POLITEKNIK NEGERI BANYUWANGI JURUSAN TEKNOLOGI
INFORMASI PROGRAM STUDI TEKNIK REKAYASA PERANGKAT LUNAK

Laporan Final Project: CineBooking App

Pemrograman Perangkat Bergerak
Semester 3 (Tiga)

KELOMPOK 05

Moh. Jevon Attaillah	362458302035	*Backend Engineer & Data Seeder*
Adelia Fioren Zety	362458302124	*UI Engineer - Home Module*
Intan Rahma Safira	362458302099	*UI Engineer - Seat Matrix*
Salam Rizqi Mulia	362458302041	*Logic Controller*
Achmad Alfarizy Satriya G	362458302147	*QA Lead, Auth & Profile*

Dosen Pengampu: Tri Hadiah Mulia Bunda, S.Kom., M.T. Desember 2024

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 Pembagian Tanggung Jawab Kelompok 5

No	Nama	NIM	Peran	Kode Watermark
1	Moh. Jevon Attaillah	362458302035	Backend Engineer & Data Seeder	`MovieModelJevon`, `BookingModelJevon`, `MovieSeederJevon`
2	Adelia Fioren Zety	362458302124	UI Engineer - Home Module	`HomePageAdel`, `MovieCardAdel`, `MovieDetailPageAdel`
3	Intan Rahma Safira	362458302099	UI Engineer - Seat Matrix	`SeatPageIntan`, `SeatItemIntan`
4	Salam Rizqi Mulia	362458302041	Logic Controller	`SeatControllerSalam`
5	Achmad Alfarizy Satriya G	362458302147	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": "movie1",
  "title": "Avatar",
  "poster_url": "https://image.tmdb.org/t/p/original/6wkfovpn7Eq8dYNKaG5PY3q2
oq6.jpg",
  "base_price": 55000,
  "rating": 4.5,
  "duration": 162
}
```

```
// File: utils/movie_seeder_jevon.dart
final movies = [
  {
    'movie_id': 'movie1',
    'title': 'Avatar',
    'poster_url': 'https://image.tmdb.org/t/p/original/6wkfovpn7Eq8dYNKaG5PY3q
2oq6.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 `SharedPreferences` untuk remember me
- Error handling dengan snackbar

```
// 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
UI

BAB 6
State Management & Logic

Dikerjakan oleh Transaction Logic Specialist[cite: 37].

6.1 State Booking Seat

State management menggunakan (Provider/Bloc/GetX) untuk fitur **Book Now**[cite: 38, 39]. Pembuatan logika pemesanan kursi dilakukan dengan menggunakan Provider, pada baris pertama terdapat class dengan nama SeatControllerSalam sebagai controller state yang mengatur logika pemesanan kursi pada aplikasi. Class tersebut menggunakan ChangeNotifier sebagai pembaruan UI ketika ada perubahan data.

6.2 Inisialisasi dan Deklarasi Variabel

```
final List<String> _selectedSeatsSalam = [];
List<String> get selectedSeats => _selectedSeatsSalam;
List<String> _soldSeatsSalam = [];
List<String> get soldSeats => _soldSeatsSalam;

int basePriceSalam = 0;
String movieTitleSalam = "";
StreamSubscription? _bookingSubscriptionSalam;
```

`_selectedSeatsSalam` variabel yang berfungsi untuk menyimpan kursi yang dipilih oleh pengguna melalui state lokal. `_soldSeatsSalam` variabel yang berfungsi untuk meyimpan kursi yang sudah dibeli, data diambil dari real-time Firestore. `basePrice` harga dasar untuk setiap kursi. `movieTitle` untuk menyimpan judul film yang dipilih. `_bookingSubscription` untuk menyimpan hasil monitoring perubahan data dari Firestore secara realtime.

6.3 InitData() - Mengatur data awal

```
void initData({
  required int basePriceInput,
  required String movieTitleInput,
}) {
  basePriceSalam = basePriceInput;
  movieTitleSalam = movieTitleInput;
  _listenSoldSeatsSalam(movieTitleSalam);
}
```

6.4 Logika Toggle dan Checkout

6.4.1 Toggle Seat

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

`toggleSeat(String seat)` memungkinkan pengguna memilih atau membatalkan pilihan kursi, kursi ditambahkan ke `_selectedSeatsSalam` jika belum dipilih, atau dihapus jika sudah ada, setiap perubahan memanggil `notifyListeners()` agar UI update.

6.4.2 Perhitungan Total Harga

```
int calculateTotalPrice() {
  int total = basePriceSalam * _selectedSeatsSalam.length;
  if (movieTitleSalam.length > 10) {
    total = total + (2500 * _selectedSeatsSalam.length);
  }
  for (var seat in _selectedSeatsSalam) {
    final number = int.tryParse(seat.substring(1)) ?? 0;
    if (number % 2 == 0) {
      total -= (basePriceSalam * 0.10).toInt();
    }
  }
  return total;
}
```

Logika diatas menghitung total berdasarkan jumlah kursi x harga dasar yang didapat dari `basePriceSalam`, tambahan biaya 2500 per kursi jika judul film lebih dari 10 karakter, Diskon 10% untuk kursi genap diambil dari nomor kursi. Total harga akan otomatis digunakan saat checkout.

6.4.3 Checkout

```
  Future<String?> checkout(String userId) async {
    try {
      final result = await InternetAddress.lookup('google.com');
      if (result.isEmpty || result[0].rawAddress.isEmpty) {
        return "Tidak ada koneksi internet. Silahkan cek jaringan Anda";
      }
    } catch (_) {
      return "Tidak ada koneksi internet. Silahkan cek jaringan Anda";
    }

    if (_selectedSeatsSalam.isEmpty || basePriceSalam == 0 ||
movieTitleSalam.isEmpty) {
      return "Data booking tidak lengkap";
    }

    final bookingCollection =
FirebaseFirestore.instance.collection('bookings');
    final docRef = bookingCollection.doc();
    final bookingId = docRef.id;
    final data = {
      'booking_id': bookingId,
      'user_id': userId,
      'movie_title': movieTitleSalam,
      'seats': _selectedSeatsSalam,
      'total_price': calculateTotalPrice(),
      'booking_date': Timestamp.now(),
```

```
    };

    await docRef.set(data);
    await
FirebaseFirestore.instance.collection("bookings").where("movie_title",
isEqualTo: movieTitleSalam).snapshots().first;
    _selectedSeatsSalam.clear();
    notifyListeners();
    return null;
  }
```

Melakukan checkout dengan memeriksa koneksi internet terlebih dahulu. Jika tidak ada, akan mengembalikan pesan error dan validasi data booking. Untuk validasi data booking, jika kursi belum dipilih, harga belum diatur, atau judul film kosong, maka akan mengembalikan pesan error. Penyimpanan data booking ke Firestore di koleksi `bookings`, field yang disimpan `booking_id`, `user_id`, `movie_title`, `seats`, `total_price`, `booking_date`. Setelah booking berhasil, `_selectedSeatsSalam` akan dikosongkan untuk memulai transaksi baru.