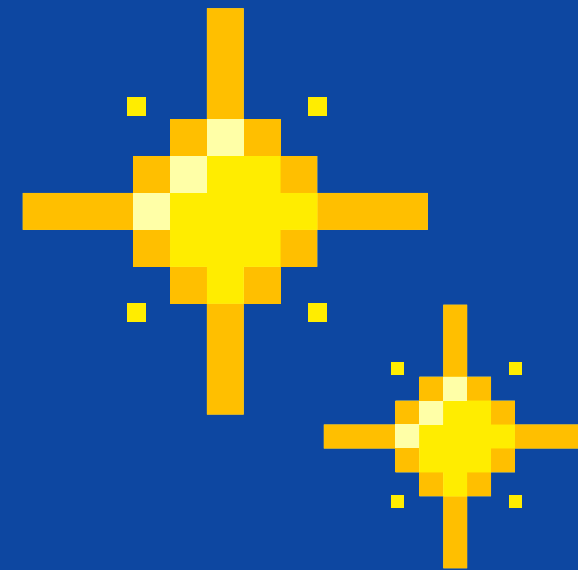
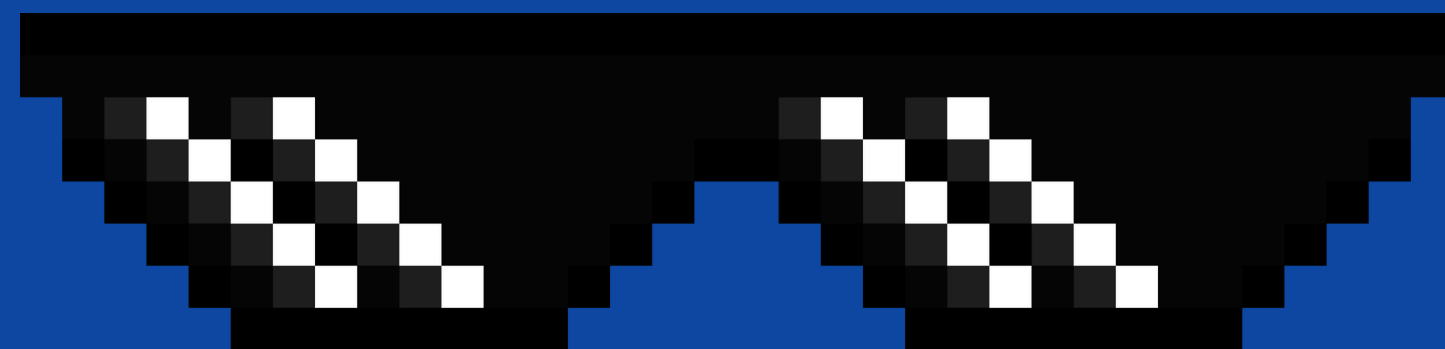


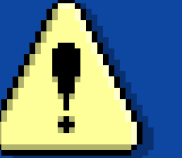
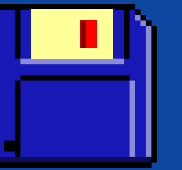
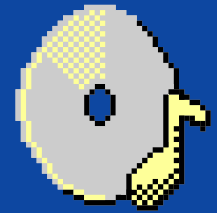
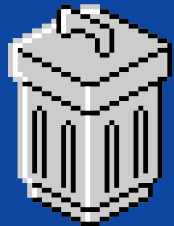
# Rakha Alghifary



SMK 3 PERCIK

XI TKJ

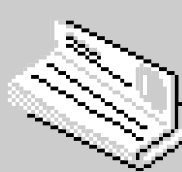
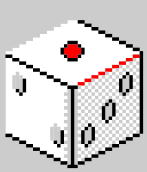
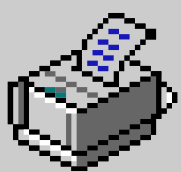




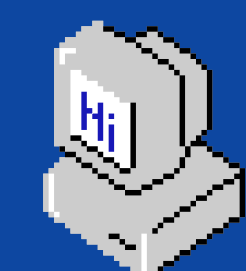
# Bukoo.



Digital Library App



11:11PM



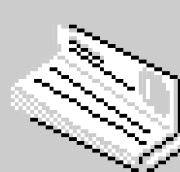
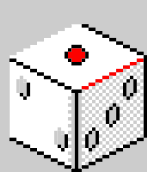
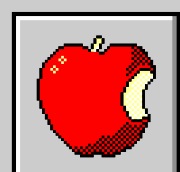
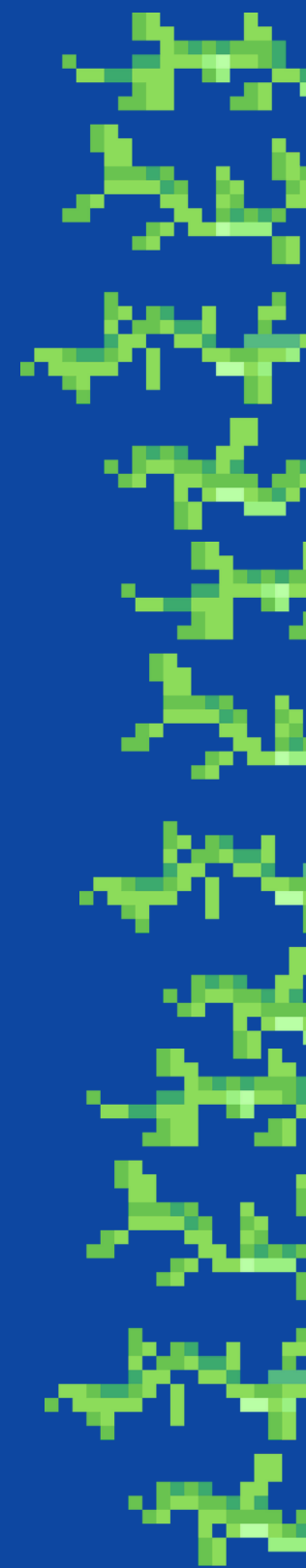
## Deskripsi

Aplikasi katalog buku digital yang memudahkan pengguna mencari, melihat detail, dan menyewa buku secara real-time.








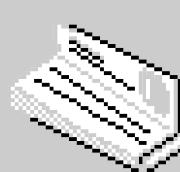
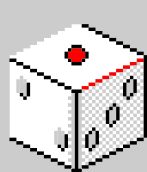
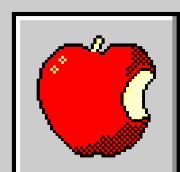
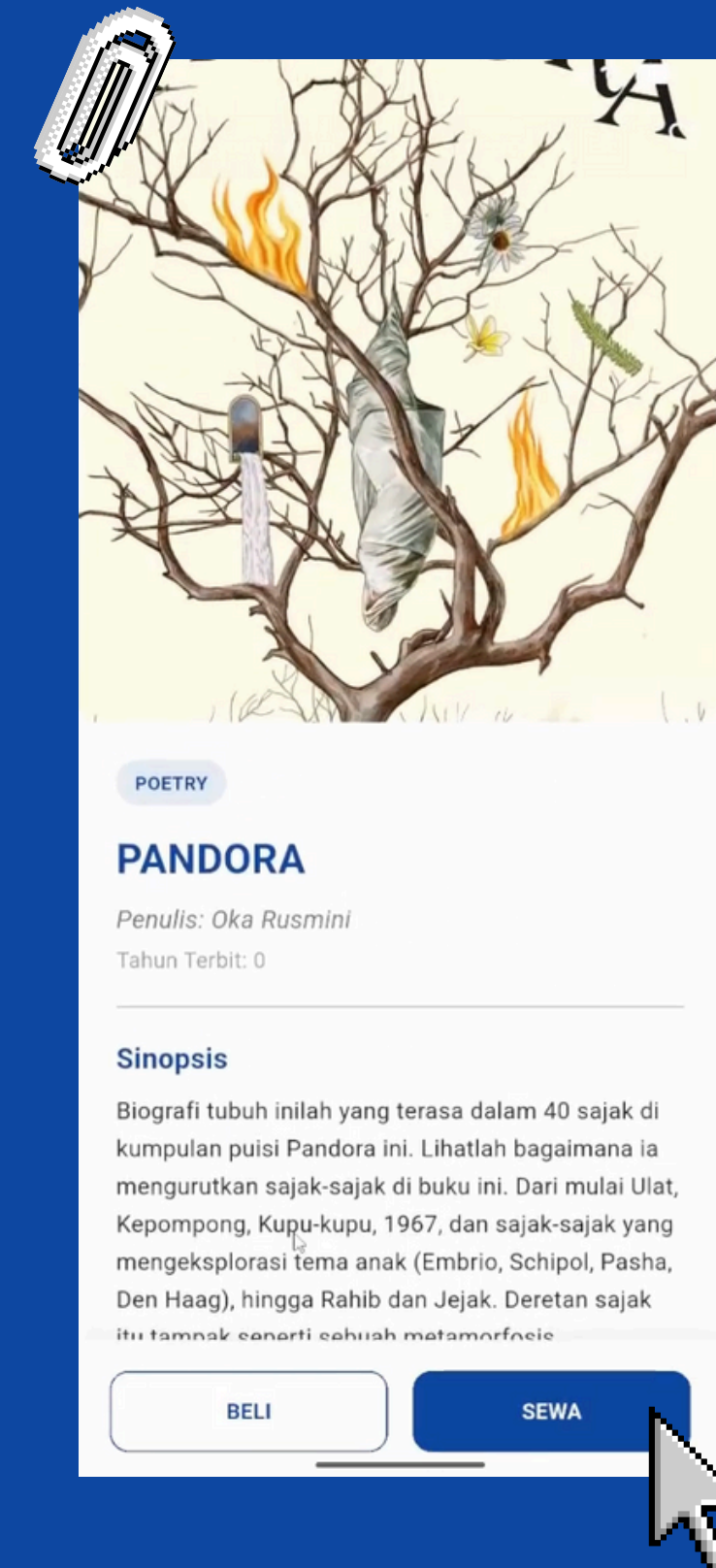
## Target

Pecinta buku (Pembaca) yang membutuhkan akses cepat ke berbagai katalog buku. Akibat dari sulitnya mencari rekomendasi buku secara acak dan keterbatasan informasi ketersediaan buku di satu tempat.



# Fitur Utama

-  User Management: Registrasi dan login aman menggunakan Firebase Auth.
-  Smart Catalog: Paginasi (Lazy Loading) untuk performa aplikasi yang ringan.
-  Discovery Mode: Fitur "Random Book" (Buku Acak) dan Filter per Genre.
-  Personalized Favorites: Simpan buku pilihan ke daftar favorit via Firestore.
-  Dynamic View: Perpindahan mode tampilan List dan Grid yang mulus.



[Back to Agenda Page](#)



# Arsitektur Sistem

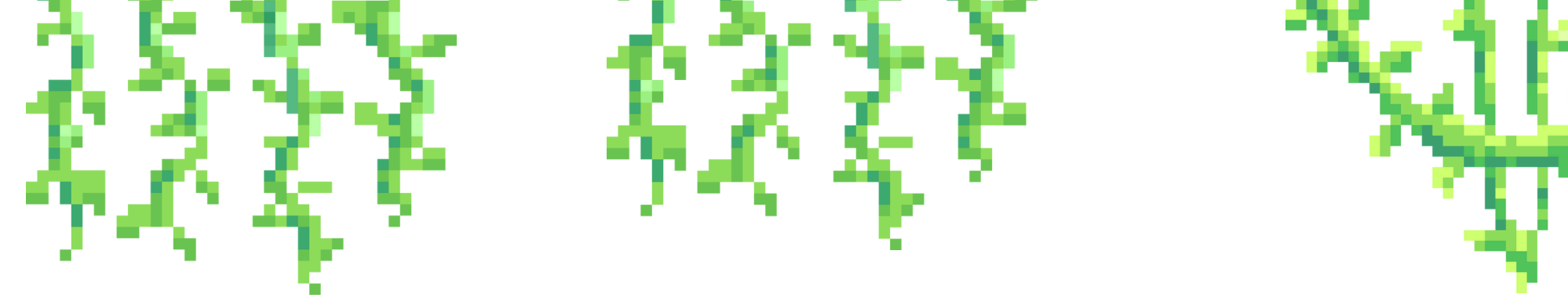



- **Alur:** Login -> Browse Catalog -> Filter/Search -> Detail -> Rent/Favorite.
  - **Arsitektur:** Model-View-Controller (MVC) yang disederhanakan.
  - **State Management:** Riverpod (untuk performa reaktif).
  - **Library Utama:** GetX (Navigation), Dio (API), Firebase Core (Auth/Firestore).
- 
- 



# Proses Pengembangan



- **Plan:** Analisis kebutuhan fitur dan pemetaan data API.
  - **Design:** Implementasi UI yang responsif menggunakan Flutter Widgets.
  - **Code:** Integrasi API, Firebase, dan logika State Management.
  - **Test & Deploy:** Pengujian aplikasi di Tecno Pova 6 Pro 5G dan build APK (split per ABI)
  - **Tools:** VS Code, GitHub, Postman, Google Drive.
- 
- 

# Masalah & Solusi

- Masalah 1: Error 404 saat memuat gambar sampul buku dari API yang sudah mati.
- Solusi: Menggunakan `errorBuilder` pada widget `Image.network` untuk menampilkan ikon fallback yang tetap rapi.
- Masalah 2: Terjadi "Red Screen" (Dead Code) karena pengecekan state data yang tidak tepat.
- Solusi: Implementasi `Type Safety` pada Data Model dan penggunaan `nullable` untuk menangani data kosong.
- Pelajaran: Selalu siapkan fallback UI dan prioritaskan validasi tipe data (`Type Safety`).

# Video Review



Terimakasih...