



CHALLENGE

DOKUMENTASI TECHNICAL

CHAPTER 5

RESERVASI
TIKET BIOSKOP

TIM A CH 5



Daftar Isi

DAFTAR ISI	2
Catatan Release Dokumen	3
Version History	4
I	5
FLOW CHART AND ENTITY RELATIONSHIP DIAGRAM	5
II	7
TABLE AND INSERT DATA	11
11I	
DOKUMENTASI RUNNING API	11



Catatan Release Dokumen

Tanggal Release	Versi	Oleh	Catatan Perubahan
20 Juni 2022	1.0.0	Tim C CH 4	Versi Pertama
20 Juni 2022	1.0.1	Tim C CH 4	Revisi Pertama
21 Juni 2022	1.0.2	Tim C CH 4	Versi Kedua
28 Juni 2022	1.0.3	TIM C CH 4	Versi Ketiga
01 Juli 2022	1.0.4	Tim A CH 5	Versi Keempat

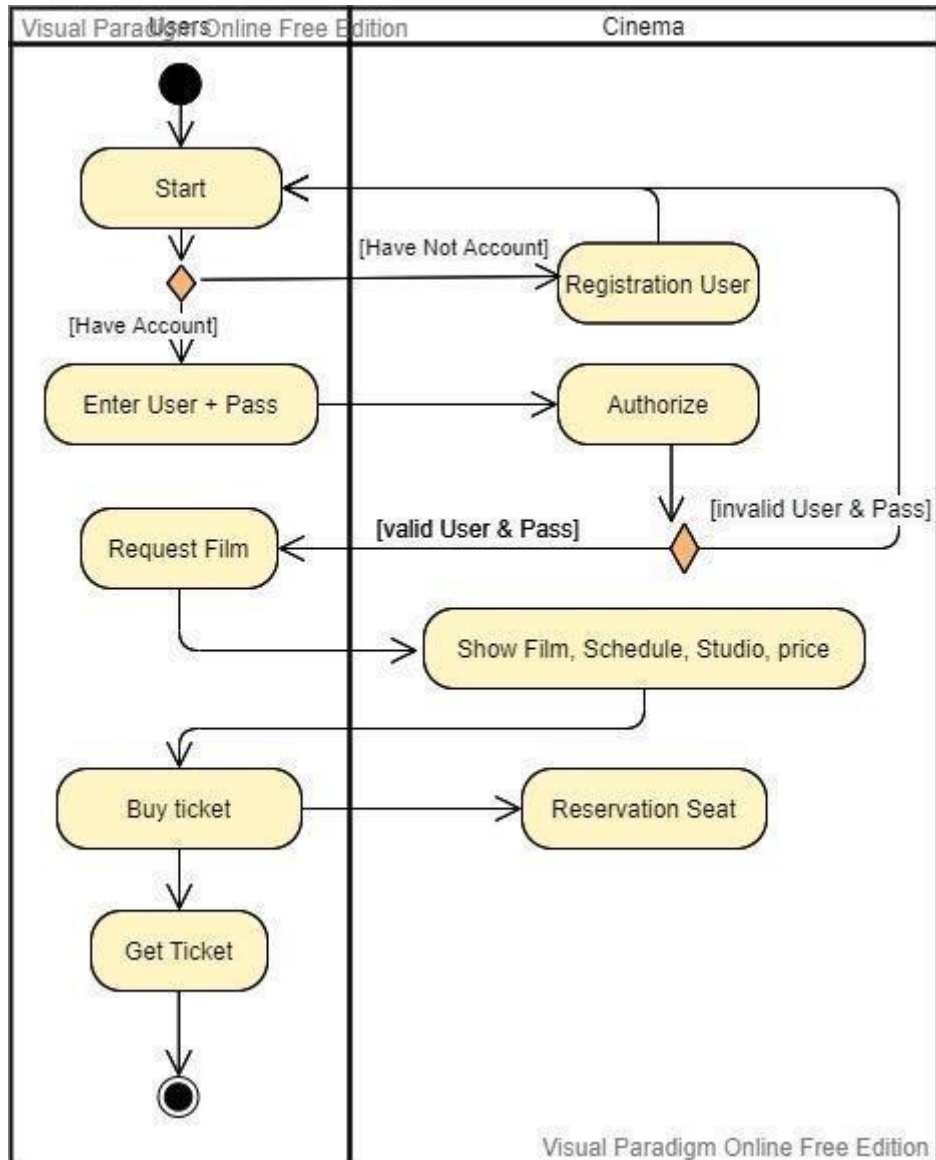


Version History

DATE	VERSION	AUTHOR	DESCRIPTION
20/06/2022	1.0.0	TIM C CH 4	BERISI PROJECT FLOW, ERD, DAN QUERY PEMBUATAN TABEL (Implementasi query DDL) DAN MELAKUKAN SELECT, INSERT, DAN JOIN (Implementasi Query DML) PADA DATABASE BIOSKOP DB
20/06/2022	1.0.1	TIM C CH 4	REVISI DOC.
21/06/2022	1.0.2	TIM C CH 4	
28/06/2022	1.0.3	TIM C CH 4	MENAMBAHKAN RESPON HANDLER DI SETIAP ENDPOINT
01/06/2022	1.0.4	TIM A CH 5	MIGRATE FROM MYSQL TO POSGRESQL

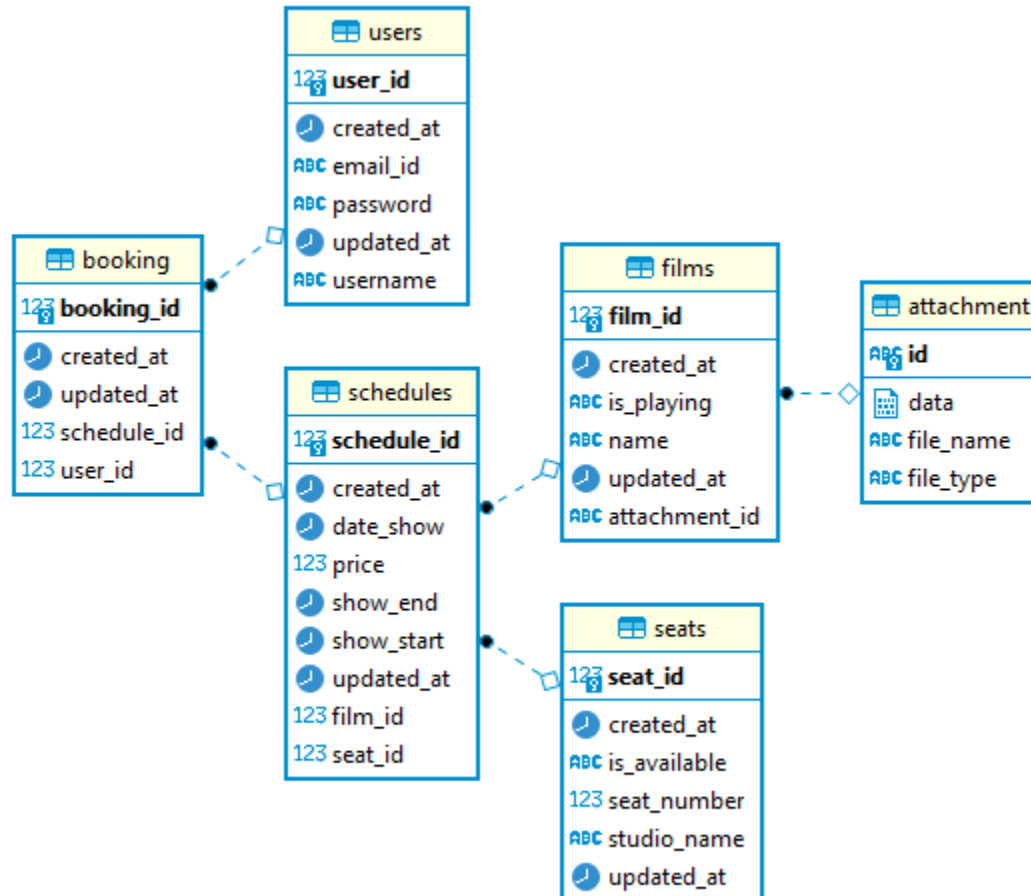
Flow Chart And Entity Relationship Diagram

A. Project Flow





B. Entity Relationship Diagram





II

Table And Insert Data

1. Tables

- films

Column Name	Data Type	Length	Not Null	Auto Increment	Key
<i>film_id</i>	Bigserial	-	v		Primary Key
<i>Is_playing</i>	Int4	-	-		
<i>Name</i>	varchar	255	-		
<i>Created_at</i>	Timestamp	-	-		
<i>updated_at</i>	timestamp	-	-		
<i>Attachment_id</i>	Varchar	255	-		Foreign Key

Query :

```
CREATE TABLE films (  
    film_id bigserial NOT NULL,  
    created_at timestamp NOT NULL,  
    is_playing varchar(255) NULL,  
    name varchar(255) NULL,  
    updated_at timestamp NULL,  
    attachment_id varchar(255) NULL,  
    CONSTRAINT films_pkey PRIMARY KEY (film_id)  
);
```

- attachment

Column Name	Data Type	Length	Not Null	Auto Increment	Key
<i>id</i>	Varchar	255	v		Primary Key
<i>Data</i>	bytea	-	-		
<i>File_name</i>	varchar	255	-		
<i>File_type</i>	varchar	255	-		

Query :

```
CREATE TABLE attachment (  
    id varchar(255) NOT NULL,  
    "data" bytea NULL,  
    file_name varchar(255) NULL,  
    file_type varchar(255) NULL,  
    CONSTRAINT attachment_pkey PRIMARY KEY (id)  
);
```

- seats

Column Name	Data Type	Length	Not Null	Auto Increment	Key
<i>Seat_id</i>	Big serial	-	V		Primary Key



<i>Is_available</i>	Int4	-	-		-
<i>Seat_number</i>	Int8	-	-		-
<i>Studio_name</i>	varchar	255	-		
<i>created_at</i>	Timestamp	-	-		-
<i>updated_at</i>	datetime	-	-		-

Query :

```
CREATE TABLE public.seats (  
  seat_id bigserial NOT NULL,  
  is_available int4 NULL,  
  seat_number int8 NULL,  
  studio_name varchar(255) NULL,  
  created_at timestamp(0) NULL,  
  updated_at timestamp(0) NULL,  
  CONSTRAINT seats_pkey PRIMARY KEY (seat_id)  
);
```

- users

Column Name	Data Type	Length	Not Null	Auto Increment	Key
<i>user_id</i>	Big serial	-	v		Primary Key
<i>Email_id</i>	varchar	255	-		-
<i>Password</i>	varchar	255	-		-
<i>username</i>	time	-	-		-
<i>created_at</i>	timestamp	-	-		-
<i>updated_at</i>	timestamp	-	-		-

Query :

```
CREATE TABLE public.users (  
  user_id bigserial NOT NULL,  
  email_id varchar(255) NULL,  
  "password" varchar(255) NULL,  
  username varchar(255) NULL,  
  created_at timestamp(0) NULL,  
  updated_at timestamp(0) NULL,  
  CONSTRAINT users_pkey PRIMARY KEY (user_id)  
);
```

- schedules

Column Name	Data Type	Length	Not Null	Auto Increment	Key
<i>schedule_id</i>	Serial	-	v	-	Primary Key
<i>Date_show</i>	date	-	-	-	-
<i>price</i>	int4	-	-	-	-
<i>Show_start</i>	time	-	-	-	-



<i>Show_end</i>	time	-	-	-	-
<i>Film_id</i>	int8		-	-	foreign key
<i>Seat_id</i>	int8		-	-	foreign key
<i>created_at</i>	timestamp		-	-	-
<i>updated_at</i>	timestamp	-	-	-	-

Query :

```
CREATE TABLE public.schedules (
  schedule_id serial NOT NULL,
  date_show date NULL,
  price int4 NULL,
  show_end time NULL,
  show_start time NULL,
  film_id int8 NULL,
  seat_id int8 NULL,
  created_at timestamp(0) NULL,
  updated_at timestamp(0) NULL,
  CONSTRAINT schedules_pkey PRIMARY KEY (schedule_id),
  CONSTRAINT fkd4yk1lqx5o4urgj84k055u89u FOREIGN KEY (film_id) REFERENCES
  films(film_id),
  CONSTRAINT fkq2lk4wwng0p8s5avud2f8sfg3 FOREIGN KEY (seat_id) REFERENCES
  seats(seat_id)
);
```

- booking

Column Name	Data Type	Length	Not Null	Auto Increment	Key
<i>booking_id</i>	bigserial	-	v	-	Primary Key
<i>schedule_id</i>	int4	-	-	-	Foreign Key
<i>user_id</i>	int8	-	-	-	Foreign Key
<i>created_at</i>	timestamp	-	-	-	-
<i>updated_at</i>	timestamp	-	-	-	-

Query :

```
CREATE TABLE public.booking (
  booking_id bigserial NOT NULL,
  schedule_id int4 NULL,
  user_id int8 NULL,
  created_at timestamp(0) NULL,
  updated_at timestamp(0) NULL,
  CONSTRAINT booking_pkey PRIMARY KEY (booking_id),
  CONSTRAINT fk7udbel7q86k041591kj6lfmfvw FOREIGN KEY (user_id) REFERENCES
  users(user_id),
  CONSTRAINT fkg1rycgklvhsghf10kyoike3b8 FOREIGN KEY (schedule_id) REFERENCES
  schedules(schedule_id)
);
```



2. INSERT DATA

- Insert data films table

INSERT INTO films

(is_playing, "name", created_at, updated_at, attachment_id)

('value', 'value', 'value', 'value');

Example:

INSERT INTO films

(is_playing, "name", created_at, updated_at, attachment_id)

(1, 'Doraemon', '2022-06-01 15:00:00.000', '2022-06-01 15:00:00.000', defd994f-13fe-49ba-8d9b-7b8ec41b0805);

- Insert data attachment table

INSERT INTO attachment

(id, "data", file_name, file_type)

('value', 'value', 'value', 'value');

Example:

INSERT INTO attachment

(id, "data", file_name, file_type)

(defd994f-13fe-49ba-8d9b-7b8ec41b0805, decode('binary-of-image','hex'), 'doraemon', 'image/png);

- Insert data seats table

INSERT INTO seats (seat_id, is_available, seat_number, studio_name, created_at, updated_at)

VALUES ('value', 'value');

Example:

INSERT INTO seats (seat_id, is_available, seat_number, studio_name, created_at, updated_at)

VALUES

(1,1,1,'Studio A','2022-06-13 23:58:01','2022-01-07 05:08:16');

- Insert data users table

INSERT INTO users

(email_id, "password", username, created_at, updated_at)

VALUES ('value', 'value', 'value', 'value', 'value');

Example:

INSERT INTO users



(email_id, "password", username, created_at, updated_at)

VALUES('email@email.com', 'password', 'Rudi', '2022-06-01 00:00:00.000', '2022-06-01 00:00:00.000');

- Insert data schedules table

INSERT INTO public.schedules

(schedule_id, date_show, price, show_end, show_start, film_id, seat_id, created_at, updated_at)

VALUES ('value','value','value');

Example:

INSERT INTO public.schedules

(schedule_id, date_show, price, show_end, show_start, film_id, seat_id, created_at, updated_at)

VALUES(1, '2022-06-01', 30000, '16:00:00', '14:00:00', 1, 1, '2022-06-01 00:00:00.000', '2022-06-01 00:00:00.000');

- Insert data booking table

INSERT INTO booking

(created_at, schedule_id, user_id, updated_at)

VALUES ('value','value');

Example:

INSERT INTO booking

(created_at, schedule_id, user_id, updated_at)

VALUES ('2022-06-01', 1, 1, '2022-06-01'),

('2022-06-02', 2, 2, '2022-06-02')



III

RUNNING API

Ringkasan :

Semua permintaan BIOSKOP API dibuat ke titik akhir mulai: <http://localhost:8083/teamC/v1/> dan <http://localhost:8083/teamC/v1/DTO>

1. Resources

1.1. User

1.1.1. GET By ID

- End Point :
<http://localhost:8083/swagger-ui/index.html#/user-controller/getUserById>
- Request URL
localhost:8083/users/1
- Sample Request

```
{  
  "userId": 1  
}
```

- Sample Response Request

Success Response

```
{  
  "data": [  
    {  
      "ID": 1,  
      "Username": "Rudi",  
      "Email": "email@email.com",  
      "Password": "password"  
    }  
  ],  
  "message": "Successfully Get User By ID!",  
  "status": 200  
}
```

Failed Response



```
{
  "data": "Data Not Found!"
  "message": "User not exist with id :66666",
  "status": 404
}
```

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
username	String	M		User name yang dimiliki oleh konsumen.	Lily
emailId	String	M		Email yang digunakan oleh konsumen saat mendaftar.	Lily@gmail.com
password	String	M		Password yang diinput oleh konsumen saat mendaftar.	123456aaaa1
userId	Long	M		User Id yang diberikan system untuk konsumen.	9

1.1.2. PUT

- End Point :
<http://localhost:8083/swagger-ui/index.html#/user-controller/updateUser>
- Request URL
localhost:8083/users/1
- Sample Request

```
{
  "username": "imam",
  "emailId": "imam@email.com",
  "password": "adadeh",
  "createdAt": "2022-07-01T14:20:16.313Z",
  "updatedAt": "2022-07-01T14:20:16.313Z"
}
```

- Sample Response Request

Success Response
<pre>{ "data": { "userId": 1,</pre>



```
"username": "imam",
"emailId": "imam@email.com",
"password": "adadeh",
"createdAt": "2022-06-01T00:00:00+07:00",
"updatedAt": "2022-07-01T21:21:29.8510876+07:00"
},
"message": "Successfully Updated User!",
"status": 200
}
```

Failed Response

```
{
  "data": "Data Not Found!",
  "message": "User not exist with id :66666",
  "status": 404
}
```

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
username	String	M		User name baru konsumen yang ingin diupdate.	Lily
emailId	String	M		Email baru konsumen yang ingin diupdate.	Lily@gmail.com
password	String	M		Password baru konsumen yang ingin diupdate.	123456aaaa1
userId	Long	M		User Id konsumen yang ingin diupdate..	9

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
username	String	M		User name baru yang telah diupdate.	Lily
emailId	String	M		Email baru baru yang telah diupdate.	Lily@gmail.com
password	String	M		Password baru baru yang telah diupdate.	123456aaaa1
userId	Long	M		User Id konsumen yang ingin diupdate..	9

1.1.3. DELETE

- End Point :

<http://localhost:8083/swagger-ui/index.html#/user-controller/deleteUser>



- Request URL
- Sample Request

```
{  
  "userId": 8  
}
```

- Sample Response Request

Success Response
<pre>{ "data": { "deleted": true }, "message": "Successfully Delete User! ", "status": 200 }</pre>

Failed Response
<pre>{ "data": "Data Not Found!", "message": "User not exist with id :8999999", "status": 404 }</pre>

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
userid	Long	M		User Id yang ingin di delete.	5

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
deleted	Boolean	M		Status perintah Delete.	true

1.1.4. POST

End Point

<http://localhost:8083/teamC/v1/user-controller/createUser>

- Request URL

```
http://localhost:8083/swagger-ui/index.html#/user-controller/createUser
```



- Sample Request

```
{
  "username": "Luffy",
  "emailId": "luffybajaklaut@onepiece.com",
  "password": "1231231",
  "userId": 101,
  "id": 0
}
```

- Sample Response Request

Success Response

```
{
  "data": [
    {
      "ID": 62,
      "Username": "Luffy",
      "Email": "luffybajaklaut@onepiece.com",
      "Password": "1231231"
    }
  ],
  "message": "Successfully Created User!",
  "status": 201
}
```

Failed Response

```
{
  "data": "User Already Exist!",
  "message": "could not execute statement; SQL [n/a]; constraint [null]; nested exception is org.hibernate.exception.ConstraintViolationException: could not execute statement",
  "status": 400
}
```

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
-------	-----------	-----------	--------	-------------	---------



username	String	M		User name konsumen baru yang ingin diinsert.	Luffy
emailId	String	M		Email konsumen baru yang ingin diinsert.	luffybajaklaut@onepiece.com
password	String	M		Password konsumen baru yang ingin diinsert.	1231231
userId	Long	M		User Id konsumen baru yang ingin diinsert.	101

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
username	String	M		User name konsumen yang berhasil diinsert.	Luffy
emailId	String	M		Email konsumen yang berhasil diinsert.	luffybajaklaut@onepiece.com
password	String	M		Password konsumen yang berhasil diinsert.	1231231
userId	Long	M		User Id konsumen yang berhasil diinsert.	101

1.1.5. GET ALL

End Point :

<http://localhost:8083/teamC/v1/user-controller/getAllUser>

- Request URL

<http://localhost:8083/swagger-ui/index.html#/user-controller/getAllUser>

- Sample Request

- Sample Response Request

Success Response
{ "data": [



```
{
  "userId": 1,
  "username": "Wahyu Wijanarko",
  "emailId": "WahyuW@gmail.com",
  "password": "Rahasiaaaa4x"
},
.....
{
  "userId": 62,
  "username": "Luffy",
  "emailId": "luffybajaklaut@onepiece.com",
  "password": "1231231"
}
],
"message": "Successfully Get All User!",
"status": 200
}
```

Failed Response
<pre>{ "data": [], "message": "Tabel has no value!", "status": 404 }</pre>

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
username	String	M		User name yang dimiliki oleh konsumen.	Lily
emailId	String	M		Email yang digunakan oleh konsumen saat mendaftar.	Lily@gmail.com
password	String	M		Password yang diinput oleh konsumen saat mendaftar.	123456aaaa1
userId	Long	M		User Id yang diberikan system untuk konsumen.	9

1.2. Seats

1.2.1. GET By Id



- End Point :
`http://localhost:8083/seats-controller/getseatsById`
- Request URL

`http://localhost:8083/swagger-ui/index.html#/seats-controller/getseatsById`

- Sample Request

```
{  
  seatId : 3  
}
```

- Sample Response Request

Success Response

```
{  
  "data": 3,  
  "message": "Successfully retrieved data!",  
  "status": 200  
}
```

Failed Response

```
{  
  "data": "Data Not Found",  
  "message": " Seats not Exist with id :500",  
  "status": 404  
}
```

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
seatId	Long	M		Nomor Identifikasi seat dari system.	1
seatNumber	Long	M		Nomor Seat di Studio.	1
studioName	String	M		Nama Studio.	A
isAvailable	int	M		Status tersedia atau tidak,	1

1.2.2. PUT

End Point :



<http://localhost:8083/seats-controller/updateSeats>

- Request URL

<http://localhost:8083/swagger-ui/index.html#/seats-controller/updateSeats>

- Sample Request

```
{
  "seatId": 10,
  "seatNumber": 3,
  "studioName": "string",
  "isAvailable": 0
}
```

- Sample Response Request

Success Response

```
{
  "data": {
    "seatId": 10,
    "seatNumber": 3,
    "studioName": "string",
    "isAvailable": 0
  },
  "message": "Success Update Seats",
  "status": 200
}
```

Failed Response

```
{
  "data": "data not found",
  "message": "Seats not Exist with id :500",
  "status": 404
}
```

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
seatId	Long	M		Seats Id kosumen yang ingin di	1



				update.	
seatNumber	Long	M		Nomor Seats konsumen yang ingin di update.	2
studioName	String	M		Nama Studio konsumen yang ingin di update.	
isAvailable	int	M		Status Seats konsumen yang ingin di update.	0

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
seatId	Long	M		Seats Id baru yang telah diupdate.	1
seatNumber	Long	M		Nomor Seats yang telah diupdate.	2
studioName	String	M		Nama Studio yang telah diupdate.	C
isAvailable	int	M		Status Seats konsumen yang telah diupdate	0

1.2.3. DELETE

- End Point :

<http://localhost:8083/seats-controller/deleteseats>

- Request URL

<http://localhost:8083/swagger-ui/index.html#/seats-controller/deleteseats>

- Sample Request

```
{  
  seatId : 3  
}
```

- Sample Response Request

Success Response
<pre>{ "data": 3, "message": "Deleted! ", "status": 200 }</pre>



```
}
```

Failed Response

```
{
  "data": "Data Not Found",
  "message": "Seats not Exist with id :500",
  "status": 404
}
```

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
seatId	Long	M		Seat Id yang ingin di delete.	52

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
deleted	Boolean	M		Status perintah Delete.	true

1.2.4. GET ALL

- End Point :
<http://localhost:8083/seats-controller/getSeats>
- Request URL

```
http://localhost:8083/swagger-ui/index.html#/seats-controller/getSeats
```

- Sample Request

- Sample Response Request

Success Response

```
{
  "data": [
    {
      "seatId": 1,
```



```
"seatNumber": 1,
"studioName": "A",
"isAvailable": 1
},
.....
{
  "seatId": 50,
  "seatNumber": 10,
  "studioName": "E",
  "isAvailable": 1
}
],
"message": "Succesfully Read All Data Seats !",
"status": 200
}
```

Failed Response
<pre>{ "data": "Table has no value", "message": "table seats have not value", "status": 404 }</pre>

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
seatId	Long	M		Nomor Identifikasi seat dari system.	2
seatNumber	Long	M		Nomor Seat di Studio.	2
studioName	String	M		Nama Studio.	A
isAvailable	int	M		Status tersedia atau tidak,	1

1.2.5. POST

- End Point :
<http://localhost:8083/seats-controller/createseats>
- Request URL



<http://localhost:8083/swagger-ui/index.html#/seats-controller/createseats>

- Sample Request

```
{
  "seatId": 2,
  "seatNumber": 2,
  "studioName": "A",
  "isAvailable": 1
}
```

- Sample Response Request

Success Response

```
{
  "data": {
    "seatId": 2,
    "seatNumber": 2,
    "studioName": "A",
    "isAvailable": 1
  },
  "message": "Succesfully Add Data Seats !",
  "status": 200
}
```

Failed Response

```
{
  "data": [],
  "message": "Tabel has no value!",
  "status": 404
}
```

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
seatId	Long	M		Seats Id User name konsumen baru yang ingin diinsert.	52
seatNumber	Long	M		Nomor Seats konsumen	1



				baru yang ingin diinsert	
studioName	String	M		Nama Studio konsumen baru yang ingin diinsert	B
isAvailable	int	M		Status Seats konsumen baru yang ingin diinsert	1

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
seatId	Long	M		Seats Id User name konsumen baru yang berhasil diinsert.	52
seatNumber	Long	M		Nomor Seats konsumen baru yang berhasil diinsert	1
studioName	String	M		Nama Studio konsumen baru yang berhasil diinsert	B
isAvailable	int	M		Status Seats konsumen baru yang berhasil diinsert	1

1.2.6. POST isAvailable

- End Point :
<http://localhost:8083/swagger-ui/index.html#/seats-controller/findSeats>
- Request URL

localhost:8083/teamC/v1/schedule

- Sample Request

```
{
  "seatId": 51 ,
  "seatNumber": 5,
  "studioName": "B",
  "isAvailable": 1
}
```

- Sample Response Request

Success Response
<pre>{ "data": [{</pre>



```
"seatId": 1,
"seatNumber": 1,
"studioName": "A",
"isAvailable": 1,
....
{
"seatId": 50,
"seatNumber": 10,
"studioName": "E",
"isAvailable": 1
},
],
"message": "Successfully seats is Available ! ",
"status": 200
}
```

Failed Response

● Request struktur

Field	Data Type	Mandatory	Length	Description	Example
isAvailable	int	M		Menampilkan semua Status Seats konsumen	1

● Response Structure

Field	Data Type	Mandatory	Length	Description	Example
seatId	Long	M		Nomor Identifikasi seat dari system.	2
seatNumber	Long	M		Nomor Seat di Studio.	2
studioName	String	M		Nama Studio.	A
isAvailable	int	M		Status tersedia atau tidak,	1

1.3. Schedule

1.3.1. GET ALL

● End Point :

<http://localhost:8083/teamC/v1/schedule-controller/ ScheduleList>



- Request URL

localhost:8083/teamC/v1/schedule

- Sample Request

http://localhost:8083/swagger-ui/index.html#/schedule-controller-dto/ScheduleList_1

- Sample Response Request

Success Response

```
{
  "data": {
    "scheduleId": 1,
    "filmId": 1,
    "seatId": 1,
    "dateShow": "2022-06-19",
    "showStart": "18:00:00",
    "showEnd": "19:00:00",
    "price": 35000
  },
  "message": "Successfully retrived data!",
  "status": 200
}
```

Failed Response

```
{
  "data": "Tabel has no Value",
  "message": "Schedule not exist",
  "status": 404
}
```

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
scheduleId	Integer	M		Nomor Identifikasi schedule dari system.	1
filmId	Integer	M		Nomor Identifikasi film dari system.	1



seatId	Integer	M		Nomor Identifikasi seat dari system.	1
dateShow	Date	M		Tanggal tayang.	2022-06-19
showStart	Time	M		Tanggal dan waktu film tayang.	3:00:00
showEnd	Time	M		Tanggal dan waktu film selesai.	15:00:00
price	Integer	M		Harga tiket.	35000

1.3.2. PUT SCHEDULE

- End Point :

<http://localhost:8083/swagger-ui/index.html#/schedule-controller/updateSchedule>

- Request URL

localhost:8083/teamC/v1/schedule

- Sample Request

```
{
  "scheduleId": 9,
  "films": {
    "filmId": 3
  },
  "seats": {
    "seatId": 7
  },
  "dateShow": "2022-06-19",
  "showStart": "00:00:00",
  "showEnd": "00:00:00",
  "price": 20000
}
```

- Sample Response Request

Success Response

```
{
  "data": {
    "scheduleId": 9,
```



```
"filmId": 3,
"seatId": 7,
"dateShow": "2022-06-19",
"showStart": "00:00:00",
"showEnd": "00:00:00",
"price": 20000
},
"message": "Data Updated!",
"status": 201
}
```

Failed Response
<pre>{ "data": "Bad Request", "message": "Schedule not exist with id : 900", "status": 400 }</pre>

● Request Structure

Field	Data Type	Mandatory	Length	Description	Example
scheduleId	Integer	M		Nomor Identifikasi schedule dari system.	1
Films	Entity	M		Entity/Model dari Films	
filmId	Integer	M		Nomor Identifikasi film dari system.	1
Seats	Entitiy			Entity/Model dari seats	
seatId	Integer	M		Nomor Identifikasi seat dari system.	1
dateShow	Date	M		Tanggal tayang.	2022-06-19
showStart	Time	M		Tanggal dan waktu film tayang.	3:00:00
showEnd	Time	M		Tanggal dan waktu film selesai.	15:00:00
price	Integer	M		Harga tiket.	35000

● Response Structure



Field	Data Type	Mandatory	Length	Description	Example
scheduleId	Integer	M		Nomor Identifikasi schedule dari system.	1
filmId	Integer	M		Nomor Identifikasi film dari system.	1
seatId	Integer	M		Nomor Identifikasi seat dari system.	1
dateShow	Date	M		Tanggal tayang.	2022-06-19
showStart	Time	M		Tanggal dan waktu film tayang.	3:00:00
showEnd	Time	M		Tanggal dan waktu film selesai.	15:00:00
price	Integer	M		Harga tiket.	35000

1.3.3. POST/CREATE Schedule

- End Point :

<http://localhost:8083/swagger-ui/index.html#/schedule-controller/createSchedule>

- Request URL

localhost:8083/teamC/v1/schedule

- Sample Request

```
{
  "films": {
    "filmId": 90
  },
  "seats": {
    "seatId": 7
  },
  "dateShow": "2022-06-19",
  "showStart": "00:00:00",
  "showEnd": "00:00:00",
  "price": 20000
}
```

- Sample Response Request

Success Response



```
{
  "data": {
    "scheduleId": 64,
    "filmId": 1,
    "seatId": 7,
    "dateShow": "2022-06-19",
    "showStart": "00:00:00",
    "showEnd": "00:00:00",
    "price": 20000
  },
  "message": "Successfully added data!",
  "status": 201
}
```

Failed Response

```
{
  "data": null,
  "message": "could not execute statement; SQL [n/a]; constraint [null]; nested exception is
org.hibernate.exception.ConstraintViolationException: could not execute statement",
  "status": 400
}
```

● Request Structure

Field	Data Type	Mandatory	Length	Description	Example
scheduleId	Integer	M		Nomor Identifikasi schedule dari system.	1
Films	Entity	M		Entity/Model dari Films	
filmId	Integer	M		Nomor Identifikasi film dari system.	1
Seats	Entitiy			Entity/Model dari seats	
seatId	Integer	M		Nomor Identifikasi seat dari system.	1
dateShow	Date	M		Tanggal tayang.	2022-06-19
showStart	Time	M		Tanggal dan waktu film tayang.	3:00:00
showEnd	Time	M		Tanggal dan waktu film	15:00:00



				selesai.	
price	Integer	M		Harga tiket.	35000

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
scheduleId	Integer	M		Nomor Identifikasi schedule dari system.	1
filmId	Integer	M		Nomor Identifikasi film dari system.	1
seatId	Integer	M		Nomor Identifikasi seat dari system.	1
dateShow	Date	M		Tanggal tayang.	2022-06-19
showStart	Time	M		Tanggal dan waktu film tayang.	3:00:00
showEnd	Time	M		Tanggal dan waktu film selesai.	15:00:00
price	Integer	M		Harga tiket.	35000

1.3.4. POST/FIND Schedule By Film Name

- End Point :

<http://localhost:8083/swagger-ui/index.html#/schedule-controller/findScheduleByFilmName>

- Request URL

localhost:8083/schedule/byfilmname

- Sample Request

{ "name": "Avatar 2" }

- Sample Response Request

Success Response
{ "data": [{



```
{
  "filmName": "Avatar 2",
  "studioName": "A",
  "price": 20000
},
{
  "message": "Success",
  "status": 200
}
```

Failed Response
<pre>{ "data": "Bad Request", "message": "Schedule with film name Comic 8888 is not exist!!!!", "status": 400 }</pre>

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
name	String	M		Nama film.	1

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
filmName	String	M		Nama Film	Avatar 2
studioName	String	M		Nama Studio	A
price	Integer	M		harga Film	25000

1.3.5. POST/FIND Schedule By Film Name Like

- End Point :

<http://localhost:8083/swagger-ui/index.html#/schedule-controller/findScheduleByFilmNameLike>

- Request URL

localhost:8083/schedule/ByfilmnameLike
--

- Sample Request

```
{
  "name": "Avatar"
}
```



- Sample Response Request

Success Response
<pre>{ "data": [{ "filmName": "Avatar 2", "studioName": "A", "price": 20000 }], "message": "Success", "status": 200 }: 200 }</pre>

Failed Response
<pre>{ "data": "Data Not Found", "message": "Schedule with film name Comichghb jb is not exist!!!!", "status": 404 }</pre>

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
name	String	M		Nama film.	1

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
filmName	String	M		Nama Film	Avatar 2
studioName	String	M		Nama Studio	A
price	Integer	M		harga Film	25000

1.3.6. GET By ID

- End Point :

<http://localhost:8083/swagger-ui/index.html#/schedule-controller/getscheduleById>

- Request URL



localhost:8083/teamC/v1/schedule/33

- Sampel Request

```
{  
  "scheduleId": 33  
}
```

- Sampel Response

Success Response

```
{  
  "data": {  
    "scheduleId": 3,  
    "filmId": 3,  
    "seatId": 3,  
    "dateShow": "2022-06-19",  
    "showStart": "18:00:00",  
    "showEnd": "19:00:00",  
    "price": 35000  
  },  
  "message": "Successfully retrivied data!",  
  "status": 200  
}
```

Failed Response

```
{  
  "data": "Data Not Found!",  
  "message": "Schedule not exist with id : 1000",  
  "status": 404  
}
```

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
-------	-----------	-----------	--------	-------------	---------



scheduleId	int	M		Nomor Identifikasi schedule dari system.	3
filmId	int	M		Nomor Identifikasi film dari system.	3
name	String	M		Nama film.	Avatar 2
isPlaying	Integer	M		Status sedang tayang atau tidak.	0
dateShow	Date	M		Tanggal tayang.	2022-06-19T00:00:00.000+00:00
seatId	Long	M		Nomor Identifikasi seat dari system.	5
seatNumber	Long	M		Nomor kursi.	4
studioName	String	M		Nama Studio.	E
isAvailable	int	M		Status kursi tersedia atau tidak.	0
showStart	Date	M		Tanggal dan waktu film tayang.	2022-06-19T13:00:00.000+00:00
showEnd	Date	M		Tanggal dan waktu film selesai.	2022-06-19T15:00:00.000+00:00
price	Integer	M		Harga tiket.	35000

1.3.7. DELETE

- End Point :

<http://localhost:8083/swagger-ui/index.html#/schedule-controller/delete>

-

- Request URL

localhost:8083/schedule/15

- Sampel Request

```
{
  "scheduleId": 15
}
```

- Sampel Response

Success Response

```
{
```



```
"data": {
  "deleted": true
},
"message": "Deleted! ",
"status": 200
}
```

Failed Response

```
{
"data": "Data Not Found",
"message": "Schedule not exist with id :15",
"status": 404
}
```

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
scheduleId	Long	M		Seat Id yang ingin di delete.	13

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
deleted	Boolean	M			true

1.4. Film

1.4.1. GET By Id

- End Point :

http://localhost:8083/teamC/v1/films-controller/getUserById_1

- Request URL

http://localhost:8083/swagger-ui/index.html#/schedule-controller-dto/ScheduleList_1

- Sampel Request

```
{
"filmId": 1
}
```



- Sampel Response

Success Response
<pre>{ "data": { "filmId": 1, "name": "Jurassic World Dominion", "isPlaying": 1 }, "message": "Succes Get", "status": 200 }</pre>

Failed Response
<pre>{ "data": "Data not found", "message": "Films with id 100 is not exist", "status": 404 }</pre>

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
filmId	Long	M		Nomor Identifikasi film dari system.	1
name	String	M		Nama film.	Jurassic World Dominion
isPlaying	Integer	M		Film tersedia atau tidak.	1

1.4.2. PUT

- End Point :

<http://localhost:8083/teamC/v1/films-controller/updateFilms>

- Request URL

http://localhost:8083/teamC/v1/DTO/films/65

- Sampel Request



```
{  
  
  "title": "kamu",  
  "status": 0  
}
```

- Sampel Response

Success Response
<pre>{ "data": { "filmId": 65, "name": "kamu", "isPlaying": 0 }, "message": "Sukses Update", "status": 201 }</pre>

Failed Response
<pre>{ "data": "Data not found", "message": "Films with id 100 is can't to update", "status": 404 }</pre>

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
filmId	Long	M		Film Id konsumen yang Berhasil diupdate..	1
name	String	M		Nama film konsumen yang berhasil diupdate..	kamu
isPlaying	Integer	M		Status tayang film yang berhasil diupdate.	0

1.4.3. DELETE



- End Point :

[http://localhost:8083/ films-controller/deleteFilms](http://localhost:8083/films-controller/deleteFilms)

- Request URL

<http://localhost:8083/swagger-ui/index.html#/films-controller-dto/deleteFilm>

- Sampel Request

```
{  
  "filmId": 1  
}
```

- Sampel Response

Success Response

```
{  
  "data": {  
    "deleted": true  
  },  
  "message": "Succes Delete",  
  "status": 200  
}
```

Failed Response

```
{  
  "data": "Failed delete data",  
  "message": "Films with id 100 is failed to delete",  
  "status": 404  
}
```

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
filmId	Long	M		Film Id yang ingin di delete.	20

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
deleted	Boolean	M		Status perintah Delete.	true

1.4.4. GET ALL



- End Point

<http://localhost:8083/teamC/v1/films-controller/findAllFilms>

- Request URL

http://localhost:8083/swagger-ui/index.html#/films-controller-dto/findAllFilms_1

- Sampel Request

- Sampel Response

Success Response

```
{
  "data": [
    {
      "filmId": 1,
      "name": "Jurassic World Dominion",
      "isPlaying": 1
    },
    {
      "filmId": 66,
      "name": "GERAKAN UJI NYALI",
      "isPlaying": 1
    }
  ],
  "message": "Succes All",
  "status": 200 }
]
```

Failed Response

```
{
  "data": "Table has no value",
  "message": "Table films has not value",
  "status": 404
}
```



- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
filmId	Long	M		Nomor Identifikasi film dari system.	1
name	String	M		Nama film.	kamu
isPlaying	Integer	M		Film tersedia atau tidak.	0

1.4.5. POST

- End Point

<http://localhost:8083/teamC/v1/films-controller/createFilm>

- Request URL

http://localhost:8083/swagger-ui/index.html#/films-controller-dto/createFilm_1

- Sampel Request

```
{
  "name": "dzikir 24 hours",
  "isPlaying": 1
}
```

- Sampel Response

Success Response

```
{
  "data": {
    "filmId": 69,
    "name": "dzikir 24 hours",
    "isPlaying": 0
  },
  "message": "Succes Create",
  "status": 200
}
```

Failed Response

```
{
  "data": "Failed create data",
  "message": "could not execute statement; SQL [n/a]; constraint [null]; nested exception is
```



```
org.hibernate.exception.ConstraintViolationException: could not execute statement",
  "status": 400
}
```

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
filmId	Long	M		Film Id konsumen yang ingin di insert..	62
name	String	M		Nama film konsumen yang ingin di insert.	dzikir 24 hours
isPlaying	Integer	M		Status tayang film tersedia tau tidak yang ingin di insert.	1

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
filmId	Long	M		Film Id konsumen yang berhasil di insert.	62
name	String	M		Nama film konsumen yang berhasil di insert.	dzikir 24 hours
isPlaying	Integer	M		Status tayang film tersedia atau tidak yang berhasil di insert.	1

1.5. Booking

1.5.1. GET ALL

- End Point :

<http://localhost:8083/teamC/v1/DTO/booking>

- Request URL

localhost:8083/teamC/v1/booking

- Sample Response Request

Success Response
<pre>{ "data": [{ "book_id": 1,</pre>



```
"usr_id": 1,
"usr_name": "Rudi",
"email_usr": "email@email.com",
"film_name": "Doraemon",
"price": 30000,
"studio": "Studio A",
"status_show": 1,
"seat_num": 1,
"status_seat": 1,
"date_film": "2022-06-01",
"start_film": "14:00:00",
"end_film": "16:00:00"
},
.....
{
  "book_id": 10,
  "usr_id": 10,
  "usr_name": "Agung",
  "email_usr": "agung@email.com",
  "film_name": "Ms. Marvel",
  "price": 30000,
  "studio": "Studio G",
  "status_show": 1,
  "seat_num": 2,
  "status_seat": 1,
  "date_film": "2022-06-08",
  "start_film": "12:00:00",
  "end_film": "14:00:00"
}
],
"message": "Success Get All",
"status": 200
}
```

Failed Response					
{ "data": "Table has no value", "message": "Data Booking not exist", "status": 404 }					

● Response Structure

Field	Data Type	Mandatory	Length	Description	Example
book_id	Integer	M		Nomor Identifikasi booking dari	1



				system.	
usr_id	Integer	M		Nomor Identifikasi user dari system.	1
usr_name	String	M		Nama user	Budiman Piatu
email_usr	String	M		Email User	BudimanPiatu@gmail.com
film_name	String	M		Nama film	Jurassic World Domination
price	Integer	M		Harga tiket film	20000
studio	String	M		Nama Studio	A
status_show	Integer	M		Film tayang (1) atau tidak (0)	1
seat_num	Integer	M		Nomor kursi	69
status_seat	Integer	M		Ketersediaan kursi (1) atau tidak (0)	1
date_film	date	M		Tanggal Tayang Film	2022-06-19
start_film	time	M		Waktu Mulai Film	18:00:00
end_film	time	M		Waktu Selesai Film	19:00:00

1.5.2. GET BY ID

- End Point :

<http://localhost:8083/teamC/v1/DTO/booking/{id}>

- Request URL

localhost:8083/teamC/v1/booking/{id}

- Sample Response Request

Success Response
<pre>{ "data": { "book_id": 1, "usr_id": 1, "usr_name": "Rudi", "email_usr": "email@email.com", "film_name": "Doraemon", "price": 30000, "studio": "Studio A",</pre>



```
"status_show": 1,
"seat_num": 1,
"status_seat": 1,
"date_film": "2022-06-01",
"start_film": "14:00:00",
"end_film": "16:00:00"
},
"message": "Success Get By Id",
"status": 200
}
```

Failed Response

```
{
  "data": "Data not found",
  "message": "Booking not exist with id 100",
  "status": 404
}
```

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
book_id	Integer	M		Nomor Identifikasi booking dari system.	1
usr_id	Integer	M		Nomor Identifikasi user dari system.	1
usr_name	String	M		Nama user	Budiman Piatu
email_usr	String	M		Email User	BudimanPiatu@gmail.com
film_name	String	M		Nama film	Jurassic World Domination
price	Integer	M		Harga tiket film	20000
studio	String	M		Nama Studio	A
status_show	Integer	M		Film tayang (1) atau tidak (0)	1
seat_num	Integer	M		Nomor kursi	69
status_seat	Integer	M		Ketersediaan kursi (1) atau tidak (0)	1
date_film	date	M		Tanggal Tayang Film	2022-06-19
start_film	time	M		Waktu Mulai Film	18:00:00
end_film	time	M		Waktu Selesai Film	19:00:00



1.5.3. CREATE/POST

- End Point :
`http://localhost:8083/teamC/v1/DTO/booking`
- Request URL

`localhost:8083/teamC/v1/DTO/booking`

- Sample Request

```
{
  "usr": {
    "userId": 15
  },
  "sch": {
    "scheduleId": 1
  }
}
```

- Sample Response Request

Success Response

```
{
  "data": {
    "book_id": 15,
    "sch_id": 1,
    "usr_id": 15
  },
  "message": "Success Create Booking",
  "status": 201
}
```

Failed Response

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
Usr	Entity/Mode 1	M		Enitivity.User	
userId	Integer	M		Nomor Identifikasi user dari system.	1
Sch	Entity/Mode 1	M		Entity Schedule	



scheduleId	String	M		Nomor Identifikasi schedule dari System	1
------------	--------	---	--	---	---

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
book_id		M		Nomor Identifikasi booking dari system.	11
sch_id	Integer	M		Nomor Identifikasi schedule dari system.	12
usr_id	Integer	M		Nomor Identifikasi user dari system.	13

1.5.4. UPDATE/PUT

- End Point :

<http://localhost:8083/teamC/v1/DTO/booking/{id}>

- Request URL

localhost:8083/teamC/v1/DTO/booking/{id}
--

- Sample Request

<pre>{ "usr": { "userId": 15 }, "sch": { "scheduleId": 1 } }</pre>
--

- Sample Response Request

Success Response
<pre>{ "data": { "book_id": 15, "usr_id": 4, "usr_name": "Nur Aida", "email_usr": "Aida_Nur@gmail.com", "film_name": "Sonic the Hedgehog 2", "price": 35000, } }</pre>



```
{
  "studio": "A",
  "status_show": 1,
  "seat_num": 4,
  "status_seat": 1,
  "date_film": "2022-06-19",
  "start_film": "18:00:00",
  "end_film": "19:00:00"
},
{
  "message": "Success Update Booking",
  "status": 201
}
```

Failed Response
<pre>{ "data": "Bad Request", "message": "Booking not exist with id 200", "status": 400 }</pre>

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
Usr	Entity/Model	M		Entity.User	
userId	Integer	M		Nomor Identifikasi user dari system.	1
Sch	Entity/Model	M		Entity Schedule	
scheduleId	String	M		Nomor Identifikasi schedule dari System	1

- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
book_id	Integer	M		Nomor Identifikasi booking dari system.	1
usr_id	Integer	M		Nomor Identifikasi user dari system.	1
usr_name	String	M		Nama user	Budiman Piatu
email_usr	String	M		Email User	BudimanPiatu@gmail.com
film_name	String	M		Nama film	Jurassic



					World Domination
price	Integer	M		Harga tiket film	20000
studio	String	M		Nama Studio	A
status_show	Integer	M		Film tayang (1) atau tidak (0)	1
seat_num	Integer	M		Nomor kursi	69
status_seat	Integer	M		Ketersedian kursi (1) atau tidak (0)	1
date_film	date	M		Tanggal Tayang Film	2022-06-19
start_film	time	M		Waktu Mulai Film	18:00:00
end_film	time	M		Waktu Selesai Film	19:00:00

1.5.5. DELETE

- End Point :
`http://localhost:8083/teamC/v1/DTO/booking/{id}`
- Request URL

`localhost:8083/teamC/v1/DTO/booking/{id}`

- Sample Response Request

Success Response

```
{
  "data": true,
  "message": "Success Delete Booking by ID",
  "status": 200
}
```

Failed Response

```
{
  "data": "Data not found",
  "message": "Booking not exist with id 100",
  "status": 404
}
```

1.5.6. POST/QUERY BY FILMNAME

- End Point :
`http://localhost:8083/teamC/v1/DTO/booking/Filmname`
- Request URL

`localhost:8083/teamC/v1/DTO/booking/Filmname`



- Sample Request

```
{
  "name": "avatar"
}
```

- Sample Response Request

Success Response

```
{
  "data": [
    {
      "book_id": 2,
      "usr_id": 6,
      "usr_name": "Lionel",
      "email_usr": "Lionel@gmail.com",
      "film_name": "Avatar 2",
      "price": 35000,
      "studio": "A",
      "status_show": 1,
      "seat_num": 3,
      "status_seat": 1,
      "date_film": "2022-06-19",
      "start_film": "18:00:00",
      "end_film": "19:00:00"
    }
  ],
  "message": "Succes Query By Filmname",
  "status": 200
}
```

Failed Response

```
{
  "data": "Failed Query By Filename",
  "message": "Booking not exist with Filmname cinta itu nyata",
  "status": 400
}
```

- Request Structure

Field	Data Type	Mandatory	Length	Description	Example
name	String	M		Filmname	Avatar



- Response Structure

Field	Data Type	Mandatory	Length	Description	Example
book_id	Integer	M		Nomor Identifikasi booking dari system.	1
usr_id	Integer	M		Nomor Identifikasi user dari system.	1
usr_name	String	M		Nama user	Budiman Piatu
email_usr	String	M		Email User	BudimanPiatu@gmail.com
film_name	String	M		Nama film	Jurassic World Domination
price	Integer	M		Harga tiket film	20000
studio	String	M		Nama Studio	A
status_show	Integer	M		Film tayang (1) atau tidak (0)	1
seat_num	Integer	M		Nomor kursi	69
status_seat	Integer	M		Ketersedian kursi (1) atau tidak (0)	1
date_film	date	M		Tanggal Tayang Film	2022-06-19
start_film	time	M		Waktu Mulai Film	18:00:00
end_film	time	M		Waktu Selesai Film	19:00:00