Jobs).
* **OCR:** Tesseract / EasyOCR.

## 2. Desain Arsitektur Sistem

### 2.1 Diagram Komponen

graph TD
    User((User)) --> FE[Next.js Dashboard]
    FE -- REST API (JSON) --> BE[FastAPI Backend]
    
    subgraph Backend_Services [Backend Services Logic]
        direction TB
        BE --> Auth[Auth Service]
        BE --> Wallet[Wallet Logic]
        BE --> Trans[Transaction Logic]
        BE --> Pred[Prediction Engine]
        BE --> OCR[OCR Service]
        BE --> Scheduler[Recurring Job]
    end
    
    DB[(PostgreSQL Database)]
    
    Trans -- Atomic Updates --> DB
    Wallet -- Query Balance --> DB
    Scheduler -- Insert Pending Tx --> DB
    Pred -- Read Analytics --> DB

## 3. Desain Data (Database Schema)

### 3.1 Entity Relationship Diagram (ERD)

#### A. Tabel Users

* id (PK), email, password\_hash, full\_name, avatar\_url, created\_at.

#### B. Tabel Wallets (Dompet)

* id (PK)
* user\_id (FK)
* name (String): Nama dompet (e.g., "BCA").
* type (Enum): 'BANK', 'CASH', 'EWALLET'.
* balance (Decimal): Saldo saat ini.
* deleted\_at (DateTime, Nullable): Penanda Soft Delete.

#### C. Tabel Categories

* id (PK), user\_id (FK)
* name (String), icon (String)
* type (Enum): 'INCOME', 'EXPENSE'.
* is\_fixed\_cost (Boolean): Penanda Fixed vs Variable.
* deleted\_at (DateTime, Nullable).

#### D. Tabel Transactions (Log Utama)

* id (PK)
* user\_id (FK)
* wallet\_id (FK): Dompet Sumber.
* target\_wallet\_id (FK, Nullable): **Dompet Tujuan (Hanya untuk TRANSFER).**
* category\_id (FK)
* amount (Decimal): Nilai transaksi.
* type (Enum): 'INCOME', 'EXPENSE', 'TRANSFER'.
* date (DateTime)
* note (String)
* status (Enum): 'COMPLETED', 'PENDING' (Default untuk Recurring).
* source (Enum): 'MANUAL', 'OCR', 'RECURRING'.
* deleted\_at (DateTime, Nullable): Penanda Soft Delete.

#### E. Tabel RecurringSchedules (Jadwal)

* id (PK), user\_id (FK)
* name, amount, category\_id
* frequency (Enum): 'DAILY', 'WEEKLY', 'MONTHLY'.
* next\_run\_date (Date): Tanggal eksekusi berikutnya.
* is\_active (Boolean).

## 4. Desain Prosedural & Algoritma

### 4.1 Logika Transfer (Atomic Transaction)

Wajib menggunakan Database Transaction (db.begin() / commit()) untuk mencegah data korup.

1. **Validasi:** Cek saldo wallet\_id >= amount.
2. **Debit Source:** UPDATE wallets SET balance = balance - amount WHERE id = source.
3. **Credit Target:** UPDATE wallets SET balance = balance + amount WHERE id = target.
4. **Log:** INSERT INTO transactions (type='TRANSFER', wallet\_id=source, target\_wallet\_id=target, ...)
5. **Commit.**

### 4.2 Logika Delete (Flexible Option)

API menerima parameter query ?permanent=true/false.

* **Soft Delete (permanent=false):**
  + Revert saldo dompet (Uang dikembalikan).
  + Update deleted\_at = NOW().
* **Hard Delete (permanent=true):**
  + Revert saldo dompet.
  + Delete row dari database selamanya.

### 4.3 Logika Prediksi Kebangkrutan

Rumus:  
Sisa Hari = (Akhir Bulan - Hari Ini)  
Burn Rate = Sum(Expense Variable yg Aktif) / Hari Berjalan  
Total Pending = Sum(Transaksi Status PENDING)  
  
Prediksi Akhir = Total Aset - (Burn Rate \* Sisa Hari) - Total Pending

*Catatan: Transfer antar dompet tidak mengurangi Total Aset, jadi tidak masuk hitungan Burn Rate.*

## 5. Spesifikasi API Endpoint (Lengkap)

Semua endpoint dilindungi header Authorization: Bearer <token>.

### A. Modul Wallets

* GET /api/v1/wallets
  + *Desc:* List semua dompet aktif & Total Aset.
* POST /api/v1/wallets
  + *Body:* {name, type, initial\_balance}.
* PUT /api/v1/wallets/{id}
  + *Body:* {name, type} (Saldo tidak bisa diedit lgsg).
* DELETE /api/v1/wallets/{id}?permanent={bool}
  + *Desc:* Hapus dompet.

### B. Modul Transactions (CRUD & Transfer)

* GET /api/v1/transactions
  + *Query:* start\_date, end\_date, wallet\_id, category\_id.
* POST /api/v1/transactions
  + *Body:* {amount, type, category\_id, wallet\_id, target\_wallet\_id, date, note}.
  + *Logic:* Handle logic Income/Expense/Transfer secara otomatis.
* PUT /api/v1/transactions/{id}
  + *Desc:* Edit transaksi (Revert saldo lama -> Apply saldo baru).
* DELETE /api/v1/transactions/{id}?permanent={bool}
  + *Desc:* Hapus transaksi & kembalikan saldo.

### C. Modul Recurring & Approval

* GET /api/v1/recurring
  + *Desc:* List jadwal tagihan rutin.
* POST /api/v1/recurring
  + *Body:* {name, amount, frequency, start\_date}.
* GET /api/v1/transactions/pending
  + *Desc:* List tagihan yang menunggu konfirmasi user.
* POST /api/v1/transactions/{id}/confirm
  + *Body:* {wallet\_id, final\_amount}.
  + *Logic:* Ubah status PENDING -> COMPLETED, potong saldo dompet.

### D. Modul Categories

* GET /api/v1/categories
* POST /api/v1/categories
* DELETE /api/v1/categories/{id}

### E. Modul Smart Features

* GET /api/v1/analysis/health
  + *Return:* {status: "DANGER", score: 45, projection: -200000}.
* POST /api/v1/scan/receipt
  + *Body:* Multipart Image.
  + *Return:* JSON Draft Transaction.

*Akhir Dokumen SDD v4.0*