

Nama: Gusti Fahmi Fadhila

Kode Peserta: JVS001ONL012

Link Heroku: <https://final-project-gustifahmi.herokuapp.com>

Link Github: <https://github.com/gustifahmi/FinalProjectSpringGusti>

## **Panduan Final Project – Gusti Fahmi Fadhila**

### **Daftar Isi**

A. Table & Columns	2
B. Database Migration	6
C. Controller List	9
D. Authentication Instructions	11
E. API Details	12
1. User Controller	12
2. Auth Controller	12
3. Reservation Controller	14
4. Agency Controller	17
5. Bus Controller	20
6. Stop Controller	23
7. Ticket Controller	25
8. Trip Controller	26
9. Trip Schedule Controller	30

## A. Tables & Columns (Database menggunakan PostgreSQL, tipe data berdasarkan yang tertera di PostgreSQL)

### 1. agency

- id bigserial -> Primary Key
- code varchar(255)
- details varchar(255)
- name varchar(255)
- owner\_user\_id int8 -> Foreign Key ke table user\_info

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
id	1	bigserial			[v]	nextval('agency_id_seq'::regclass)	
code	2	varchar(255)		default	[ ]		
details	3	varchar(255)		default	[ ]		
name	4	varchar(255)		default	[ ]		
owner_user_id	5	int8			[ ]		

Gambar 1. Screenshot DBeaver table agency properties

### 2. bus

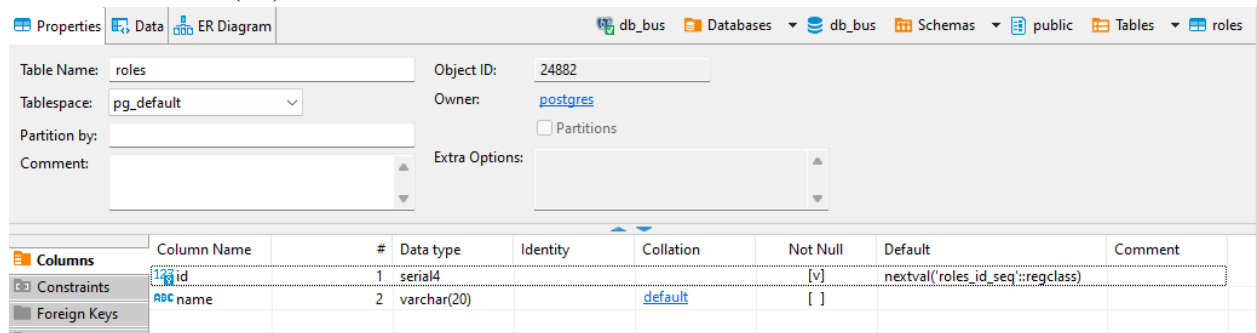
- id bigserial -> Primary Key
- capacity int4
- code varchar(255)
- make varchar(255)
- agency\_id int8 -> Foreign Key ke table agency

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
id	1	bigserial			[v]	nextval('bus_id_seq'::regclass)	
capacity	2	int4			[v]		
code	3	varchar(255)		default	[ ]		
make	4	varchar(255)		default	[ ]		
agency_id	5	int8			[ ]		

Gambar 2. Screenshot DBeaver table bus properties

### 3. roles

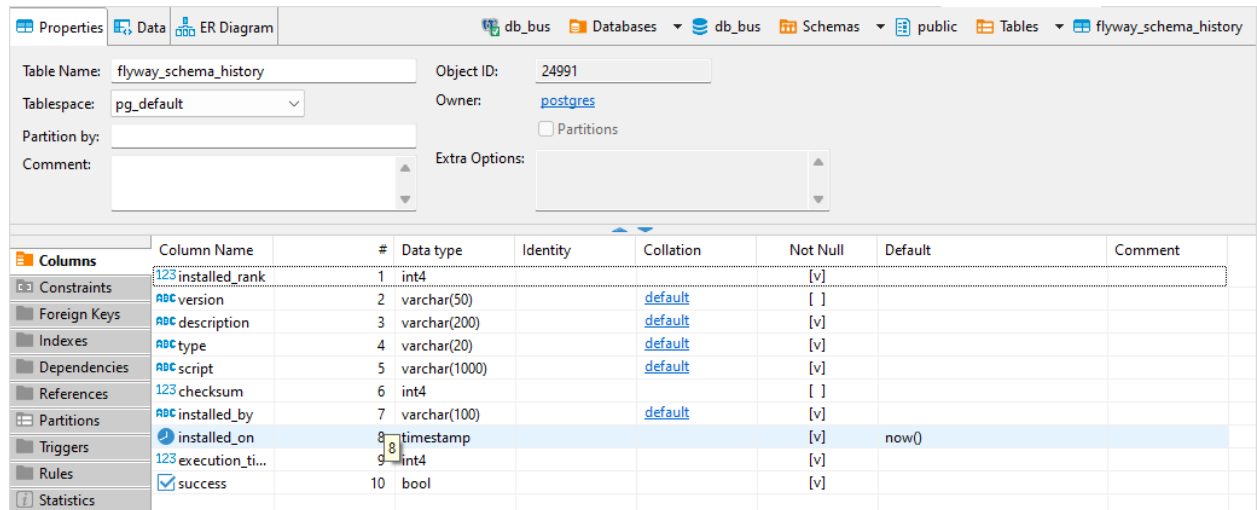
- id serial4 -> Primary Key
- name varchar(20)



Gambar 3. Screenshot DBeaver table roles properties

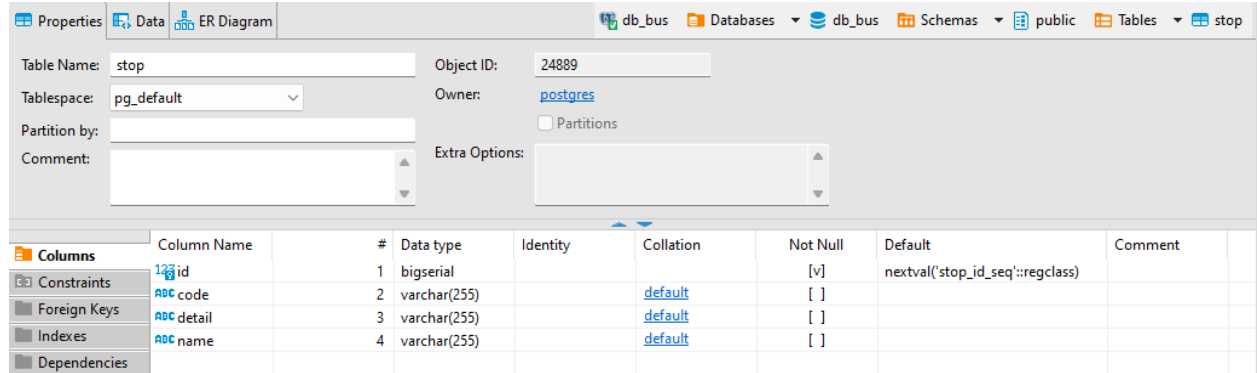
### 4. flyway\_schema\_history (ter-generate otomatis oleh flyway)

- installed\_rank int4
- version varchar(50)
- description varchar(200)
- type varchar(20)
- script varchar(1000)
- checksum int4
- installed\_by varchar(100)
- installed\_on timestamp
- execution\_time int4
- success bool



Gambar 4. Screenshot DBeaver table flyway\_schema\_history properties

5. stop
  - id bigserial -> Primary Key
  - code varchar(255)
  - detail varchar(255)
  - name varchar(255)



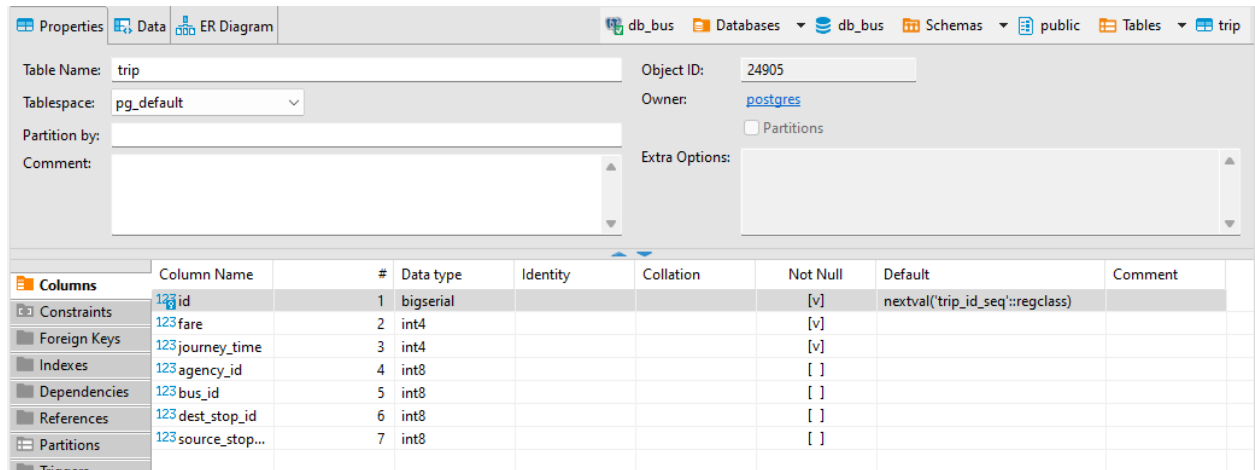
Gambar 5. Screenshot DBeaver table stop properties

6. ticket
  - id bigserial -> Primary Key
  - cancellable bool
  - journey\_date varchar(255)
  - seat\_number int4
  - passenger\_id int8 -> Foreign Key ke table user\_info
  - trip\_schedule\_id int8 -> Foreign Key ke table trip\_schedule

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
id	1	bigserial			[v]	nextval('ticket_id_seq'::regclass)	
cancellable	2	bool			[ ]		
journey_date	3	varchar(255)		default	[ ]		
seat_number	4	int4			[v]		
passenger_id	5	int8			[ ]		
trip_schedule_id	6	int8			[ ]		

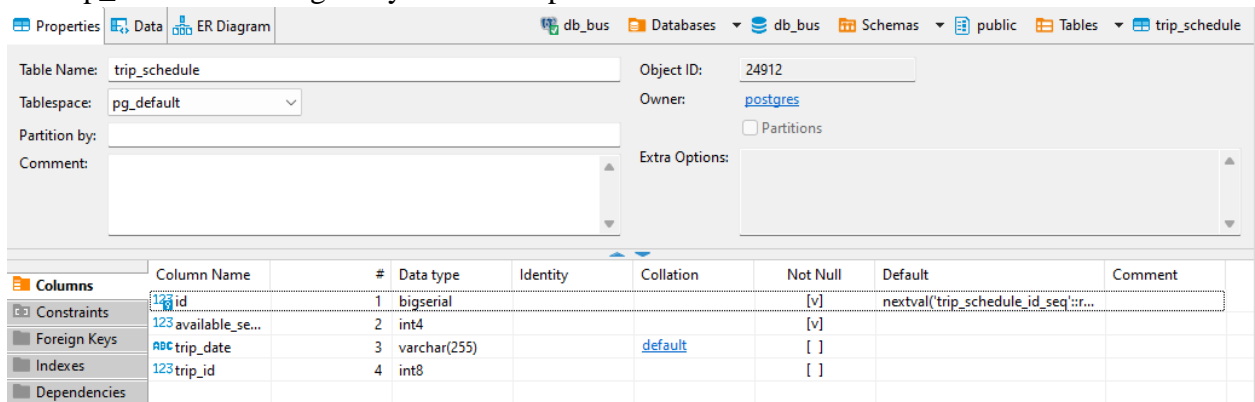
Gambar 6. Screenshot DBeaver table ticket properties

7. trip
  - id bigserial -> Primary Key
  - fare int4
  - journey\_time int4
  - agency\_id int8 -> Foreign Key ke table agency
  - bus\_id int8 -> Foreign Key ke table bus
  - dest\_stop int8 -> Foreign Key ke table stop
  - source\_stop int8 -> Foreign Key ke table stop



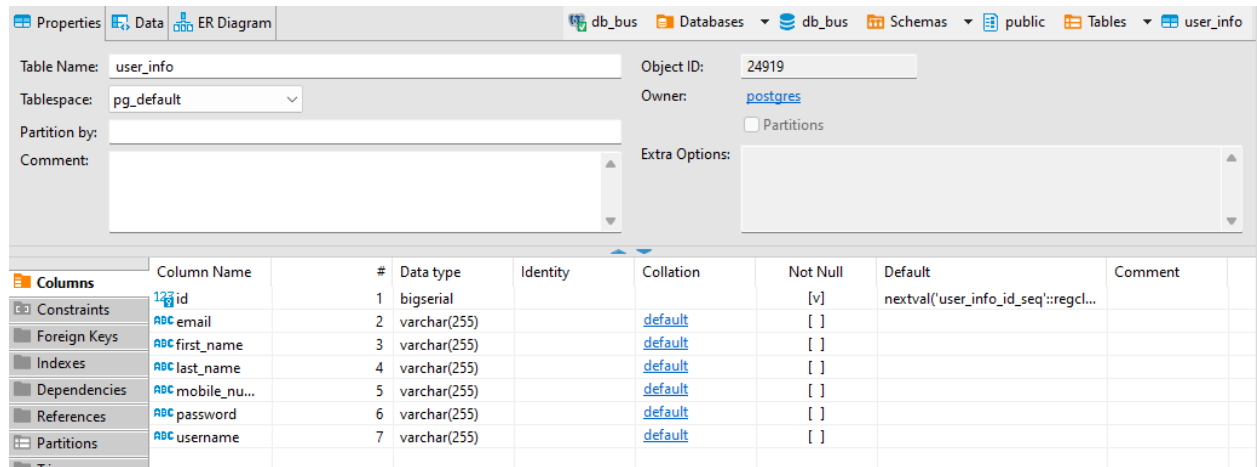
Gambar 7. Screenshot DBeaver table trip properties

8. trip\_schedule
  - id bigserial -> Primary Key
  - available\_seats int4
  - trip\_date varchar(255)
  - trip\_id int8 -> Foreign Key ke table trip



Gambar 8. Screenshot DBeaver table trip\_schedule properties

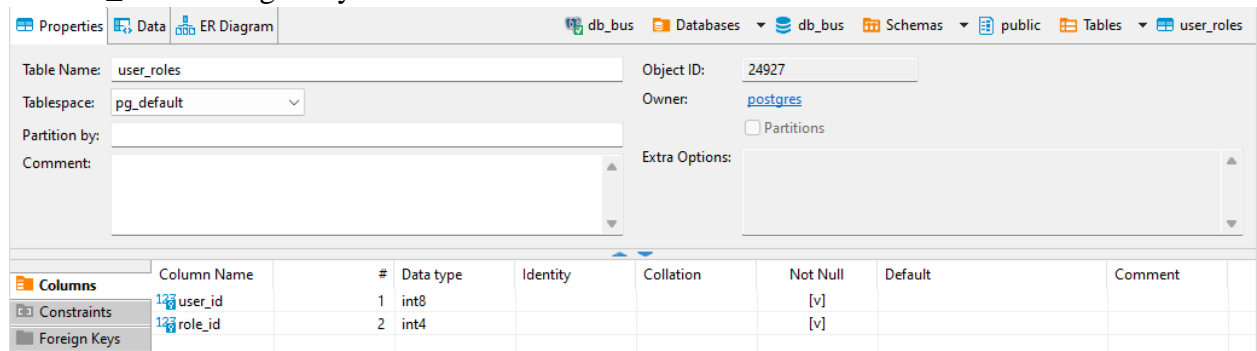
9. user\_info (di PostgreSQL tidak bisa menggunakan nama table "user")
  - id bigserial -> Primary Key
  - email varchar(255)
  - first\_name varchar(255)
  - last\_name varchar(255)
  - mobile\_number varchar(255)
  - password varchar(255)
  - username varchar(255)



Gambar 9. Screenshot DBeaver table user\_info properties

#### 10. user\_roles

- user\_id -> Foreign Key ke table user\_info
- role\_id -> Foreign Key ke table roles

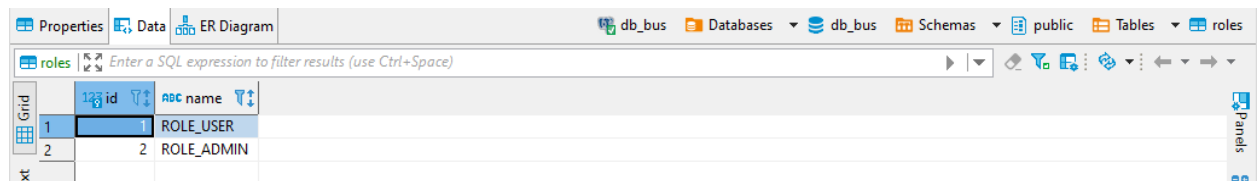


Gambar 10. Screenshot DBeaver table user\_roles properties

## B. Database Migration

### 1. role

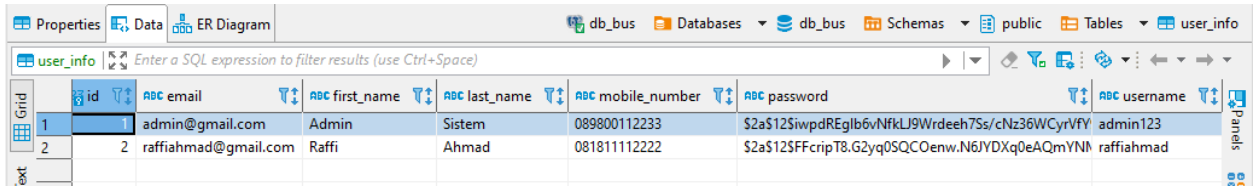
id	name
1	ROLE_USER
2	ROLE_ADMIN



Gambar 11. Screenshot DBeaver table roles data

2. user\_info (passwordnya terenkripsi, tapi di table ini saya tampilkan raw-nya agar bisa dipakai untuk login)

id	email	first_name	last_name	mobile_number	password	username
1	admin@gmail.com	Admin	Sistem	089800112233	admin123	admin123
2	raffiahmad@gmail.com	Raffi	Ahmad	081811112222	password	raffiahmad

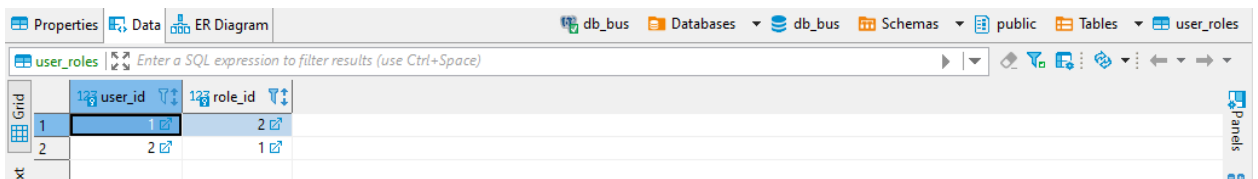


id	email	first_name	last_name	mobile_number	password	username
1	admin@gmail.com	Admin	Sistem	089800112233	\$2a\$12\$iwpdREglb6vNfkLJ9Wrdeeh7Ss/cNz36WCyrVfY	admin123
2	raffiahmad@gmail.com	Raffi	Ahmad	081811112222	\$2a\$12\$FFcripT8.G2yq0SQCOenw.N6/JYDXq0eAQmYNN	raffiahmad

Gambar 12. Screenshot DBeaver table user\_info data

3. user\_roles

user_id	role_id
1	2
2	1

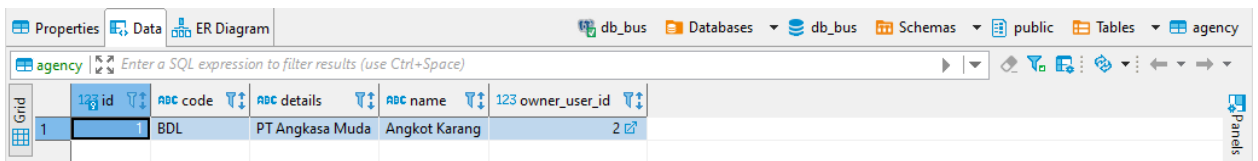


user_id	role_id
1	2
2	1

Gambar 13. Screenshot DBeaver table user\_roles data

4. agency

id	code	details	name	owner_user_id
1	BDL	PT Angkasa Muda	Angkot Karang	2



id	code	details	name	owner_user_id
1	BDL	PT Angkasa Muda	Angkot Karang	2

Gambar 14. Screenshot DBeaver table agency data

5. bus

id	capacity	code	make	agency_id
1	20	BDL01	20	1
2	15	BDL02	15	1

	id	capacity	code	make	agency_id
1	1	20	BDL01	20	1
2	2	15	BDL02	15	1

Gambar 15. Screenshot DBeaver table bus data

6. stop

id	code	detail	name
1	1-2	Teluk	Karang
2	1-8	Cimeng	Sukaraja 1-8
3	1-3	Raja Basal	Kedaton 1-3
4	1-2	Sukabumi	Sukarame 1-2
5	1-10	Tanjung Senang	Jati Agung 1-10

	id	code	detail	name
1	1	1-2	Teluk	Karang
2	2	1-8	Cimeng	Sukaraja 1-8
3	3	1-3	Raja Basal	Kedaton 1-3
4	4	1-2	Sukabumi	Sukarame 1-2
5	5	1-10	Tanjung Senang	Jati Agung 1-10

Gambar 16. Screenshot DBeaver table stop data

7. trip

id	fare	journey_time	agency_id	bus_id	dest_stop_id	source_stop_id
1	15000	15	1	1	3	1
2	15000	2	1	1	5	2
3	15000	3	1	1	2	3
4	15000	5	1	2	3	4
5	15000	4	1	2	2	5
6	15000	7	1	2	1	4

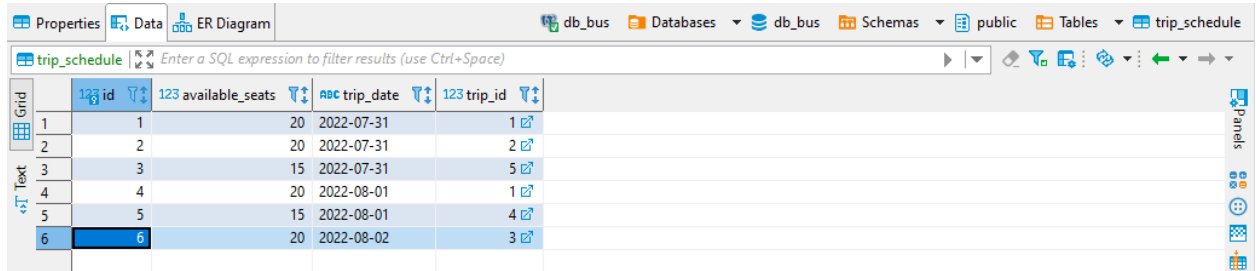
	id	fare	journey_time	agency_id	bus_id	dest_stop_id	source_stop_id
1	1	15,000	15	1	1	3	1
2	2	15,000	2	1	1	5	2
3	3	15,000	3	1	1	2	3
4	4	15,000	5	1	2	3	4
5	5	15,000	4	1	2	2	5
6	6	15,000	7	1	2	1	4

Gambar 17. Screenshot DBeaver table trip data



### 8. trip\_schedule

id	available_seats	trip_date	trip_id
1	20	2022-07-31	1
2	20	2022-07-31	2
3	15	2022-07-31	5
4	20	2022-08-01	1
5	15	2022-08-01	4
6	20	2022-08-02	3



id	available_seats	trip_date	trip_id
1	20	2022-07-31	1
2	20	2022-07-31	2
3	15	2022-07-31	5
4	20	2022-08-01	1
5	15	2022-08-01	4
6	20	2022-08-02	3

Gambar 18. Screenshot DBeaver table trip\_schedule data

## C. Controller List

### 1. User Controller

Name	Endpoint	Method	Admin	User
registerUser	/api/v1/user/signup	POST	No Login	No Login
updateUser	/api/v1/user/update	PUT	✓	✓
updatePassword	/api/v1/user/update/password	PUT	✓	✓

Catatan: Untuk update user dan password tidak perlu menggunakan user id pada URL param karena otomatis didapatkan berdasarkan user yang login.

### 2. Auth Controller

Name	Endpoint	Method	Admin	User
authenticateUser	/api/auth	POST	No Login	No Login

### 3. Reservation Controller

Name	Endpoint	Method	Admin	User
bookTicket	/api/v1/reservation	POST	✗	✓
getAvailableSeatNumberByTripScheduleId	/api/v1/reservation/ available-seat- number/{tripScheduleId}	GET	✗	✓
getAllOwnedTicketByLoggedInUser	/api/v1/reservation/passenger	GET	✗	✓
cancelTicket	/api/v1/reservation/{id}	DELETE	✗	✓

getOwnedTicketById	/api/v1/reservation/{id}	GET	✗	✓
--------------------	--------------------------	-----	---	---

Catatan: Controller ini khusus untuk role user, jika suatu akun hanya punya role admin, maka perlu membuat akun user untuk mengakses API yang ada di controller ini.

#### 4. Agency Controller

Name	Endpoint	Method	Admin	User
getAllAgency	/api/v1/agency	GET	✓	✓
createAgency	/api/v1/agency	POST	✓	✗
getAgencyById	/api/v1/agency/{id}	GET	✓	✓
deleteAgency	/api/v1/agency/{id}	DELETE	✓	✗
updateAgency	/api/v1/agency/{id}	PUT	✓	✗

#### 5. Bus Controller

Name	Endpoint	Method	Admin	User
getAllBus	/api/v1/bus	GET	✓	✓
createBus	/api/v1/bus	POST	✓	✗
getBusById	/api/v1/bus/{id}	GET	✓	✓
deleteBus	/api/v1/bus/{id}	DELETE	✓	✗
updateBus	/api/v1/bus/{id}	PUT	✓	✗

#### 6. Stop Controller

Name	Endpoint	Method	Admin	User
getAllStop	/api/v1/stop	GET	✓	✓
createStop	/api/v1/stop	POST	✓	✗
deleteStop	/api/v1/stop/{id}	DELETE	✓	✗
updateStop	/api/v1/stop/{id}	PUT	✓	✗

#### 7. Ticket Controller

Name	Endpoint	Method	Admin	User
getAllTicket	/api/v1/ticket	GET	✓	✗
getTicketById	/api/v1/ticket/{id}	GET	✓	✗

Catatan: Controller ini hanya untuk admin melihat ticket yang terpesan, namun admin tidak dapat memesan, mengubah, atau membatalkan ticket milik user.

#### 8. Trip Controller

Name	Endpoint	Method	Admin	User
getAllTrip	/api/v1/trip	GET	✓	✓
createTrip	/api/v1/trip	POST	✓	✗
getTripByAgencyId	/api/v1/trip/agency/{agencyId}	GET	✓	✓
getTripByBusId	/api/v1/trip/bus/{busId}	GET	✓	✓
getTripByDestStopId	/api/v1/trip/dest-stop/{destStopId}	GET	✓	✓
getTripBySourceStopId	/api/v1/trip/source-stop/{sourceStopId}	GET	✓	✓
deleteTrip	/api/v1/trip/{id}	DELETE	✓	✗
getTripById	/api/v1/trip/{id}	GET	✓	✓
updateTrip	/api/v1/trip/{id}	PUT	✓	✗

#### 9. Trip Schedule Controller

Name	Endpoint	Method	Admin	User
getAllTripSchedule	/api/v1/trip-schedule	GET	✓	✓
createTripSchedule	/api/v1/trip-schedule	POST	✓	✗
getTripScheduleByTripId	/api/v1/trip-schedule/trip/{tripId}	GET	✓	✓
deleteTripSchedule	/api/v1/trip-schedule/{id}	DELETE	✓	✗
getTripScheduleById	/api/v1/trip-schedule/{id}	GET	✓	✓
updateTripSchedule	/api/v1/trip-schedule/{id}	PUT	✓	✗

### D. Authentication Instructions

Berikut ini adalah langkah-langkah untuk autentikasi:

- Memiliki akun terlebih dahulu, jika belum punya akun, bisa membuat terlebih dahulu melalui endpoint /api/v1/user/signup, namun saya sudah menyediakan masing-masing 1 akun admin dan user yang bisa digunakan, namun bias juga jika ingin membuat akun baru.
- Melakukan autentikasi melalui endpoint /api/auth dan masukkan username dan password. Jika ingin menggunakan user yang sudah tersedia adalah sebagai berikut:
  - Admin  
username: admin123  
password: admin123
  - User  
username: raffiahmad  
password: password
- Jika berhasil, maka pada response code pada field token berupa "Bearer <kodetoken>" copy token tersebut
- Klik tombol authorize paste pada isian value, lalu klik Authorize

## E. API Details

### 1. User Controller

#### 1) registerUser

Endpoint: /api/v1/user/signup

Method: POST

URL Param: -

Request Body Example:

```
{
  "email": "string",
  "firstName": "string",
  "lastName": "string",
  "mobileNumber": "string",
  "password": "string",
  "role": [
    "string"
  ],
  "username": "string"
}
```

Response:

Condition	Code	Status	Message
Berhasil	200	OK	User berhasil didaftarkan
Ada field yang tidak ada atau ada tapi bernilai 0/null/""	400	Bad Request	Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong
Username sudah terpakai	400	Bad Request	Username sudah terpakai
Email sudah terpakai	400	Bad Request	Email sudah terpakai
Ada role selain "user", "admin", "role_admin", "role_user" (case insensitive)	400	Bad Request	Role tidak dikenal

- 2) updateUser  
Endpoint: /api/v1/user/update  
Method: PUT  
URL Param: -

Request Body Example:

```
{
  "firstName": "string",
  "lastName": "string",
  "mobileNumber": "string"
}
```

Response:

Condition	Code	Status	Message
Berhasil	200	OK	Data user berhasil diupdate
Ada field yang tidak ada atau ada tapi bernilai 0/null/""	400	Bad Request	Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong
Belum authorize	401	Unauthorized	<unauthorized response>
Data user tidak ditemukan (kemungkinan kalau sudah berhasil authenticate, harusnya ada di database, kecuali diubah manual di database)	404	Not Found	Data user tidak ditemukan

- 3) updatePassword  
Endpoint: /api/v1/user/update/password  
Method: PUT  
URL Param: -

Request Body Example:

```
{
  "password": "string"
}
```

Response:

Condition	Code	Status	Message
Berhasil	200	OK	Password berhasil diubah
Ada field yang tidak ada atau ada tapi bernilai 0/null/""	400	Bad Request	Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong
Belum authorize	401	Unauthorized	<unauthorized response>
Data user tidak ditemukan (kemungkinan kalau sudah berhasil authenticate, seharusnya ada di database, kecuali diubah manual di database)	404	Not Found	Data user tidak ditemukan

## 2. Auth Controller

### 1) authenticateUser

Endpoint: /api/auth

Method: POST

URL Param: -

Request Body Example:

```
{
  "password": "string",
  "username": "string"
}
```

Response:

Condition	Code	Status	Message
Berhasil	200	OK	<jwt response >
Authentication gagal (username kosong/salah atau password kosong/salah)	401	Unauthorized	<unauthorized response>

## 3. Reservation Controller

### 1) bookTicket

Endpoint: /api/v1/reservation

Method: POST

URL Param: -

Request Body Example:

```
{  
  "cancellable": true,  
  "seatNumber": 0,  
  "tripScheduleId": 0  
}
```

Response:

Condition	Code	Status	Message
Berhasil	200	OK	Ticket berhasil dipesan
Ada field yang tidak ada atau ada tapi bernilai 0/null/""	400	Bad Request	Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong
Field tripScheduleId tidak valid (tidak ada di database)	400	Bad Request	Data trip schedule tidak ditemukan
Field tripScheduleId merujuk pada trip schedule yang sudah berlalu	400	Bad Request	Trip schedule sudah lewat, silahkan pilih trip schedule yang lain
Data passenger tidak ditemukan (kemungkinan kalau sudah berhasil authenticate, seharusnya ada di database, kecuali diubah manual di database)	400	Bad Request	Data passenger tidak ditemukan
Kolom available_seats untuk tripScheduleId tersebut 0 (ticket sudah terpesan semua sesuai kapasitas bus)	400	Bad Request	Mohon maaf, ticket sudah habis
User sudah memesan ticket untuk tripScheduleId tersebut	400	Bad Request	Mohon maaf, anda tidak dapat memesan lagi untuk trip schedule ini
seatNumber > kapasitas bus, sehingga tidak valid	400	Bad Request	Seat number tidak valid. Kapasitas bus hanya <capacity> kursi
seatNumber yang dipilih sudah dipesan oleh user lain	400	Bad Request	Seat number sudah dipesan, silahkan pilih seat number lain
Belum authorize	401	Unauthorized	<unauthorized response>
Menggunakan akun admin	403	Forbidden	<forbidden response>

- 2) `getAvailableSeatNumberByTripScheduleId`  
 Endpoint: `/api/v1/reservation/available-seat-number/{tripScheduleId}`  
 Method: GET  
 URL Param: `tripScheduleId`  
 Request Body Example: -

Response:

Condition	Code	Status	Message
Berhasil	200	OK	<list of available seat number by that trip schedule id>
Belum authorize	401	Unauthorized	<unauthorized response>
Menggunakan akun admin	403	Forbidden	<forbidden response>
<code>tripScheduleId</code> tidak valid (tidak ada di database)	404	Not Found	Data trip schedule tidak ditemukan

- 3) `getAllOwnedTicketByLoggedInUser`  
 Endpoint: `/api/v1/reservation/passenger`  
 Method: GET  
 URL Param: -  
 Request Body Example: -

Response:

Condition	Code	Status	Message
Berhasil	200	OK	<list of owned ticket by logged in user>
Belum authorize	401	Unauthorized	<unauthorized response>
Menggunakan akun admin	403	Forbidden	<forbidden response>
User belum memiliki ticket	404	Not Found	Anda belum memiliki ticket yang tersimpan

- 4) `cancelTicket`  
 Endpoint: `/api/v1/reservation/{id}`  
 Method: DELETE  
 URL Param: `id`  
 Request Body Example: -



Response:

Condition	Code	Status	Message
Berhasil	200	OK	Ticket berhasil didaftarkan
Id yang dimasukkan valid, namun bukan milik user yang login	400	Bad Request	Ticket hanya dapat dibatalkan oleh pemilik ticket
Kolom cancellable false, sehingga tidak dapat dibatalkan	400	Bad Request	Mohon maaf, ticket tidak dapat dibatalkan
Belum authorize	401	Unauthorized	<unauthorized response>
Menggunakan akun admin	403	Forbidden	<forbidden response>
Id tidak valid (tidak ada di database)	404	Not Found	Data ticket tidak ditemukan

5) getTicketById

Endpoint: /api/v1/reservation/{id}

Method: GET

URL Param: -

Request Body Example: -

Response:

Condition	Code	Status	Message
Berhasil	200	OK	<ticket response>
Id ticket adalah milik user lain, sehingga tidak dapat melihat ticket tersebut	400	Bad Request	Anda tidak dapat melihat ticket milik penumpang lain
Belum authorize	401	Unauthorized	<unauthorized response>
Menggunakan akun admin	403	Forbidden	<forbidden response>
Id tidak valid (tidak ada di database)	404	Not Found	Data ticket tidak ditemukan

4. Agency Controller

1) getAllAgency

Endpoint: /api/v1/agency

Method: GET

URL Param: -

Request Body Example: -

Response:

Condition	Code	Status	Message
Berhasil	200	OK	<list of agency>
Belum authorize	401	Unauthorized	<unauthorized response>
Tidak ada data agency di database	404	Not Found	Tidak ada agency yang tersimpan

- 2) createAgency  
Endpoint: /api/v1/agency  
Method: POST  
URL Param: -  
Request Body Example:
- ```
{  
  "code": "string",  
  "details": "string",  
  "name": "string",  
  "ownerId": 0  
}
```

Response

| Condition                                                 | Code | Status       | Message                                                                       |
|-----------------------------------------------------------|------|--------------|-------------------------------------------------------------------------------|
| Berhasil                                                  | 200  | OK           | Agency berhasil disimpan                                                      |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/"" | 400  | Bad Request  | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| ownerId tidak valid (tidak ada di database)               | 400  | Bad request  | Data owner tidak ditemukan                                                    |
| Belum authorize                                           | 401  | Unauthorized | <unauthorized response>                                                       |
| Menggunakan akun user                                     | 403  | Forbidden    | <forbidden response>                                                          |

- 3) getAgencyByOwnerId  
Endpoint: /api/v1/agency/owner/{ownerId}  
Method: GET  
URL Param: ownerId  
Request Body Example: -

Response:

| Condition                                   | Code | Status       | Message                     |
|---------------------------------------------|------|--------------|-----------------------------|
| Berhasil                                    | 200  | OK           | <list of agency by ownerId> |
| Belum authorize                             | 401  | Unauthorized | <unauthorized response>     |
| ownerId tidak valid (tidak ada di database) | 404  | Not Found    | Data owner tidak ditemukan  |
| Owner tidak memiliki agency                 | 404  | Not Found    | User tidak memiliki agency  |

4) deleteAgency

Endpoint: /api/v1/agency/{id}

Method: DELETE

URL Param: id

Request Body Example: -

Response:

| Condition                              | Code | Status       | Message                     |
|----------------------------------------|------|--------------|-----------------------------|
| Berhasil                               | 200  | OK           | Agency berhasil dihapus     |
| Belum authorize                        | 401  | Unauthorized | <unauthorized response>     |
| Menggunakan akun user                  | 403  | Forbidden    | <forbidden response>        |
| Id tidak valid (tidak ada di database) | 404  | Not Found    | Data agency tidak ditemukan |

5) getAgencyById

Endpoint: /api/v1/agency/{id}

Method: GET

URL Param: id

Request Body Example: -

Response

| Condition                              | Code | Status       | Message                     |
|----------------------------------------|------|--------------|-----------------------------|
| Berhasil                               | 200  | OK           | <agency response>           |
| Belum authorize                        | 401  | Unauthorized | <unauthorized response>     |
| Id tidak valid (tidak ada di database) | 404  | Not Found    | Data agency tidak ditemukan |

6) updateAgency

Endpoint: /api/v1/agency/{id}

Method: PUT

URL Param: id

Request Body Example:

```
{
  "code": "string",
  "details": "string",
  "name": "string",
  "ownerId": 0
}
```

Response:

| Condition                                                 | Code | Status       | Message                                                                       |
|-----------------------------------------------------------|------|--------------|-------------------------------------------------------------------------------|
| Berhasil                                                  | 200  | OK           | Agency berhasil diubah                                                        |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/"" | 400  | Bad request  | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| ownerId tidak valid (tidak ada di database)               | 400  | Bad request  | Data owner tidak ditemukan                                                    |
| Belum authorize                                           | 401  | Unauthorized | <unauthorized response>                                                       |
| Menggunakan akun user                                     | 403  | Forbidden    | <forbidden response>                                                          |
| Id tidak valid (tidak ada di database)                    | 404  | Not Found    | Data agency tidak ditemukan                                                   |

## 5. Bus Controller

### 1) getAllBus

Endpoint: /api/v1/bus

Method: GET

URL Param: -

Request Body Example: -

Response:

| Condition                      | Code | Status       | Message                      |
|--------------------------------|------|--------------|------------------------------|
| Berhasil                       | 200  | OK           | <list of bus>                |
| Belum authorize                | 401  | Unauthorized | <unauthorized response>      |
| Tidak ada data bus di database | 404  | Not Found    | Tidak ada bus yang tersimpan |

### 2) createBus

Endpoint: /api/v1/bus

Method: POST

URL Param: -

Request Body Example:

```
{
  "agencyId": 0,
  "capacity": 0,
  "code": "string",
  "make": "string"
}
```

Response:

| Condition                                                 | Code | Status       | Message                                                                       |
|-----------------------------------------------------------|------|--------------|-------------------------------------------------------------------------------|
| Berhasil                                                  | 200  | OK           | Bus berhasil disimpan                                                         |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/"" | 400  | Bad Request  | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| agencyId tidak valid (tidak ada di database)              | 400  | Bad Request  | Data agency tidak ditemukan                                                   |
| Belum authorize                                           | 401  | Unauthorized | <unauthorized response>                                                       |
| Menggunakan akun user                                     | 403  | Forbidden    | <forbidden response>                                                          |

- 3) getBusByAgencyId  
Endpoint: /api/v1/bus/agency/{agencyId}  
Method: GET  
URL Param: agencyId  
Request Body Example: -

Response:

| Condition                                    | Code | Status       | Message                     |
|----------------------------------------------|------|--------------|-----------------------------|
| Berhasil                                     | 200  | OK           | <list of bus by agency id>  |
| Belum authorize                              | 401  | Unauthorized | <unauthorized response>     |
| agencyId tidak valid (tidak ada di database) | 404  | Not Found    | Data agency tidak ditemukan |
| Agency tidak memiliki bus                    | 404  | Not Found    | Agency tidak memiliki bus   |

- 4) deleteBus  
Endpoint: /api/v1/bus/{id}  
Method: DELETE  
URL Param: id  
Request Body Example: -

Response:

| Condition                              | Code | Status       | Message                  |
|----------------------------------------|------|--------------|--------------------------|
| Berhasil                               | 200  | OK           | Bus berhasil dihapus     |
| Belum authorize                        | 401  | Unauthorized | <unauthorized response>  |
| Menggunakan akun user                  | 403  | Forbidden    | <forbidden response>     |
| Id tidak valid (tidak ada di database) | 404  | Not Found    | Data bus tidak ditemukan |

- 5) getBusById  
Endpoint: /api/v1/bus/{id}  
Method: GET  
URL Param: id  
Request Body Example: -

Response:

| Condition                                 | Code | Status       | Message                  |
|-------------------------------------------|------|--------------|--------------------------|
| Berhasil                                  | 200  | OK           | <bus response>           |
| Belum authorize                           | 401  | Unauthorized | <unauthorized response>  |
| Id tidak valid (tidak ada dalam database) | 404  | Not Found    | Data bus tidak ditemukan |

- 6) updateBus  
Endpoint: /api/v1/bus/{id}  
Method: PUT  
URL Param: id  
Request Body Example:  
{  
  "agencyId": 0,  
  "capacity": 0,  
  "code": "string",  
  "make": "string"  
}

Response:

| Condition                                                 | Code | Status       | Message                                                                       |
|-----------------------------------------------------------|------|--------------|-------------------------------------------------------------------------------|
| Berhasil                                                  | 200  | OK           | Bus berhasil diubah                                                           |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/"" | 400  | Bad Request  | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| agencyId tidak valid (tidak ada di database)              | 400  | Bad Request  | Data agency tidak ditemukan                                                   |
| Belum authorize                                           | 401  | Unauthorized | <unauthorized response>                                                       |
| Menggunakan akun user                                     | 403  | Forbidden    | <forbidden response>                                                          |

|                                        |     |           |                          |
|----------------------------------------|-----|-----------|--------------------------|
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data bus tidak ditemukan |
|----------------------------------------|-----|-----------|--------------------------|

## 6. Stop Controller

### 1) getAllStop

Endpoint: /api/v1/stop

Method: GET

URL Param: -

Request Body Example: -

Response:

| Condition                       | Code | Status       | Message                       |
|---------------------------------|------|--------------|-------------------------------|
| Berhasil                        | 200  | OK           | <list of all stop>            |
| Belum authorize                 | 401  | Unauthorized | <unauthorized response>       |
| Tidak ada data stop di database | 404  | Not Found    | Tidak ada stop yang tersimpan |

### 2) createStop

Endpoint: /api/v1/stop

Method: GET

URL Param: -

Request Body Example:

```
{
  "code": "string",
  "detail": "string",
  "name": "string"
}
```

Response:

| Condition                                                 | Code | Status       | Message                                                                       |
|-----------------------------------------------------------|------|--------------|-------------------------------------------------------------------------------|
| Berhasil                                                  | 200  | OK           | Stop berhasil disimpan                                                        |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/"" | 400  | Bad Request  | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| Belum authorize                                           | 401  | Unauthorized | <unauthorized response>                                                       |
| Menggunakan akun user                                     | 403  | Forbidden    | <forbidden response>                                                          |

### 3) deleteStop

Endpoint: /api/v1/stop/{id}

Method: DELETE

URL Param: id

Request Body Example: -

Response:

| Condition                              | Code | Status       | Message                   |
|----------------------------------------|------|--------------|---------------------------|
| Berhasil                               | 200  | OK           | Stop berhasil dihapus     |
| Belum authorize                        | 401  | Unauthorized | <unauthorized response>   |
| Menggunakan akun user                  | 403  | Forbidden    | <forbidden response>      |
| Id tidak valid (tidak ada di database) | 404  | Not Found    | Data stop tidak ditemukan |

4) getStopById

Endpoint: /api/v1/stop/{id}

Method: GET

URL Param: id

Request Body Example: -

Response:

| Condition                              | Code | Status       | Message                   |
|----------------------------------------|------|--------------|---------------------------|
| Berhasil                               | 200  | OK           | <stop response>           |
| Belum authorize                        | 401  | Unauthorized | <unauthorized response>   |
| Id tidak valid (tidak ada di database) | 404  | Not Found    | Data stop tidak ditemukan |

5) updateStop

Endpoint: /api/v1/stop/{id}

Method: PUT

URL Param: id

Request Body Example:

```
{
  "code": "string",
  "detail": "string",
  "name": "string"
}
```

Response:

| Condition                                                 | Code | Status       | Message                                                                       |
|-----------------------------------------------------------|------|--------------|-------------------------------------------------------------------------------|
| Berhasil                                                  | 200  | OK           | Stop berhasil diubah                                                          |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/"" | 400  | Bad Request  | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| Belum authorize                                           | 401  | Unauthorized | <unauthorized response>                                                       |
| Menggunakan akun user                                     | 403  | Forbidden    | <forbidden response>                                                          |
| Id tidak valid (tidak ada di database)                    | 404  | Not Found    | Data stop tidak ditemukan                                                     |



## 7. Ticket Controller

### 1) getAllTicket

Endpoint: /api/v1/ticket

Method: GET

URL Param: -

Request Body Example: -

Response:

| Condition                         | Code | Status       | Message                         |
|-----------------------------------|------|--------------|---------------------------------|
| Berhasil                          | 200  | OK           | <list of all ticket>            |
| Belum authorize                   | 401  | Unauthorized | <unauthorized response>         |
| Menggunakan akun user             | 403  | Forbidden    | <forbidden response>            |
| Tidak ada data ticket di database | 404  | Not Found    | Tidak ada ticket yang tersimpan |

### 2) getAllTicketByTripScheduleId

Endpoint: /api/v1/ticket/trip-schedule/{tripScheduleId}

Method: GET

URL Param: tripScheduleId

Request Body Example: -

Response:

| Condition                                                 | Code | Status       | Message                                                   |
|-----------------------------------------------------------|------|--------------|-----------------------------------------------------------|
| Berhasil                                                  | 200  | OK           | <list of ticket by trip schedule id>                      |
| Belum authorize                                           | 401  | Unauthorized | <unauthorized response>                                   |
| Menggunakan akun user                                     | 403  | Forbidden    | <forbidden response>                                      |
| tripScheduleId tidak valid (tidak ada di database)        | 404  | Not Found    | Data trip schedule tidak ditemukan                        |
| Tidak ada tiket yang terjual untuk trip schedule tersebut | 404  | Not Found    | Tidak ada ticket yang dipesan dari trip schedule tersebut |

### 3) getTicketById

Endpoint: /api/v1/ticket/{id}

Method: GET

URL Param: id

Request Body Example: -

Response:

| Condition                              | Code | Status      | Message                     |
|----------------------------------------|------|-------------|-----------------------------|
| Berhasil                               | 200  | OK          | <ticket response>           |
| Belum authorize                        | 401  | Unauthorize | <unauthorized response>     |
| Menggunakan akun user                  | 403  | Forbidden   | <forbidden response>        |
| Id tidak valid (tidak ada di database) | 404  | Not Found   | Data ticket tidak ditemukan |

## 8. Trip Controller

### 1) getAllTrip

Endpoint: /api/v1/trip

Method: GET

URL Param: -

Request Body Example: -

Response:

| Condition                       | Code | Status       | Message                       |
|---------------------------------|------|--------------|-------------------------------|
| Berhasil                        | 200  | OK           | <list of all trip>            |
| Belum authorize                 | 401  | Unauthorized | <unauthorized response>       |
| Tidak ada data trip di database | 404  | Not Found    | Tidak ada trip yang tersimpan |

### 2) createTrip

Endpoint: /api/v1/trip

Method: POST

URL Param: -

Request Body Example:

```
{
  "busId": 0,
  "destStopId": 0,
  "fare": 0,
  "journeyTime": 0,
  "sourceStopId": 0
}
```

Response:

| Condition                                                 | Code | Status      | Message                                                                       |
|-----------------------------------------------------------|------|-------------|-------------------------------------------------------------------------------|
| Berhasil                                                  | 200  | OK          | Trip berhasil disimpan                                                        |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/"" | 400  | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| busId tidak valid (tidak ada di database)                 | 400  | Bad Request | Data bus tidak ditemukan                                                      |

|                                                  |     |              |                                                    |
|--------------------------------------------------|-----|--------------|----------------------------------------------------|
| Bus tidak memiliki kolom agencyId yang valid     | 400 | Bad Request  | Data agency tidak ditemukan                        |
| sourceStopId tidak valid (tidak ada di database) | 400 | Bad Request  | Data destination stop tidak ditemukan              |
| destStopId tidak valid (tidak ada di database)   | 400 | Bad Request  | Data source stop tidak ditemukan                   |
| Nilai sourceStopId sama dengan destStopId        | 400 | Bad Request  | Field sourceStopId dan destStopId tidak boleh sama |
| Belum authorize                                  | 401 | Unauthorized | <unauthorized response>                            |
| Menggunakan akun user                            | 403 | Forbidden    | <forbidden response>                               |

- 3) **getTripByAgencyId**  
 Endpoint: /api/v1/trip/agency/{agencyId}  
 Method: GET  
 URL Param: agencyId  
 Request Body Example: -

Response:

| Condition                                    | Code | Status       | Message                     |
|----------------------------------------------|------|--------------|-----------------------------|
| Berhasil                                     | 200  | OK           | <list of trip by agencyId>  |
| Belum authorize                              | 401  | Unauthorized | <unauthorized response>     |
| Tidak ada trip dari agency tersebut          | 404  | Not Found    | Agency tidak memiliki trip  |
| agencyId tidak valid (tidak ada di database) | 404  | Not Found    | Data agency tidak ditemukan |

- 4) **getTripByBusId**  
 Endpoint: /api/v1/trip/bus/{busId}  
 Method: GET  
 URL Param: busId  
 Request Body Example: -

Response:

| Condition                                      | Code | Status       | Message                  |
|------------------------------------------------|------|--------------|--------------------------|
| Berhasil                                       | 200  | OK           | <list of trip by busId>  |
| Belum authorize                                | 401  | Unauthorized | <unauthorized response>  |
| Tidak ada trip dengan menggunakan bus tersebut | 404  | Not Found    | Bus tidak memiliki trip  |
| busId tidak valid (tidak ada di database)      | 404  | Not Found    | Data bus tidak ditemukan |

- 5) getTripByDestStopId  
Endpoint: /api/v1/trip/dest-stop/{destStopId}  
Method: GET  
URL Param: destStopId  
Request Body Example: -

Response:

| Condition                                      | Code | Status       | Message                                        |
|------------------------------------------------|------|--------------|------------------------------------------------|
| Berhasil                                       | 200  | OK           | <list of trip by destStopId>                   |
| Belum authorize                                | 401  | Unauthorized | <unauthorized response>                        |
| Tidak ada trip yang bertujuan ke stop tersebut | 404  | Not Found    | Tidak ada trip yang bertujuan ke stop tersebut |
| destStopId tidak valid (tidak ada di database) | 404  | Not Found    | Data stop tidak ditemukan                      |

- 6) getTripBySourceStopId  
Endpoint: /api/v1/trip/source-stop/{sourceStopId}  
Method: GET  
URL Param: sourceStopId  
Request Body Example: -

Response:

| Condition                                        | Code | Status       | Message                                          |
|--------------------------------------------------|------|--------------|--------------------------------------------------|
| Berhasil                                         | 200  | OK           | <list of trip by sourceStopId>                   |
| Belum authorize                                  | 401  | Unauthorized | <unauthorized response>                          |
| Tidak ada trip yang berangkat dari stop tersebut | 404  | Not Found    | Tidak ada trip yang berangkat dari stop tersebut |
| destStopId tidak valid (tidak ada di database)   | 404  | Not Found    | Data stop tidak ditemukan                        |

- 7) deleteTrip  
Endpoint: /api/v1/trip/{id}  
Method: DELETE  
URL Param: id  
Request Body Example: -

Response:

| Condition                              | Code | Status       | Message                   |
|----------------------------------------|------|--------------|---------------------------|
| Berhasil                               | 200  | OK           | Trip berhasil dihapus     |
| Belum authorize                        | 401  | Unauthorized | <unauthorized response>   |
| Menggunakan akun user                  | 403  | Forbidden    | <forbidden response>      |
| Id tidak valid (tidak ada di database) | 404  | Not Found    | Data trip tidak ditemukan |

- 8) getTripById  
Endpoint: /api/v1/trip/{id}  
Method: GET  
URL Param: id  
Request Body Example: -

Response:

| Condition                              | Code | Status       | Message                   |
|----------------------------------------|------|--------------|---------------------------|
| Berhasil                               | 200  | OK           | <trip response>           |
| Belum authorize                        | 401  | Unauthorized | <unauthorized response>   |
| Id tidak valid (tidak ada di database) | 404  | Not Found    | Data trip tidak ditemukan |

- 9) updateTrip  
Endpoint: /api/v1/trip/{id}  
Method: PUT  
URL Param: id  
Request Body Example:

```
{  
  "busId": 0,  
  "destStopId": 0,  
  "fare": 0,  
  "journeyTime": 0,  
  "sourceStopId": 0  
}
```

Response:

| Condition                                                 | Code | Status      | Message                                                                       |
|-----------------------------------------------------------|------|-------------|-------------------------------------------------------------------------------|
| Berhasil                                                  | 200  | OK          | Trip berhasil diubah                                                          |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/"" | 400  | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| busId tidak valid (tidak ada di database)                 | 400  | Bad Request | Data bus tidak ditemukan                                                      |
| Bus tidak memiliki kolom agencyId yang valid              | 400  | Bad Request | Data agency tidak ditemukan                                                   |

|                                                  |     |              |                                                    |
|--------------------------------------------------|-----|--------------|----------------------------------------------------|
| sourceStopId tidak valid (tidak ada di database) | 400 | Bad Request  | Data destination stop tidak ditemukan              |
| destStopId tidak valid (tidak ada di database)   | 400 | Bad Request  | Data source stop tidak ditemukan                   |
| Nilai sourceStopId sama dengan destStopId        | 400 | Bad Request  | Field sourceStopId dan destStopId tidak boleh sama |
| Belum authorize                                  | 401 | Unauthorized | <unauthorized response>                            |
| Menggunakan akun user                            | 403 | Forbidden    | <forbidden response>                               |
| Id tidak valid (tidak ada di database)           | 404 | Not Found    | Data trip tidak ditemukan                          |

## 9. Trip Schedule Controller

### 1) getAllTripSchedule

Endpoint: /api/v1/trip-schedule

Method: GET

URL Param: -

Request Body Example: -

Response:

| Condition                                | Code | Status       | Message                                |
|------------------------------------------|------|--------------|----------------------------------------|
| Berhasil                                 | 200  | OK           | <list of all trip schedule>            |
| Belum authorize                          | 401  | Unauthorized | <unauthorized response>                |
| Tidak ada data trip schedule di database | 404  | Not Found    | Tidak ada trip schedule yang tersimpan |

### 2) createTripSchedule

Endpoint: /api/v1/trip-schedule

Method: POST

URL Param: -

Request Body Example:

```
{
  "tripDate": "dd-MM-yyyy",
  "tripDetailId": 0
}
```

Response:

| Condition                                                 | Code | Status      | Message                                                                       |
|-----------------------------------------------------------|------|-------------|-------------------------------------------------------------------------------|
| Berhasil                                                  | 200  | OK          | Trip schedule berhasil disimpan                                               |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/"" | 400  | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| tripDetailId tidak valid (tidak ada di database)          | 400  | Bad Request | Data trip detail tidak ditemukan                                              |

|                                                                  |     |              |                                                              |
|------------------------------------------------------------------|-----|--------------|--------------------------------------------------------------|
| Trip date sudah lewat, sehingga tidak bisa membuat trip schedule | 400 | Bad Request  | Trip date sudah lewat, silahkan masukkan trip date yang lain |
| Belum authorize                                                  | 401 | Unauthorized | <unauthorized response>                                      |
| Menggunakan akun user                                            | 403 | Forbidden    | <forbidden response>                                         |

- 3) `getTripScheduleByTripId`  
 Endpoint: `/api/v1/trip-schedule/trip/{tripId}`  
 Method: GET  
 URL Param: `tripId`  
 Request Body Example: -

Response:

| Condition                                  | Code | Status       | Message                                    |
|--------------------------------------------|------|--------------|--------------------------------------------|
| Berhasil                                   | 200  | OK           | <list trip schedule by trip id>            |
| Belum authorize                            | 401  | Unauthorized | <unauthorized response>                    |
| tripId tidak valid (tidak ada di database) | 404  | Not Found    | Data trip tidak ditemukan                  |
| Trip tersebut tidak memiliki trip schedule | 404  | Not Found    | Tidak ada trip schedule dari trip tersebut |

- 4) `deleteTripSchedule`  
 Endpoint: `/api/v1/trip-schedule/{id}`  
 Method: DELETE  
 URL Param: `id`  
 Request Body Example: -

Response:

| Condition                              | Code | Status       | Message                            |
|----------------------------------------|------|--------------|------------------------------------|
| Berhasil                               | 200  | OK           | Trip schedule berhasil dihapus     |
| Belum authorize                        | 401  | Unauthorized | <unauthorized response>            |
| Menggunakan akun user                  | 403  | Forbidden    | <forbidden response>               |
| Id tidak valid (tidak ada di database) | 404  | Not Found    | Data trip schedule tidak ditemukan |

- 5) `getTripScheduleById`  
 Endpoint: `/api/v1/trip-schedule/{id}`  
 Method: GET  
 URL Param: `id`  
 Request Body Example: -

Response:

| Condition                              | Code | Status       | Message                            |
|----------------------------------------|------|--------------|------------------------------------|
| Berhasil                               | 200  | OK           | <trip schedule response>           |
| Belum authorize                        | 401  | Unauthorized | <unauthorized response>            |
| Id tidak valid (tidak ada di database) | 404  | Not Found    | Data trip schedule tidak ditemukan |

6) updateTripSchedule

Endpoint: /api/v1/trip-schedule/{id}

Method: PUT

URL Param: id

Request Body Example:

```
{
  "tripDate": "dd-MM-yyyy",
  "tripDetailId": 0
}
```

Response:

| Condition                                                           | Code | Status       | Message                                                                       |
|---------------------------------------------------------------------|------|--------------|-------------------------------------------------------------------------------|
| Berhasil                                                            | 200  | OK           | Trip schedule berhasil diubah                                                 |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/""           | 400  | Bad Request  | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| tripDetailId tidak valid (tidak ada di database)                    | 400  | Bad Request  | Data trip detail tidak ditemukan                                              |
| Trip date sudah terlewat, sehingga tidak bisa membuat trip schedule | 400  | Bad Request  | Trip date sudah lewat, silahkan masukkan trip date yang lain                  |
| Belum authorize                                                     | 401  | Unauthorized | <unauthorized response>                                                       |
| Menggunakan akun user                                               | 403  | Forbidden    | <forbidden response>                                                          |
| Id tidak valid (tidak ada di database)                              | 404  | Not Found    | Data trip schedule tidak ditemukan                                            |