

MODUL SPESIFIKASI TEKNIS: CINE-BOOKING APP

Semester Ganjil 2024/2025

Platform: Flutter (Mobile) — **Tim:** 5 Mahasiswa — **Sifat:** Strict (Anti-AI)

1 Deskripsi Proyek

Mahasiswa diminta mengembangkan aplikasi pemesanan tiket bioskop *real-time*. Fokus utama proyek ini adalah pengelolaan *State Management* yang kompleks pada pemilihan kursi dan perhitungan harga dinamis.

2 Protokol Integritas

- Setiap nama class Widget, fungsi logika, dan variabel state harus diakhiri dengan inisial pembuat (CamelCase). *Contoh:* `SeatWidget.Rian`, `calculateTax.Siti()`.
- Wajib mengimplementasikan aturan bisnis berikut secara manual (tidak boleh menggunakan library kalkulasi instan):
 - **The "Long Title" Tax:** Jika jumlah karakter pada Judul Film > 10 huruf, harga tiket ditambah Rp 2.500,- per kursi.
 - **Odd/Even Seat Rule:**
 - Kursi nomor **Ganjil** (A1, A3, B1...) = Harga Normal.
 - Kursi nomor **Genap** (A2, A4, B2...) = Diskon 10%.
- Video Defense:** Video penjelasan per baris kode durasi 2 menit wajib disertakan dalam pengumpulan.

3 Arsitektur Spesifikasi Database (Firebase Firestore)

Wajib mengikuti penamaan field berikut secara persis (*Case Sensitive*).

Collection: users

Field	Type	Deskripsi
uid	String	Primary Key (Dari Auth)
email	String	Validasi Regex: <code>@student.univ.ac.id</code>
username	String	Nama tampilan
balance	Int	Saldo (Default: 0)
created_at	Timestamp	Waktu register

Collection: movies

Field	Type	Deskripsi
movie_id	String	Primary Key
title	String	Judul Film (Penentu Logic Trap 1)
poster_url	String	URL Gambar
base_price	Int	Harga dasar sebelum pajak/diskon
rating	Double	Skala 1.0 - 5.0
duration	Int	Durasi dalam menit

Collection: bookings

Field	Type	Deskripsi
booking_id	String	Primary Key (Auto Generate)
user_id	String	FK ke Users
movie_title	String	Disimpan untuk history
seats	Array<String>	Contoh: ["A1", "A2", "B5"]
total_price	Int	Harga Final (Setelah Trap)
booking_date	Timestamp	Waktu transaksi

4 Pembagian Peran (5 Mahasiswa)

4.1 Anggota 1: Backend Engineer & Data Seeder

- **Tugas Utama:** Setup Firebase (Auth + Firestore).
- **Coding:** Membuat Class Model Data (`movie_model.dart`). Wajib membuat fungsi `fromMap` dan `toMap` manual.
- **Data Seeding:** Menginput minimal 10 film via Firebase Console dengan variasi panjang judul (ada yang pendek, ada yang panjang) untuk menguji *Logic Trap*.

4.2 Anggota 2: Frontend Engineer (Home Module)

- **UI Home:** Membuat tampilan Grid Film (`SliverGridDelegate`). Gunakan widget `Hero` pada gambar poster agar ada animasi saat pindah ke detail.
- **UI Detail:** Menampilkan info film. Tombol "Book Ticket" melayang (*Floating Action Button*) di bawah.
- **Constraint:** Layout harus responsif di HP kecil maupun besar.

4.3 Anggota 3: Frontend Engineer (Seat Matrix Module)

- **Tugas Terberat (UI):** Membuat Grid Kursi dinamis (misal 6x8).
- **Logika Visual:**
 - Kursi Kosong = Warna Abu-abu.
 - Kursi Dipilih = Warna Biru (User bisa toggle select/unselect).
 - Kursi Terjual = Warna Merah (Disabled/Tidak bisa diklik).
- **Naming:** Widget kursi wajib bernama `SeatItem_[Inisial]`.

4.4 Anggota 4: Logic Controller (The Brain)

- **State Management:** Menggunakan Provider/Bloc/GetX untuk menyimpan kursi mana saja yang sedang dipilih (`List<String> selectedSeats`).
- **Implementasi Trap:**
 1. Cek `movie.title.length`. Jika > 10 , tambahkan variabel `tax`.
 2. Loop setiap kursi yang dipilih. Parse string "A2" → ambil angka "2". Jika genap, kurangi harga total.
- **Checkout:** Fungsi final untuk kirim data ke Firebase.

4.5 Anggota 5: QA Lead, Auth & Profile

- **Auth:** Login/Register Page dengan validasi visual (border merah jika error).
- **Profile:** Halaman riwayat tiket. Gunakan library `qr_flutter` untuk generate QR Code unik dari `booking_id`.
- **Integration:** Menyatukan kode Anggota 2 (Home) → Anggota 3 (Seat) → Anggota 4 (Logic).

5 Skenario Pengujian (Untuk QA)

Anggota QA wajib memastikan skenario ini berhasil sebelum submit:

1. **Tes Trap Judul:** Pilih film "Avengers" (8 huruf) vs "Spider-Man: No Way Home" (21 huruf). Pastikan harga dasar berbeda.
2. **Tes Trap Kursi:** Pilih kursi A1 (Ganjil) dan A2 (Genap). Pastikan total harga bukan sekadar $2 \times \text{HargaDasar}$, tapi $\text{HargaDasar} + (\text{HargaDasar} - 10\%)$.
3. **Tes Crash:** Matikan internet saat menekan tombol "Bayar". Aplikasi harus menampilkan pesan error, bukan keluar sendiri (*Force Close*).

6 Timeline Sprint (7 Hari)

- **Hari 1-2:** Setup Git, Firebase, Struktur Folder.
- **Hari 3-4:** Pengerjaan UI Home, Seat UI, dan Login secara paralel.
- **Hari 5:** Anggota 4 masuk mengintegrasikan logika perhitungan harga ke UI Kursi.
- **Hari 6:** Integrasi Booking ke Firebase Testing QA.
- **Hari 7:** Pembuatan Laporan PDF Video Defense.

7 Rubrik Penilaian Teknis (Skala 100)

Komponen	Indikator Capaian	Poin
Integritas Kode	Penamaan variabel <code>_inisial</code> konsisten & Logic Trap terimplementasi manual.	30
Kompleksitas UI	Grid Kursi interaktif (bisa select/unselect) & Animasi Hero berjalan halus.	25
Fungsionalitas	Data tersimpan di Firebase & QR Code muncul.	25
Kerjasama Tim	Git History aktif (min. 10 commit/orang) & Laporan lengkap.	20