# **Author: Rakyan Krisna Dewangga**

**CASE:**

# Test Application Design

PT. XYZ adalah sebuah perusahaan fintech yang ingin mengembangkan mobile apps mereka, dalam upaya menjangkau pengguna yang lebih luas mereka ingin mengembakan aplikasi pinjaman online.

Potential High Level User Story:

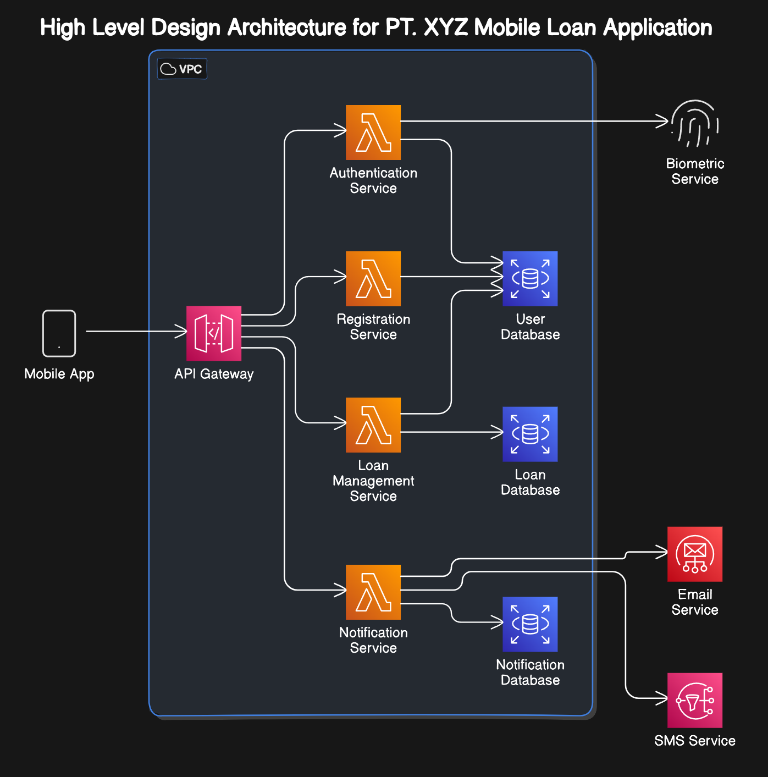
* User melakukan registrasi dengan data dir,emaili, nomor telepon dan upload foto beserta KTP
* User dapat login dengan password atau biometric (jika ada di perangkat mobilenya)
* User dapat melihat Sisa hutang dan tagihan perbulan yang harus di bayarkan (Jika ada)
* User dapat meminjam uang paling besar Rp. 12.000.000 dengan tenor maksimal 1 taun.
* User dalam proses peminjaman akan di proses dengan hasil diterima atau ditolak
* Jika pinjaman diterima maka akan ada notifikasi lewat email dan nomot telepon yang terdaftar
* User tidak dapat melakukan peminjaman uang jika sedang ada proses peminjaman dan belum di lunaskan.

Tugas Anda:

1. Buatlah high level design architecture atas project mobile apps ini.
2. Spesifikasikan design Screen Flow dan ERD atas rancangan yang ingin anda buat.
3. Buatlah detail design untuk API dengan menggunakan tools design seperti UML, ERD,flowchart etc.
4. Buatlah detail design untuk screen behavior dari mobile apps berdasarkan screen flow diatas.

**SOLUTIONS:**

1. **High-Level Architecture**

****

Berikut adalah high-level design architecture untuk project loan mobile apps dari PT. XYZ:

### High-Level Design Architecture

#### 1. **Client-Side (Mobile Application)**

* **Platforms**:
  + iOS (Swift/Flutter)
  + Android (Kotlin/Flutter)
* **Key Components**:
  + **UI/UX**: Desain antarmuka pengguna dan pengalaman pengguna untuk navigasi yang mudah dan intuitif.
  + **API Gateway Integration**: Komunikasi dengan server backend melalui API.
  + **Authentication**: Modul untuk registrasi, login, dan otentikasi biometrik (jika perangkat mendukung).
  + **Registration**: Modul untuk registrasi.
  + **Loan Management**: Modul untuk mengajukan pinjaman, melihat status pinjaman, dan mengelola tagihan.
  + **Notification**: Modul untuk menerima notifikasi push, email, dan SMS terkait status pinjaman.

#### 2. **Server-Side (Backend)**

* **Technology Stack**:
  + **Language**: NodeJS, PHP, atau bahasa lain untuk *Backend-side*
  + **Database**: PostgreSQL atau MySQL (Relational Database)
  + **Authentication Server**: OAuth 2.0 untuk manajemen otentikasi dan otorisasi
* **Key Components**:
  + **User Management**: Modul untuk mengelola registrasi pengguna, login, dan profil pengguna.
  + **Loan Processing**: Modul untuk mengelola permohonan pinjaman, verifikasi, dan persetujuan pinjaman.
  + **Payment Management**: Modul untuk mengelola tagihan dan pembayaran pinjaman.
  + **Notification Service**: Layanan untuk mengirim notifikasi email, SMS, dan push ke pengguna.
  + **API Gateway**: Pintu gerbang untuk mengelola permintaan API dari aplikasi mobile.
  + **Database**: Penyimpanan data pengguna, pinjaman, pembayaran, dan log aktivitas.

#### 3. **Security Layer**

* **Encryption**:
  + Data encryption in transit dapat menggunakan TLS/SSL
* **Authentication & Authorization**:
  + OAuth 2.0 dan JWT untuk otentikasi dan otorisasi
  + Biometric authentication untuk login (opsional)
* **Security Audits**: Penilaian keamanan rutin dan pengujian penetrasi untuk memastikan keamanan sistem

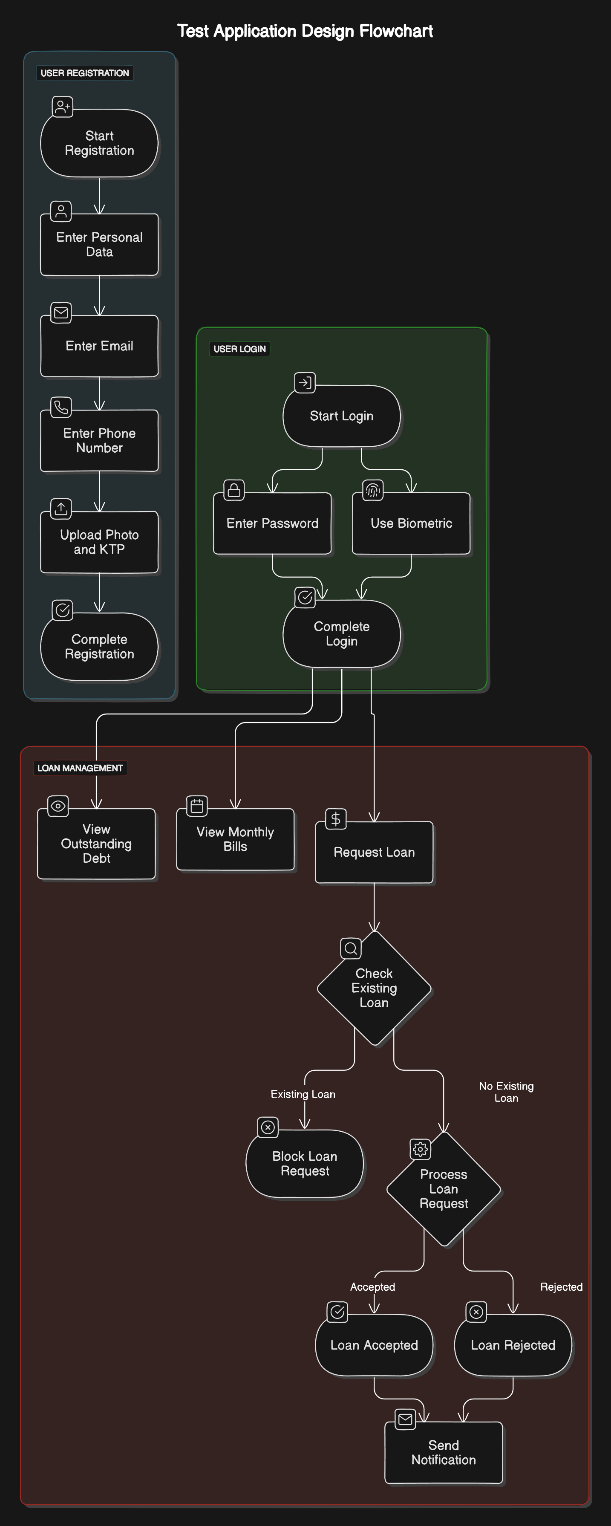
#### 4. **Third-Party Integrations**

* **Email Service Provider**: Untuk mengirim notifikasi email (misalnya, SendGrid)
* **SMS Gateway**: Untuk mengirim notifikasi SMS (misalnya, Twilio)
* **Payment Gateway**: Untuk memproses pembayaran pinjaman (misalnya, Midtrans atau Xendit)
* **Credit Scoring**: Integrasi dengan layanan pihak ketiga untuk verifikasi dan penilaian kredit pengguna jika ada

#### 5. **DevOps and Deployment**

* **CI/CD Pipeline**:
  + Jenkins, GitHub Actions, atau GitLab CI untuk integrasi dan pengiriman berkelanjutan
  + Automated testing untuk memastikan kualitas aplikasi sebelum rilis
* **Infrastructure**:
  + Cloud provider (misalnya, AWS, GCP, atau Azure) untuk hosting server backend dan database
  + Containerization menggunakan Docker dan Kubernetes untuk manajemen aplikasi yang skalabel
* **Monitoring and Logging**:
  + Monitoring aplikasi menggunakan tools seperti Prometheus dan Grafana
  + Logging dan tracing menggunakan ELK Stack (Elasticsearch, Logstash, Kibana)

1. **Flow Chart**

****

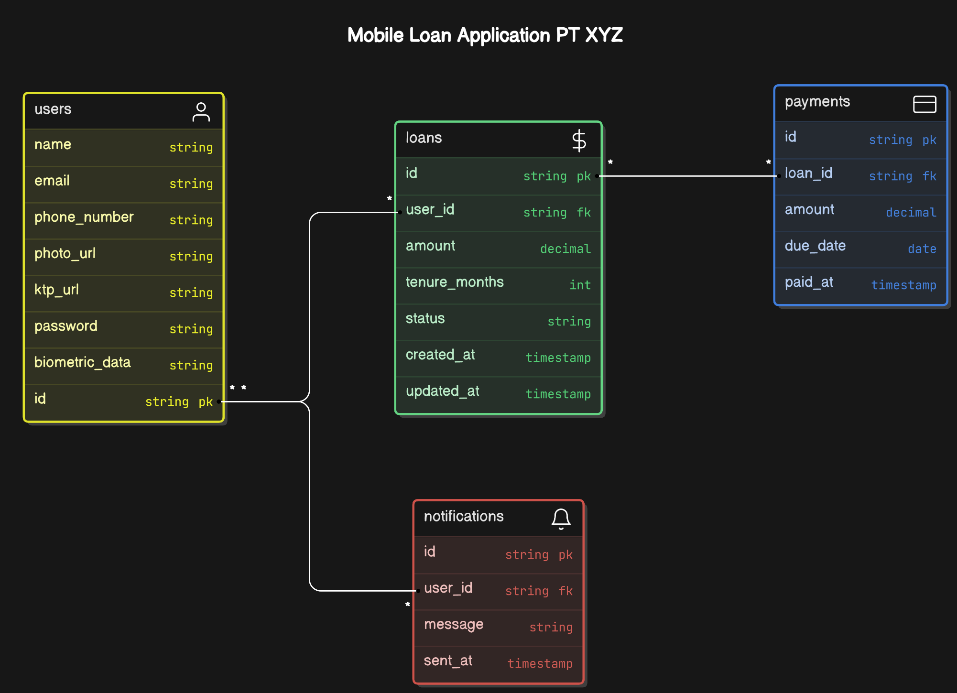
Proses dimulai dengan pengguna membuka aplikasi mobile dan memilih antara registrasi atau login. Jika pengguna baru, mereka diarahkan ke layar registrasi di mana mereka mengisi data pribadi seperti nama, email, nomor telepon, dan mengunggah foto serta KTP. Setelah data di-submit, proses registrasi dilakukan dan pengguna akan menerima notifikasi sukses atau gagal. Jika registrasi berhasil, pengguna diarahkan ke halaman login. Pengguna yang telah terdaftar bisa langsung menuju halaman login dan memasukkan kredensial mereka. Sistem akan memverifikasi kredensial tersebut, memberikan notifikasi sukses atau gagal login. Jika login berhasil, pengguna masuk ke halaman dashboard.

Di halaman dashboard, pengguna dapat melihat status hutang mereka dan tagihan bulanan. Jika pengguna ingin mengajukan pinjaman, mereka dapat memilih opsi untuk mengajukan pinjaman dan diarahkan ke formulir pengajuan pinjaman. Di sini, pengguna memasukkan jumlah pinjaman yang diinginkan (maksimal Rp. 12.000.000) dan tenor (maksimal 1 tahun). Setelah formulir diisi dan di-submit, permohonan pinjaman diproses dan diverifikasi oleh sistem. Hasil verifikasi akan menentukan apakah pinjaman disetujui atau ditolak.

Jika pinjaman disetujui, pengguna menerima notifikasi melalui email, SMS, atau push notification. Pengguna kemudian dapat melihat status pinjaman mereka di halaman status pinjaman. Jika pinjaman ditolak, pengguna akan menerima pesan penolakan dan diarahkan kembali ke dashboard. Setelah pinjaman diterima, pengguna dapat mengatur pembayaran dan melunasi pinjaman mereka. Proses pembayaran dilakukan melalui modul pembayaran pinjaman, dan setelah pembayaran berhasil, pengguna menerima notifikasi sukses dan kembali ke dashboard untuk melihat status pinjaman yang diperbarui.

Seluruh alur ini memastikan bahwa pengguna dapat dengan mudah melakukan registrasi, login, mengajukan pinjaman, menerima notifikasi, dan mengelola pembayaran pinjaman mereka dengan lancar melalui aplikasi mobile PT. XYZ.

1. **ERD**

****

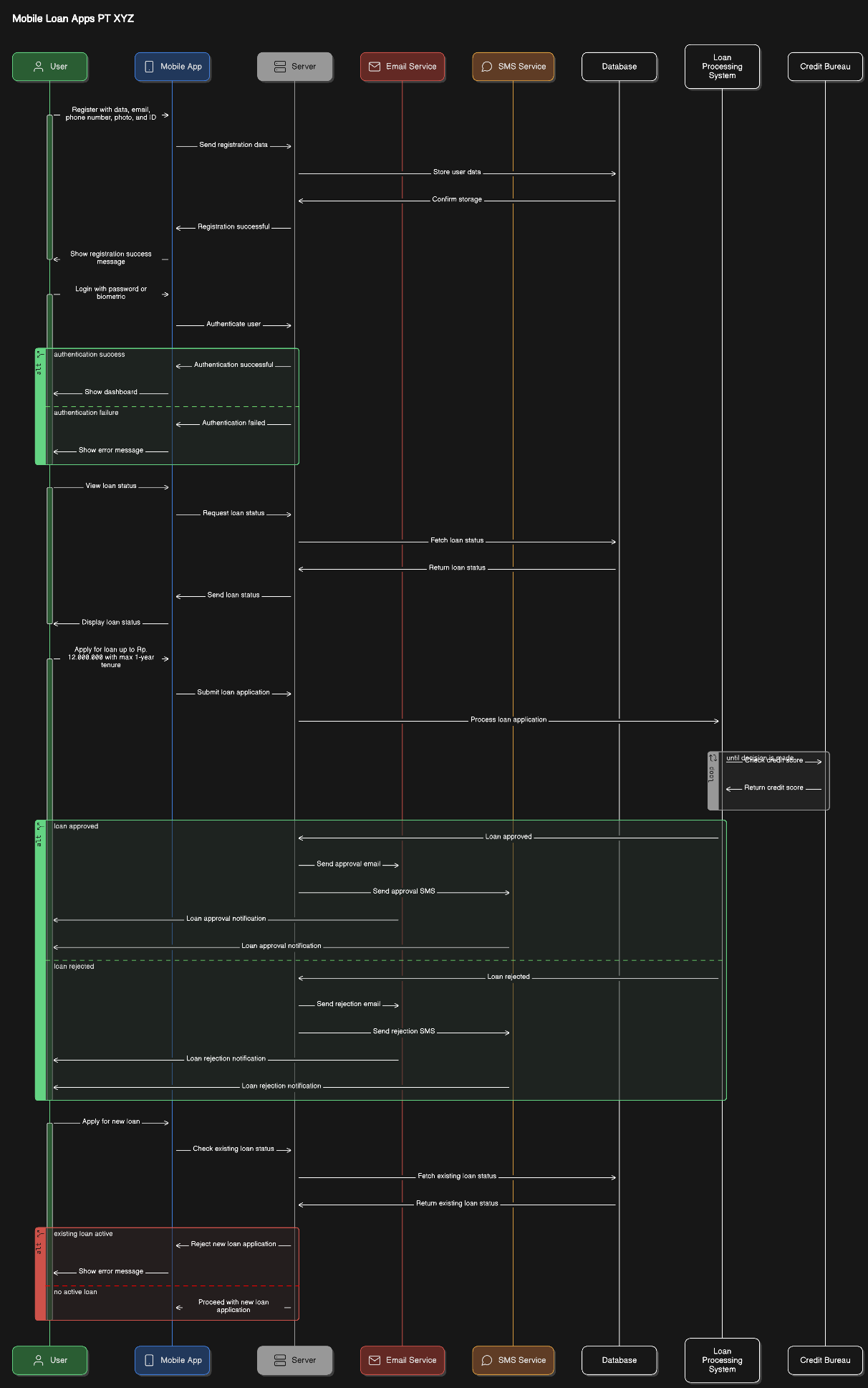
Entity-Relationship Diagram (ERD) adalah representasi visual dari data yang dibutuhkan oleh sistem, menggambarkan hubungan antara entitas-entitas dalam database. Berikut adalah penjelasan ERD untuk aplikasi mobile PT. XYZ:

### Entitas dan Atribut

1. **Users**
   * **Attributes**:
     + UserID (Primary Key)
     + Name
     + Email
     + PhoneNumber
     + Password
     + Photo
     + KTP
     + BiometricData (opsional, jika perangkat mendukung)
   * **Description**: Entitas ini menyimpan informasi pengguna yang telah terdaftar di aplikasi.
2. **Loans**
   * **Attributes**:
     + LoanID (Primary Key)
     + UserID (Foreign Key)
     + Amount
     + TenureMonths
     + ApplicationDate
     + Status (Pending, Approved, Rejected)
     + UpdateDate (Status Change Log)
   * **Description**: Entitas ini menyimpan informasi tentang permohonan pinjaman yang dilakukan oleh pengguna.
3. **Payment**
   * **Attributes**:
     + PaymentID (Primary Key)
     + LoanID (Foreign Key)
     + AmountPaid
     + DueDate
     + PaymentDate
   * **Description**: Entitas ini menyimpan informasi tentang pembayaran yang dilakukan terhadap pinjaman.
4. **Notification**
   * **Attributes**:
     + NotificationID (Primary Key)
     + UserID (Foreign Key)
     + Message
     + NotificationType (Email, SMS, Push)
     + SentDate
   * **Description**: Entitas ini menyimpan informasi tentang notifikasi yang dikirimkan kepada pengguna.

### Hubungan Antar Entitas

1. **User-Loan**
   * **Relationship**: One-to-Many (One user can have multiple loans)
   * **Description**: Setiap pengguna dapat mengajukan beberapa pinjaman, tetapi setiap pinjaman hanya terkait dengan satu pengguna.
2. **Loan-Payment**
   * **Relationship**: One-to-Many (One loan can have multiple payments)
   * **Description**: Setiap pinjaman dapat memiliki beberapa pembayaran yang terkait dengannya, tetapi setiap pembayaran hanya terkait dengan satu pinjaman.
3. **User-Notification**
   * **Relationship**: One-to-Many (One user can receive multiple notifications)
   * **Description**: Setiap pengguna dapat menerima beberapa notifikasi, tetapi setiap notifikasi hanya terkait dengan satu pengguna.
4. **Sequence Diagram**

****

Untuk detail desain screen behavior dari aplikasi pinjaman online PT. XYZ, saya akan merinci setiap langkah dari user story ke dalam screen behavior yang spesifik. Berdasarkan Sequence Diagram yang dirancang, berikut adalah desain untuk setiap screen berdasarkan alur yang dijelaskan:

### 1. **Registrasi**

**Screen 1: Registrasi**

* **Fields**: Nama, Email, Nomor Telepon, Upload Foto, Upload KTP
* **Buttons**:
  + "Submit" (aktif jika semua field terisi)
  + "Back" (kembali ke halaman utama)
* **Validation**:
  + Email harus valid
  + Nomor telepon harus valid
  + KTP harus di-upload dalam format yang diterima (JPG, PNG)
* **Error Messages**: Tampilkan pesan kesalahan jika ada field yang tidak valid atau kosong
* **Post-action**: Setelah submit berhasil, navigasi ke layar "Login"

### 2. **Login**

**Screen 2: Login**

* **Fields**: Email/Nomor Telepon, Password
* **Buttons**:
  + "Login" (aktif jika field terisi)
  + "Forgot Password?" (navigasi ke layar reset password jika ada)
  + "Back" (kembali ke layar utama)
* **Options**:
  + "Login with Biometric" (opsional, hanya muncul jika perangkat mendukung)
* **Error Messages**: Tampilkan pesan kesalahan jika login gagal
* **Post-action**: Setelah login berhasil, navigasi ke layar "Loan Dashboard"

### 3. Loan **Dashboard**

**Screen 3: Loan Dashboard**

* **Elements**:
  + "Sisa Hutang" (menampilkan jumlah hutang yang belum dilunasi)
  + "Tagihan Perbulan" (menampilkan jumlah tagihan yang harus dibayar bulan ini, jika ada)
  + "Ajukan Pinjaman" button
  + "Logout" button
* **Post-action**:
  + Jika "Ajukan Pinjaman" ditekan, navigasi ke layar "Ajukan Pinjaman"
  + Jika "Logout" ditekan, kembali ke layar "Login"

### 4. **Ajukan Pinjaman**

**Screen 4: Ajukan Pinjaman**

* **Fields**:
  + Jumlah Pinjaman (maksimal Rp. 12.000.000)
  + Tenor (maksimal 1 tahun)
* **Buttons**:
  + "Submit" (aktif jika semua field terisi)
  + "Back" (kembali ke layar "Dashboard")
* **Validation**:
  + Jumlah pinjaman tidak boleh melebihi Rp. 12.000.000
  + Tenor tidak boleh melebihi 1 tahun
* **Error Messages**: Tampilkan pesan kesalahan jika ada field yang tidak valid atau melebihi batas
* **Post-action**: Setelah submit, tampilkan layar "Proses Pinjaman"

### 5. **Proses Pinjaman**

**Screen 5: Proses Pinjaman**

* **Elements**:
  + Loading Indicator (menampilkan proses verifikasi)
  + Pesan "Permohonan sedang diproses, harap tunggu..."
* **Post-action**:
  + Jika pinjaman diterima, navigasi ke layar "Notifikasi"
  + Jika pinjaman ditolak, tampilkan pesan kesalahan dan kembali ke layar "Loan Dashboard"

### 6. **Notifikasi**

**Screen 6: Notifikasi**

* **Elements**:
  + Pesan "Pinjaman Anda telah diterima"
  + Detail pinjaman (jumlah, tenor, tanggal jatuh tempo)
  + Notifikasi email dan SMS dikirim ke user
* **Buttons**:
  + "OK" (navigasi ke layar "Loan Dashboard")

### 7. **Tidak Dapat Melakukan Peminjaman**

**Screen 7: Peminjaman Ditolak**

* **Elements**:
  + Pesan "Anda tidak dapat melakukan peminjaman baru karena masih ada pinjaman yang belum dilunasi"
  + Detail pinjaman yang belum dilunasi
* **Buttons**:
  + "OK" (navigasi ke layar "Loan Dashboard")

Desain ini mencakup alur utama dari registrasi, login, melihat status hutang, mengajukan pinjaman, proses verifikasi, hingga notifikasi hasil peminjaman. Setiap layar didesain agar mudah digunakan dengan validasi dan pesan kesalahan yang jelas untuk membantu pengguna menyelesaikan setiap tahap.