

POLITEKNIK NEGERI BANYUWANGI

Laporan Final Project: CineBooking App

Pemrograman Perangkat Bergerak
Semester 3 (Tiga)

Kelompok 05:

Jevon	NIM001	<i>Backend Engineer & Data Seeder</i>
Adel	NIM002	<i>UI Engineer - Home Module</i>
Intan	NIM003	<i>UI Engineer - Seat Matrix</i>
Salam	NIM004	<i>Logic Controller</i>
Fariz	NIM005	<i>QA Lead, Auth & Profile</i>

BAB 1

Pendahuluan & Pembagian Kerja

1.1 Deskripsi Aplikasi

Aplikasi CineBooking adalah aplikasi pemesanan tiket bioskop real-time berbasis Flutter dengan Firebase. Aplikasi ini memungkinkan pengguna untuk melihat film yang tersedia, memilih kursi, dan melakukan booking dengan perhitungan harga dinamis.

1.2 Tabel Pembagian Tanggung Jawab

Nama NIM Peran Kode Watermark	— — — — —	Jevon NIM001 Backend Engineer & Data Seeder MovieModelJevon, BookingModelJevon, MovieSeederJevon	Adel NIM002 UI Engineer - Home Module HomePageAdel, MovieCardAdel, MovieDetailPageAdel	Intan NIM003 UI Engineer - Seat Matrix SeatPageIntan, SeatItemIntan	Salam NIM004 Logic Controller SeatControllerSalam	Fariz NIM005 QA Lead, Auth & Profile LoginPageBioskopFariz, RegisterPageBioskopFariz, ProfilePageFariz
-------------------------------------	-----------	--	--	---	---	--

BAB 2

Bukti Keaslian Kode (Strict Mode)

2.1 Watermark Code

Setiap kode memiliki inisial pembuat di nama class, fungsi, dan variabel:

```
// Contoh watermark dari kode program:  
class MovieModelJevon {} // Inisial "Jevon"  
class HomePageAdel {} // Inisial "Adel"  
class SeatPageIntan {} // Inisial "Intan"  
class SeatControllerSalam {} // Inisial "Salam"  
class LoginPageBioskopFariz {} // Inisial "Fariz"
```

2.2 Logic Trap Implementasi Manual

```
// File: controllers/seat_controller.dart  
int calculateTotalPrice() {  
    int total = basePriceSalam * _selectedSeatsSalam.length;  
  
    // Trap 1: Jika judul film > 10 karakter, tambah Rp 2.500 per kursi  
    if (movieTitleSalam.length > 10) {  
        total = total + (2500 * _selectedSeatsSalam.length);  
    }  
  
    // Trap 2: Kursi genap dapat diskon 10%  
    for (var seat in _selectedSeatsSalam) {  
        final number = int.tryParse(seat.substring(1)) ?? 0;  
        if (number % 2 == 0) {  
            total -= (basePriceSalam * 0.10).toInt();  
        }  
    }  
    return total;  
}
```

BAB 3

Arsitektur Backend

Dikerjakan oleh [Jevon] (Backend Engineer)[cite: 23].

3.1 Struktur Database (Firestore)

Menggunakan 3 Collection utama: users, movies, dan bookings[cite: 13, 15, 17].

3.2 Data Seeding

Bukti seeding minimal 10 film dummy ke Firebase menggunakan MovieSeederJevon[cite: 27].

```
{
    "movie_id": "moviel",
    "title": "Avatar",
    "poster_url": "https://image.tmdb.org/t/p/original/6wkfovpn7Eq8dYNKaG5PY3q2oq6.jpg",
    "base_price": 55000,
    "rating": 4.5,
    "duration": 162
}
// File: utils/movie_seeder_jevon.dart
final movies = [
    {
        'movie_id': 'moviel',
        'title': 'Avatar',
        'poster_url': 'https://image.tmdb.org/t/p/original/6wkfovpn7Eq8dYNKaG5PY3q2oq6.jpg',
        'base_price': 55000,
        'rating': 4.5,
        'duration': 162,
    },
];
```

BAB 4

Implementasi Antarmuka & Auth

4.1 Halaman Login & Register (Fariz)

- Validasi email harus menggunakan domain @poliwangi.ac.id
- Implementasi Shared Preferences untuk remember me
- Error handling dengan snack bar

```
// File: models/auth/login_page_fariz.dart
final regexEmail = RegExp(r'^[a-zA-Z0-9._%+-]+@poliwangi\.ac\.id$');
```

4.2 Halaman Home (Adel)

- Menggunakan GridView.builder dengan SliverGridDelegate
- Hero animation pada poster film
- Responsif untuk berbagai ukuran layar

```
// File: models/home/home_page_adel.dart
int gridCount = screenWidth > 600 ? 3 : 2;
```

4.3 Halaman Seat Selection (Intan)

- Grid kursi 6x6 dengan visual state:
 - Abu-abu: Available
 - Biru: Selected
 - Merah: Sold
- Interaksi toggle select/unselect

```
// File: modules/seat/seat_item_intan.dart
Color seatColorIntan;
if (isSoldIntan) {
    seatColorIntan = Color(0xFF2D1B1B); // Merah
} else if (isSelectedIntan) {
    seatColorIntan = Color(0xFF1A2634); // Biru
} else {
    seatColorIntan = Color(0xFF1A1F29); // Abu-abu
}
```

BAB 5

State Management & Logic

5.1 State Management (Salam)

Menggunakan ChangeNotifier untuk mengelola state kursi yang dipilih:

```
// File: controllers/seat_controller.dart
class SeatControllerSalam with ChangeNotifier {
    final List<String> _selectedSeatsSalam = [];
    List<String> get selectedSeats => _selectedSeatsSalam;

    void toggleSeat(String seat) {
        if (_selectedSeatsSalam.contains(seat)) {
            _selectedSeatsSalam.remove(seat);
        } else {
            _selectedSeatsSalam.add(seat);
        }
        notifyListeners();
    }
}
```

5.2 Real-time Sold Seats Tracking

Menggunakan StreamSubscription untuk update kursi yang sudah terjual secara real-time:

```
void _listenSoldSeatsSalam(String movieTitle) {
    _bookingSubscriptionSalam = FirebaseFirestore.instance
        .collection("bookings")
        .where("movie_title", isEqualTo: movieTitle)
        .snapshots()
        .listen((snapshot) {
            // Update sold seats
            notifyListeners();
        });
}
```

5.3 Checkout Process

Validasi koneksi internet sebelum melakukan booking:

```
Future<String?> checkout(String userId) async {
    try {
        final result = await InternetAddress.lookup('google.com');
        if (result.isEmpty) {
            return "Tidak ada koneksi internet";
        }
    } catch (_) {
        return "Tidak ada koneksi internet";
    }
    // ... proses booking
}
```

BAB 6

Integrasi & Git Branching

6.1 Struktur Git Branch

```
main
├── feature/backend-jevon
├── feature/ui-home-adel
├── feature/ui-seat-intan
├── feature/logic-salam
└── feature/auth-profile-fariz
```

6.2 Integrasi Provider di Main App

```
// File: main.dart
runApp(
  MultiProvider(
    providers: [
      ChangeNotifierProvider(create: (_) => TicketProviderFariz()),
      ChangeNotifierProvider(create: (_) => SeatControllerSalam()),
    ],
    child: const MyApp(),
  )
);
```

6.3 Routing Antara Modul

```
// Dari home ke detail film
Navigator.push(context, MaterialPageRoute(
  builder: (_) => MovieDetailPageAdel(movie: movies[i])));

// Dari detail ke seat selection
Navigator.push(context, MaterialPageRoute(
  builder: (context) => SeatPageIntan(
    movieTitleIntan: movie.title,
    basePriceIntan: movie.base_price,
  )));

```

BAB 7

Pengujian & Penutup

7.1 Skenario Pengujian yang Dilakukan

1. [Tes Trap Judul Film]:
 - Film “Avatar” (6 karakter): harga normal
 - Film “Spider-Man: No Way Home” (21 karakter): +Rp 2.500 per kursi
2. [Tes Trap Kursi Genap]:
 - Kursi A1 (ganjil): harga normal
 - Kursi A2 (genap): diskon 10%
3. [Tes Error Handling]:
 - Matikan internet saat checkout: tampil pesan error
 - Login dengan email non-poliwangi: validasi error

7.2 QR Code untuk Ticket History

```
// File: models/profile/ticket_history_page_fariz.dart
QrImageView(
  data: ticket.bookingId,
  version: QrVersions.auto,
  size: 120,
  backgroundColor: Colors.white,
)
```

7.3 Link Video Demo

Berikut adalah link video demo aplikasi yang diunggah ke Google Drive/YouTube:

- [Link:] [MASUKKAN_LINK_VIDEO_DEMO]
- Durasi: 2 menit penjelasan kode sesuai requirement

7.4 Kesimpulan

Aplikasi CineBooking telah berhasil mengimplementasikan:

1. Arsitektur Firebase yang sesuai spesifikasi
2. UI responsive dengan animasi Hero
3. Logic trap manual tanpa library kalkulasi instan
4. State management dengan Provider
5. Validasi dan error handling yang komprehensif