

# **Back End Java Challenge**

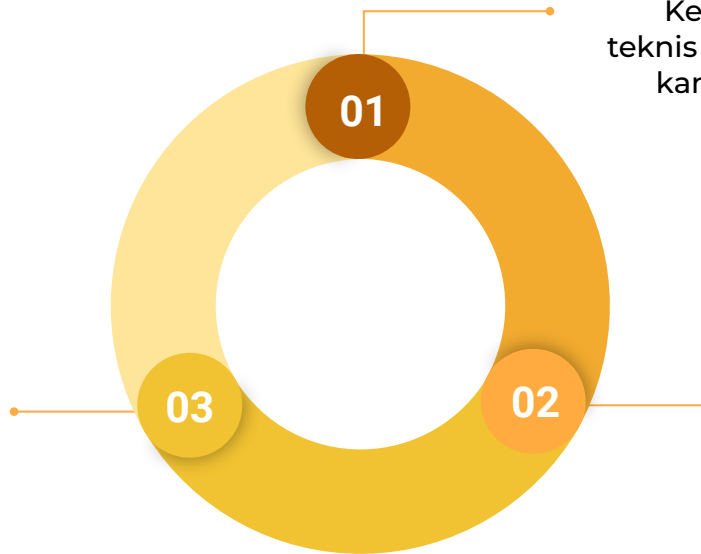
## Chapter 4

Selamat datang di **Challenge Chapter 4**  
online course **Back End Java** dari  
Binar Academy!



### ACCEPTANCE CRITERIA

Kriteria pengumpulan challenge yang harus kamu penuhi untuk dapat melewati chapter



### SKILL METRICS

Kemampuan teknis yang akan kamu pelajari

### DELIVERY

Hal yang akan kamu lakukan untuk dapat melewati chapter

### SKILL METRICS

(Kemampuan teknis yang akan kamu pelajari)

1. Spring Framework
2. Spring Boot
3. Relational Databases, ORM and Database Operation
4. Store Procedure
5. NoSQL
6. Spring Data JPA

### DELIVERY

(Hal yang akan kamu lakukan untuk dapat melewati chapter)

1. Membuat aplikasi
2. Menerapkan Spring Boot
3. Membuat ERD menggunakan Store Procedure
4. Membuat CRUD data menggunakan Store Procedure
5. Kemampuan mengenai database
6. Push hasil program pada GitLab

### CRITERIA

(Kriteria pengumpulan challenge yang harus kamu penuhi untuk dapat melewati chapter)

Ada beberapa hal yang perlu kamu selesaikan untuk menyelesaikan challenge, yaitu:

1. Bagaimana membuat ERD: 20 pts
2. Melakukan CRUD pada database: 10 pts
3. Operasi dasar SQL + Store Procedure: 20 pts
4. Membuat project menggunakan Spring Boot: 50 pts

Pada challenge chapter 4 ini, kamu diminta untuk **membuat ERD + CRUD data menggunakan Store Procedure pada Database.**

Untuk teknis pengerjaannya, kamu bisa cek pada halaman berikutnya, ya!



### Project challenge ini akan digunakan untuk Chapter 4-8.

Mas Gun adalah seorang pengusaha yang baru merintis usaha bioskop. Karena pelayanannya mendapatkan review bintang lima, banyak yang berkunjung ke bioskopnya.

Suatu hari, film yang berjudul Pengabdian Mantan sedang tayang. Film ini menarik lebih banyak orang untuk datang ke bioskop dan membuat Mas Gun kewalahan.





Saking banyaknya kawula muda yang menonton film ini, sering kali mereka kehabisan tiket dan kecewa karena tidak bisa menikmati filmnya.

Fyi, pembelian tiket hanya bisa dilakukan secara on the spot atau beli di tempat.



Melihat permasalahan ini, mas Gun mendatangi kamu sebagai developer untuk meminta solusi.

Mas Gun berkata **“aku ingin bioskopku punya aplikasi reservasi tiket online supaya orang-orang bisa pesan tiket disana dan nggak perlu repot-repot harus antre”**.

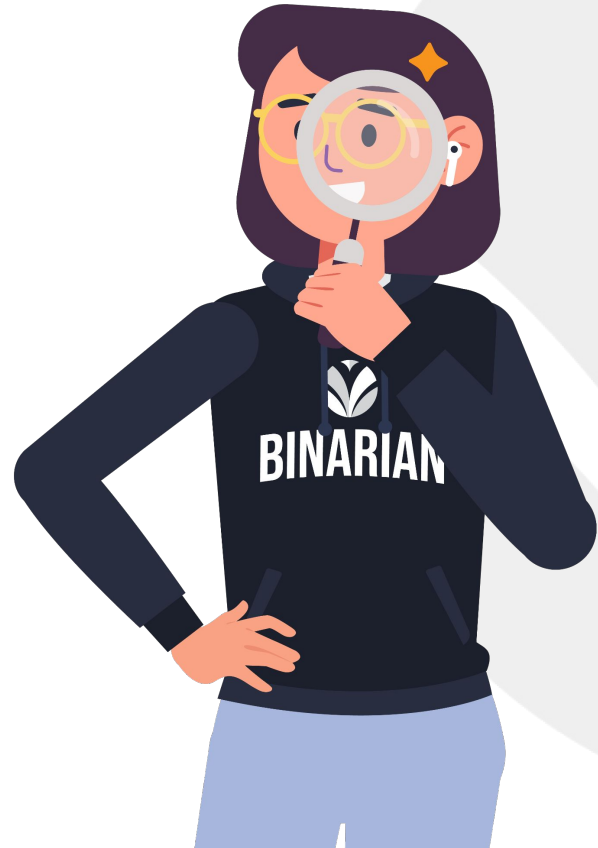
Fitur yang perlu ada di aplikasinya yaitu:

1. Menampilkan film yang sedang tayang
2. Menampilkan jadwal film tertentu, studionya dan harga tiketnya
3. Menampilkan kursi yang masih tersedia
4. Melakukan reservasi tiket



**Untuk challenge Chapter 4, beberapa table yang harus ada dan field harus memenuhi kolom berikut:**

1. Users: username, email address, password
2. Films: film code, film name, sedang tayang atau tidak
3. Schedules: schedule id, film code, tanggal tayang, jam mulai, jam selesai, harga tiket
4. Films dan schedules memiliki relasi one to many
5. Seats: studio name (A, B, C), nomor kursi (menggunakan composite key)



Dari lima ketentuan tadi, buatlah ERD dari table tersebut.

Kamu boleh meng-improve kolom table dan menambahkan table-nya.

**Buat database schema pada lokal kamu serta buat store procedure untuk function CRUDnya.**



Untuk memulai projectnya, buat inisiasi project dengan menggunakan Spring initializr. Gunakan packaging jar dan Spring Boot default bawaan dari Spring initializr

Berikut dependency yang diperlukan:

- **Lombok**
- **Spring Data JPA**
- **PostgreSQL driver**

Dari table yang harus ada, semua table tersebut harus dibuatkan entity modelnya dan DAO (repository).



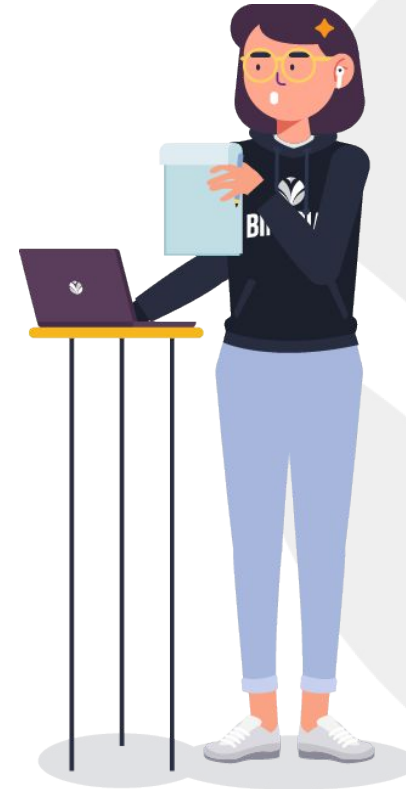
**Buat Service class untuk membuat business logic dari fitur ini:**

1. Menambahkan film baru
2. Mengupdate nama file
3. Menghapus film
4. Menampilkan film yang sedang tayang
5. Menampilkan jadwal dari film tertentu
6. Menambahkan user
7. Mengupdate user
8. Menghapus user



Masih ngobrolin tentang service, ada beberapa catatan yang perlu diperhatikan:

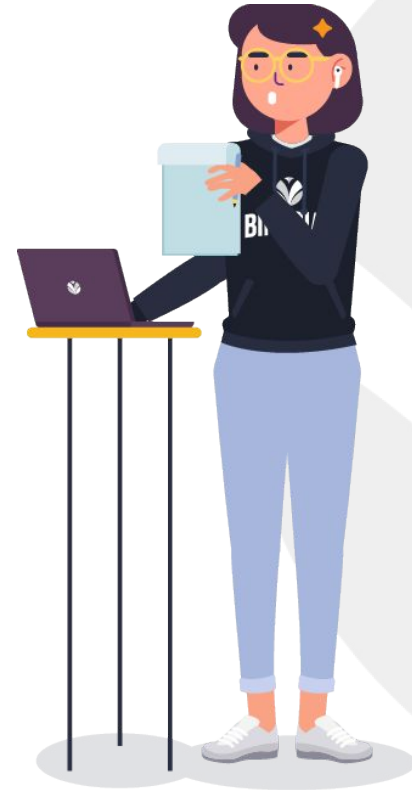
- Service yang berhubungan dengan film dan jadwal dinamakan **FilmService**
- Service yang berhubungan dengan user dinamakan **UserService**



Buat masing-masing service dengan interface sesuai nama service pada halaman 12, kemudian buat class implementasinya dengan nama **FilmServiceImpl** dan **UserServiceImpl**.

Konfigurasi spring.jpa.hibernate.ddl-auto=create

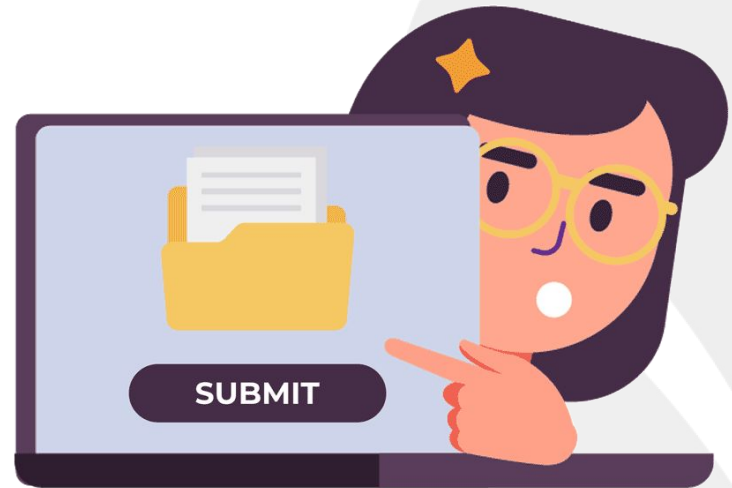
**Note: Pastikan pada saat menyalakan aplikasi tidak ada yang error ya!**





### Ketentuan Pengumpulan Tugas:

- Push hasil program pada branch “Challenge\_4” di GitLab.
- Upload tugas ke GDrive dalam bentuk .jar file.
- Kirimkan kedua link (link GitLab dan GDrive) ke fasilitator untuk diperiksa.



**Terima**  
**Kasih**