Proyek Semester

1. Buatlah kelompok berisi 3 mahasiswa
2. Selesaikan Studi Kasus berikut:

Sebuah startup lokal ingin membangun **sistem pemesanan tiket** (misalnya konser, bioskop, atau travel), dengan fitur:

- Pengguna bisa memilih **jadwal** dan **kursi**.
- Disaat ramai (high traffic), terdapat potensi terjadi masalah seperti berikut:
    - Dua orang atau lebih mencoba memesan **kursi yang sama** di waktu yang hampir bersamaan.
    - Terjadi **double booking** atau data tidak konsisten.
- Startup ini meminta kalian merancang **backend secara konseptual** menggunakan:
    - **Golang (Fiber)** sebagai web framework,
    - **Clean Architecture** sebagai pola arsitektur,
    - **MongoDB** sebagai basis data,
    - **Swagger-style API contract**,
    - **Unit testing di level domain/use case** (rencana, bukan implementasi).

Tugas anda adalah **merancang dan menjelaskan SOLUSI BACKEND secara teoretis** tanpa menulis kode, tapi dengan analisis yang baik dan terdokumentasi.

3. Untuk mempersempit ruang lingkup, hal berikut yang perlu anda bahas dalam rancangan anda:
    a. Domain
        - Event (tentukan: tiket konser, tiket bioskop, tiket pesawat, dst)
        - Seat (Kursi)
        - Booking (Pemesanan)
        - User
    b. Fitur minimal yang dibahas:
        - Melihat daftar kursi yang tersedia dan ter-booked per event / per jadwal
        - Melakukan pemesanan kursi (bisa lebih dari satu kursi)
        - Membatalkan pemensanan
        - Menahan kursi untuk waktu tertentu (ketika proses penyelesaian booking)

4. Isi dari rancangan anda adalah seperti dilampiran
5. Siapkan PTT maksimal presentasi 5 menit, isi:
    a. Penjelasan event
    b. Disain mongoDB
    c. Race Condition
    d. Solusi

1. Analisa Domain dan Use Case
    a. Deskripsi tentang domain masalah
        i. Penjelasan apa itu event, kursi, booking dan user
        ii. Apa masalah utama (double booking / race condition)?
    b. Daftar use case (minimal 4), contoh:
        i. UC1: lihatSeatMap – melihat status kursi pada event / jadwal tertentu
        ii. UC2: CreateBooking – memesan kursi
        iii. Dst.
    c. Pada setiap use case jelaskan:
        i. Pre-condition
        ii. Post-condition
        iii. Alur
        iv. Alur alternatif (contoh: kasus ketika gagal karena kursi sudah di-book)
2. Desain Clean Architecture:
    a. Struktur file proyek anda
    b. Alur kebergantungan, contoh: HTTP handler → UseCase → Repository → MongoDB
3. Disain Skema MongoDB
    a. Struktur koleksi
    b. Alasan disain yang dibuat
4. API Contract (ala Swagger)
    a. Method: GET/POST/DELETE
    b. Path: /events/{eventId}/seats
    c. Request body
    d. Response
    e. Referensi: https://editor.swagger.io/
5. Analisa Race condition dan double Booking
    a. Jelaskan skenario race condition yang mungkin terjadi
    b. Beri 2 pendekatan solusi, selain penjelasan berikan juga kelebihan dan kekurangannya
    c. Dari 2 pendekatan solusi itu, pilihlah 1 yang menurut anda cocok untuk anda gunakan pada studi kasus anda, beri penjelasannya!
6. Rencana Unit Testing
    a. Contoh:

No	Test Case	Kondisi Awal	Input	Expected Output/behaviour
1	Berhasil booking kursi	Status kursi A1 available	User_id X, jadwal Y, seat A1	Booking berhasil, status kursi A1 menjadi booked
2	Gagal booking kursi	Status kursi A1 booked	User_id X, jadwal Y, seat A1	Error “Kursi sudah di booked”

7. Referensi