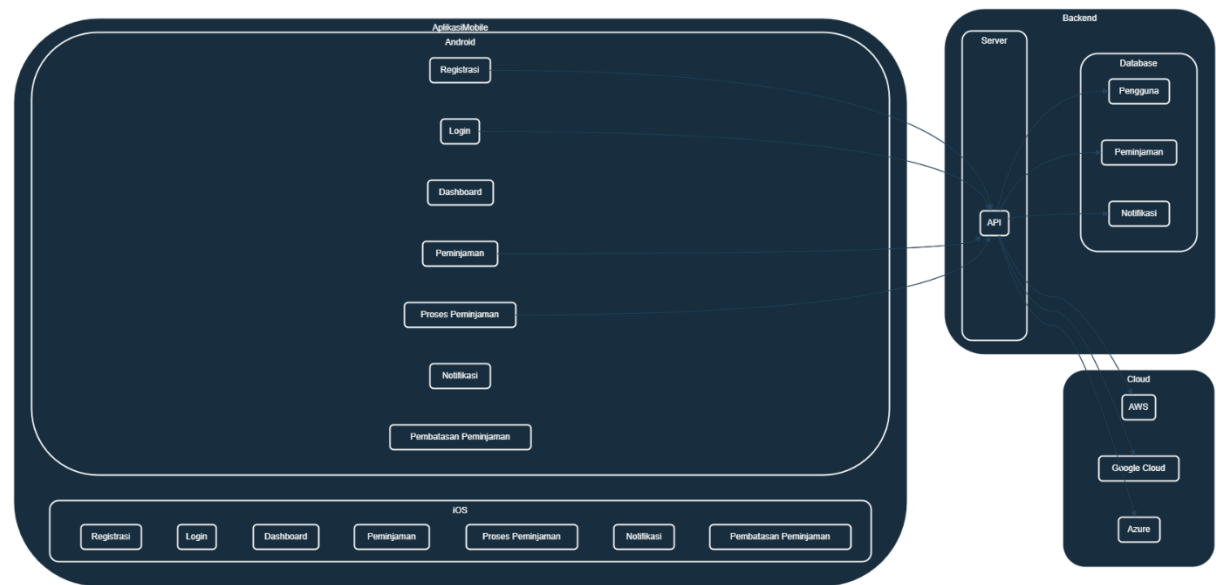


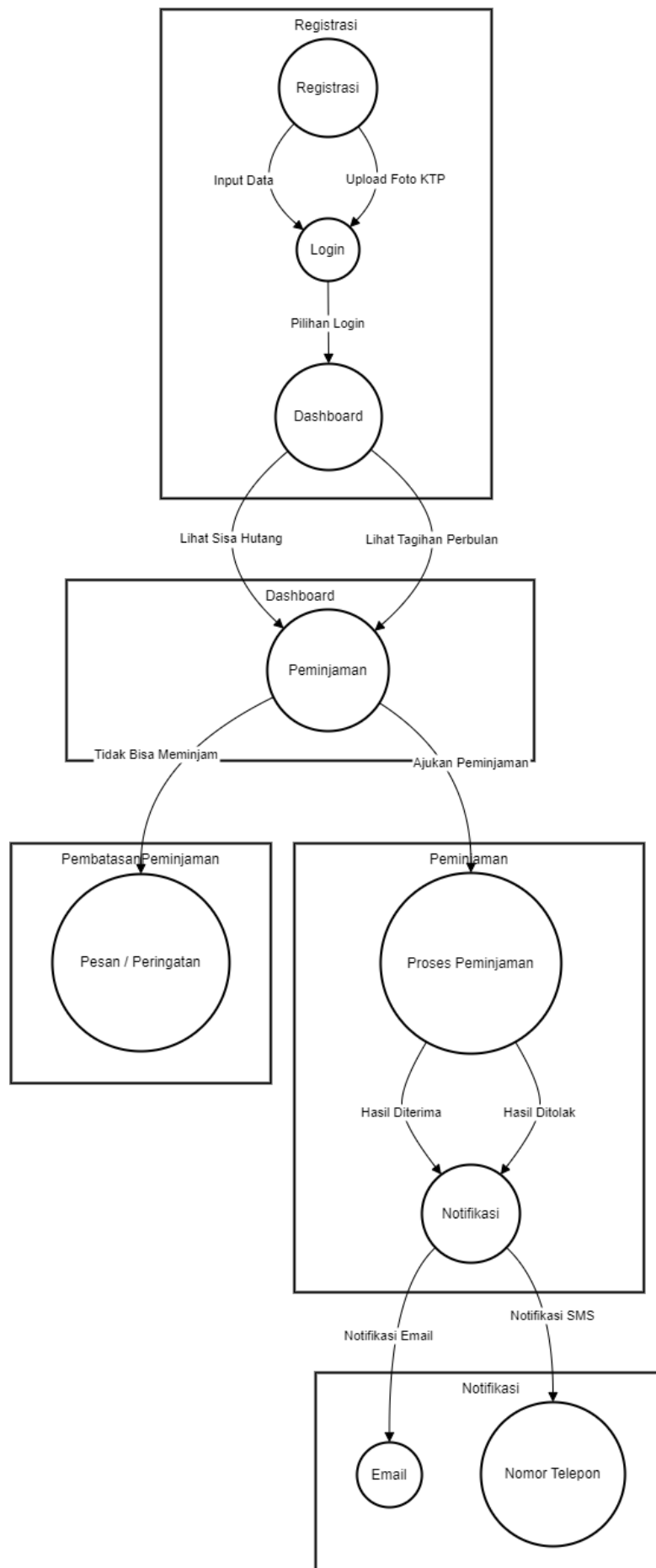
1. High Level Design (HLD)



- Aplikasi mobile dapat dibangun menggunakan platform Android dan iOS dengan pengembangan native atau cross-platform seperti Flutter atau React Native.
- Backend dapat dikembangkan dengan menggunakan bahasa pemrograman seperti Java, Kotlin, Swift, atau JavaScript dengan kerangka kerja seperti Node.js atau Ruby on Rails.
- Database yang digunakan untuk menyimpan data pengguna, informasi peminjaman, dan lainnya bisa menggunakan MySQL, PostgreSQL, atau MongoDB.
- Server dapat di-host di awan (cloud) seperti AWS, Google Cloud, atau Azure.
- Keamanan data dan transaksi harus menjadi prioritas utama, sehingga perlu diimplementasikan protokol keamanan seperti SSL/TLS, enkripsi data, dan verifikasi pengguna.
- Backend memiliki Server yang menyediakan API dan Database untuk menyimpan data pengguna, peminjaman, dan notifikasi.
- Aplikasi mobile berkomunikasi dengan Backend melalui API

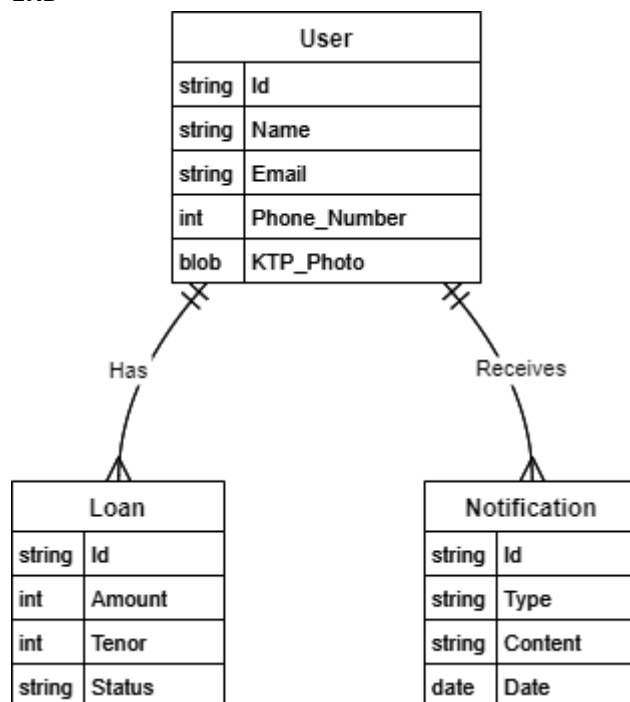
2. Screen Flow dan ERD

A. Design Screen Flow



1. Registrasi: Pengguna melakukan registrasi dengan menginput data diri dan mengupload foto KTP. Setelah berhasil registrasi, mereka akan diarahkan ke halaman Login.
2. Login: Pengguna dapat login dengan memasukkan informasi login atau menggunakan biometric (jika didukung). Setelah login, mereka akan diarahkan ke Dashboard.
3. Dashboard: Pengguna dapat melihat sisa hutang dan tagihan perbulan (jika ada) di halaman ini.
4. Peminjaman: Pengguna dapat mengajukan peminjaman dengan mengisi jumlah pinjaman dan tenor. Setelah mengajukan peminjaman, mereka akan diarahkan ke Proses Peminjaman.
5. Proses Peminjaman: Proses pengajuan peminjaman akan menghasilkan notifikasi yang mungkin berisi hasil diterima atau ditolak. Pengguna akan mendapatkan notifikasi sesuai dengan hasilnya.
6. Notifikasi: Notifikasi hasil peminjaman (diterima atau ditolak) akan dikirim melalui email atau nomor telepon yang terdaftar.
7. Pembatasan Peminjaman: Jika pengguna mencoba mengajukan peminjaman saat ada proses peminjaman yang belum dilunaskan, mereka akan menerima pesan atau peringatan.

- **ERD**

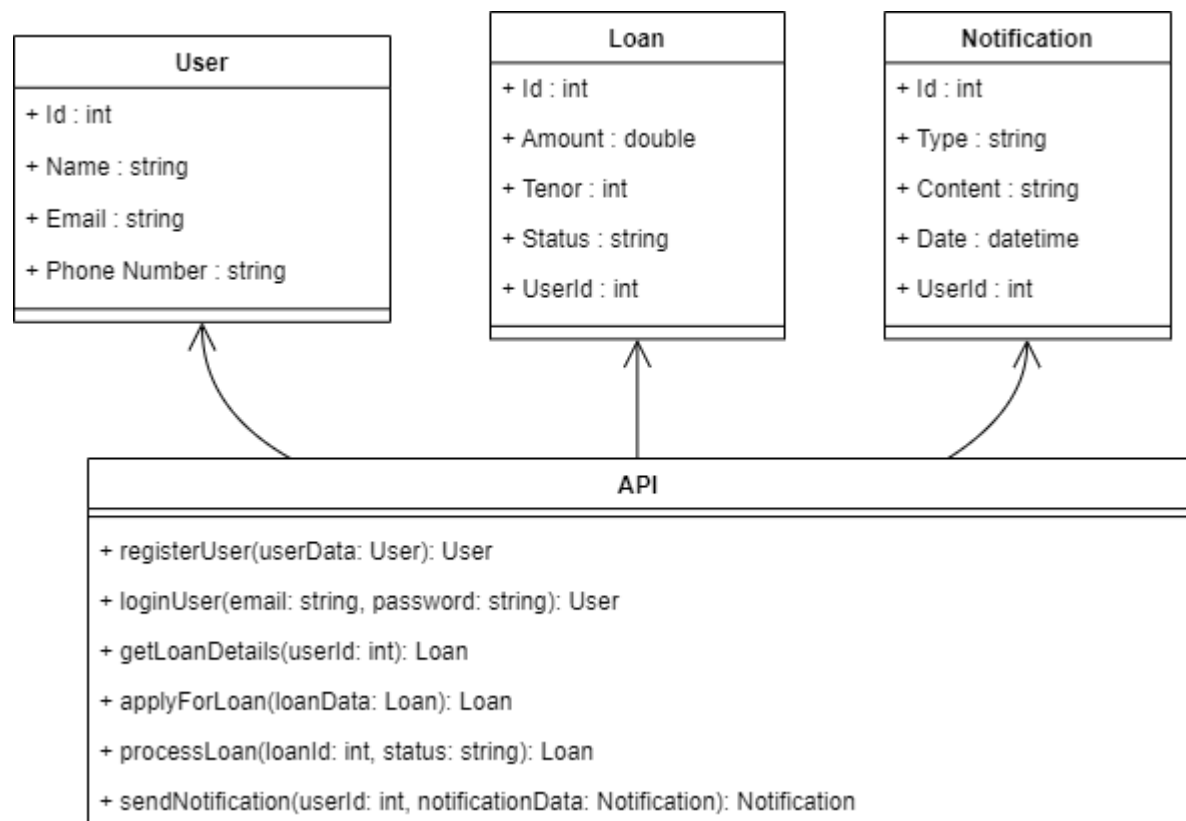


Terdapat tiga entitas utama: User (Pengguna), Loan (Peminjaman), dan Notification (Notifikasi). Hubungan antara entitas ditunjukkan dengan tanda panah dan label yang menunjukkan hubungan tersebut.

- User memiliki hubungan "Has" dengan Loan, yang berarti seorang pengguna dapat memiliki banyak peminjaman.
- User juga memiliki hubungan "Receives" dengan Notification, yang berarti seorang pengguna dapat menerima banyak notifikasi.

Setiap entitas memiliki atribut yang terkait dengan mereka, seperti Id (kunci primer), nama, email, dan lain-lain. Entitas Loan memiliki atribut Amount (jumlah peminjaman), Tenor (tenor peminjaman), dan Status (status peminjaman). Entitas Notification memiliki atribut Type (jenis notifikasi), Content (konten notifikasi), dan Date (tanggal notifikasi)

3. Detail Design API



Class API memiliki metode-metode yang sesuai dengan operasi yang ingin di lakukan pada data pengguna, peminjaman, dan notifikasi. Berikut contoh class API yang di dokumentasikan menggunakan Open API.



swagger.json

Aplikasi Pinjaman Online API 1.0 OAS 2.0

[Base URL: /api]

Dokumentasi API untuk aplikasi pinjaman online

Schemes

HTTP

Pengguna

POST

/register Registrasi Pengguna

POST

/login Login Pengguna

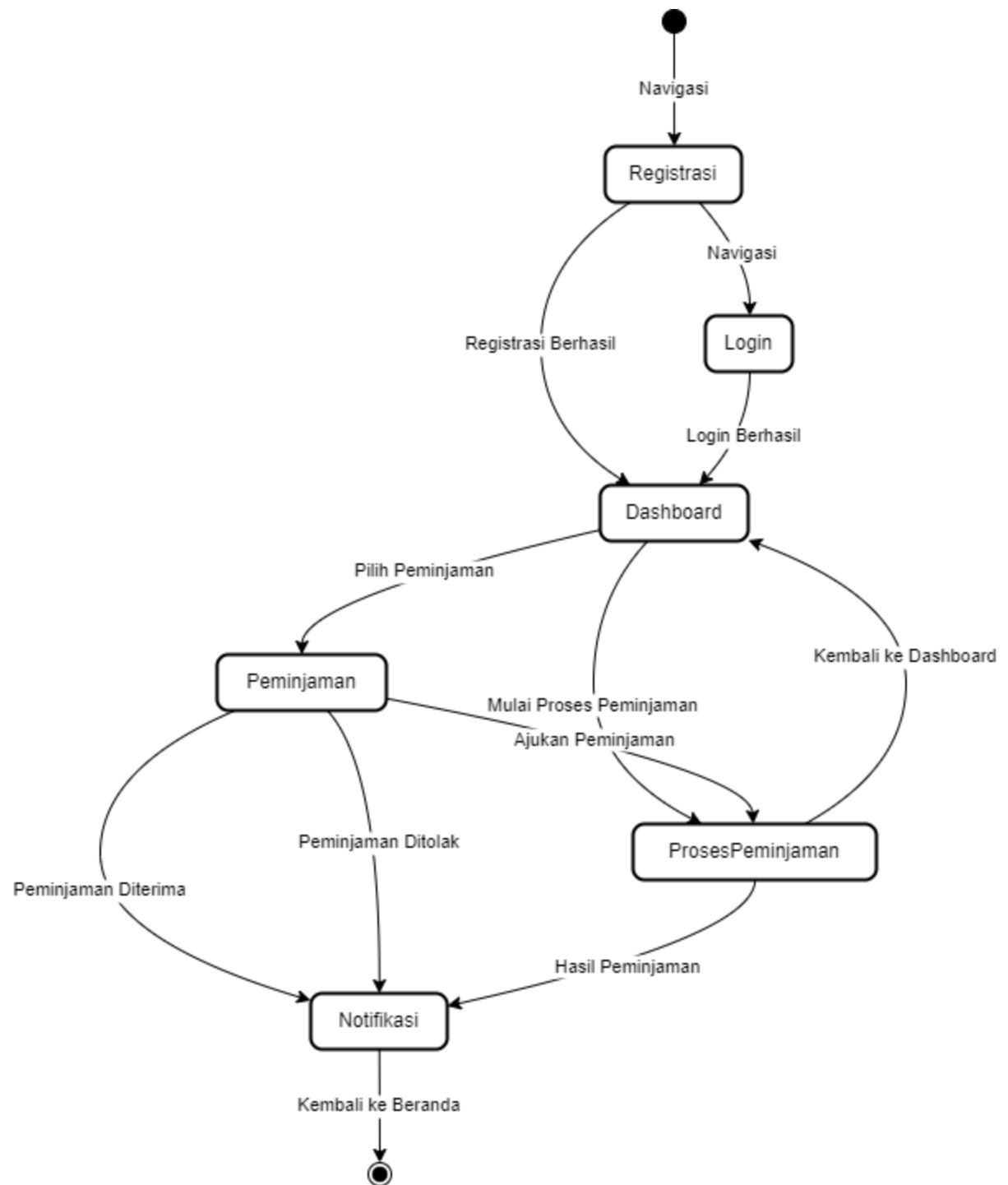
Peminjaman

GET

/loan/{userId} Mendapatkan Detail Peminjaman

Models

4. Design Screen Behavior



Setiap kotak mewakili keadaan atau halaman dalam aplikasi, dan panah menunjukkan perpindahan antara halaman-halaman tersebut. Contoh :

Pengguna mulai dari keadaan awal dan kemudian dapat berpindah ke halaman Registrasi. Setelah registrasi berhasil, mereka diarahkan ke halaman Login, dan seterusnya. Proses aliran ini menggambarkan bagaimana pengguna akan berinteraksi dengan aplikasi.