Oke 👍 saya bantu bikinkan rancangan project aplikasi **Pencatatan Utang Piutang** dengan **Flutter GetX CLI** dan **Supabase**.

**📱 Rancangan Project: Aplikasi Pencatatan Utang Piutang**

**🔥 Deskripsi Singkat**

Aplikasi ini digunakan untuk mencatat **utang (utang kita ke orang lain)** dan **piutang (uang kita yang dipinjam orang lain)**.  
Dibangun dengan **Flutter + GetX CLI** untuk arsitektur state management & navigasi, serta **Supabase** sebagai backend (database & autentikasi).

**🎯 Fitur Utama**

1. **Autentikasi**
   * Registrasi & login dengan email/password (Supabase Auth).
   * Reset password via email.
   * Logout.
2. **Manajemen Utang Piutang**
   * Tambah data utang/piutang:
     + Nama pihak terkait
     + Nominal
     + Tanggal transaksi
     + Jatuh tempo
     + Keterangan
     + Jenis transaksi (utang / piutang)
   * Edit & hapus data.
   * Tandai “Lunas”.
3. **Dashboard Ringkasan**
   * Total utang yang dimiliki.
   * Total piutang yang diberikan.
   * Daftar utang/piutang aktif (belum lunas).
   * Grafik histori sederhana.
4. **Detail Transaksi**
   * Riwayat pembayaran (jika cicil/lunas sebagian).
   * Upload bukti pembayaran (opsional).

**🗄️ Struktur Database (Supabase - PostgreSQL)**

**Tabel: users**

| **Kolom** | **Tipe** | **Keterangan** |
| --- | --- | --- |
| id | uuid (PK) | dari Supabase Auth |
| email | text | email user |
| created\_at | timestamp | waktu daftar |

**Tabel: transactions**

| **Kolom** | **Tipe** | **Keterangan** |
| --- | --- | --- |
| id | uuid (PK) | ID unik |
| user\_id | uuid (FK) | relasi ke users.id |
| type | enum | 'utang' atau 'piutang' |
| name | text | nama pihak terkait |
| amount | numeric | jumlah uang |
| date | date | tanggal transaksi |
| due\_date | date | jatuh tempo |
| note | text | keterangan tambahan |
| status | enum | 'belum lunas' / 'lunas' |
| created\_at | timestamp | dibuat kapan |

**Tabel: payments (opsional untuk cicilan/lunas sebagian)**

| **Kolom** | **Tipe** | **Keterangan** |
| --- | --- | --- |
| id | uuid (PK) | ID unik |
| transaction\_id | uuid (FK) | relasi ke transactions.id |
| amount | numeric | jumlah dibayar |
| date | date | tanggal pembayaran |
| note | text | catatan pembayaran |

**🏗️ Struktur Proyek (GetX CLI)**

lib/

┣ data/

┃ ┣ models/

┃ ┃ ┗ transaction\_model.dart

┃ ┃ ┗ user\_model.dart

┃ ┣ providers/

┃ ┃ ┗ supabase\_provider.dart

┣ modules/

┃ ┣ auth/

┃ ┃ ┣ controllers/auth\_controller.dart

┃ ┃ ┣ views/login\_view.dart

┃ ┃ ┗ views/register\_view.dart

┃ ┣ home/

┃ ┃ ┣ controllers/home\_controller.dart

┃ ┃ ┗ views/home\_view.dart

┃ ┣ transaction/

┃ ┃ ┣ controllers/transaction\_controller.dart

┃ ┃ ┣ views/transaction\_list\_view.dart

┃ ┃ ┣ views/add\_transaction\_view.dart

┃ ┃ ┗ views/transaction\_detail\_view.dart

┣ routes/

┃ ┗ app\_pages.dart

┣ services/

┃ ┗ auth\_service.dart

┣ utils/

┃ ┗ constants.dart

┗ main.dart

**🎨 UI/UX (Sederhana & Ringkas)**

1. **Login & Register** → form email & password.
2. **Dashboard (Home)** → menampilkan ringkasan utang & piutang.
3. **Daftar Transaksi** → listview semua utang/piutang.
4. **Tambah Transaksi** → form tambah data.
5. **Detail Transaksi** → rincian + opsi pembayaran/lunas.
6. **Profil** → info akun & logout.

**🚀 Alur Aplikasi**

1. User login → diarahkan ke dashboard.
2. User bisa tambah transaksi (utang/piutang).
3. Data tersimpan ke Supabase → ditampilkan kembali di aplikasi.
4. User bisa edit, hapus, atau tandai lunas.
5. Dashboard otomatis update total utang/piutang.

**🔧 Teknologi yang Digunakan**

* **Frontend**: Flutter 3.x
* **State Management & CLI**: GetX CLI
* **Backend**: Supabase (Auth + Database PostgreSQL + Storage)
* **Deployment**: Android/iOS (bisa juga web)

Apakah mau saya bikinkan **proposal APSI** (seperti format kuliah: latar belakang, tujuan, analisis sistem, perancangan, dll) untuk aplikasi ini biar bisa langsung dipakai sebagai dokumen resmi?