



## SOAL LOMBA

### Isi

Modul ini terdiri dari beberapa file:

- MODULE\_SERVER.pd
- MODULE\_SERVER\_MEDIA.zip

### Pendahuluan

Transjakarta merupakan suatu perusahaan layanan transportasi umum yang menggunakan armada bus untuk beroperasi. Transjakarta menggunakan sistem koridor untuk membagi rute mereka agar menjangkau seluruh kota. Anda sebagai developer diminta membuat sebuah sistem untuk seorang manajer manajemen bus dan driver ditempatkan di suatu koridor di transjakarta secara efisien agar tidak terjadi konflik penempatan bus dan driver dalam satu waktu.

Anda diminta untuk membuat backend API menggunakan framework Laravel PHP dan frontend yang interaktif dengan menggunakan Javascript untuk mengimplementasikan API tersebut. Detail tugas akan dijelaskan setelah bagian ini.

### Detail Tugas dan Project

#### API List:

Gunakan ERD yang telah disediakan.

Buatlah akun dummy pada table users (password di-hash menggunakan **bcrypt**):

username	password
admin	passadmin

Berikut ini merupakan web service endpoint yang dibutuhkan:

#### 1. Authentication

##### a. Login

URL: [domain]/v1/auth/login



Description: Untuk Admin generate dan mendapatkan login token menggunakan username dan password. Username dan password harus valid.

Method: POST

Request Parameter:

- Body:
  - Username
  - Password

Response:

- If login success:
  - Header: response status: 200
  - Body: { token }
- If login failed (username atau password tidak cocok atau kosong):
  - Header: response status: 401
  - Body: { message: invalid login }

#### b. Logout

URL: [domain]/v1/auth/logout

Description: Untuk server membuat token menjadi Invalid.

Method: GET

Header:

Authorization: Bearer {TOKEN}

Response:

- If logout success:
  - Header: response status: 200



Body: { *message: logout success* }

- If logout failed (token invalid):

Header: response status: 403

Body: { *message: forbidden* }

## 2. Bus

### a. Create new bus

URL: [domain]/v1/bus

Description: Untuk admin membuat armada baru.

Method: POST

Header:

Authorization: Bearer {*TOKEN*}

Request Parameter:

- Body:
  - o **plate\_number** (*string; required*)
  - o **brand** (*string; required*)
  - o **type** (*small/big/articulated; required*)
  - o **fuel** (*petrol/gas/diesel/electric; required*)

Response:

- If success:
  - Header: response status: 200
  - Body: { *message: create bus success* }
- If input validation failed:
  - Header: response status: 422
  - Body: { *message: invalid field* }
- If accessed without token:



Header: response status: 403

Body: { *message: forbidden* }

**b. Update bus**

URL: [domain]/v1/bus/{*id*}

Description: Untuk admin memperbaharui armada yang ada berdasarkan *id*.

Method: PUT

Header:

Authorization: Bearer {*TOKEN*}

Request Parameter:

- Body:
  - o **plate\_number** (*string*)
  - o **brand** (*string*)
  - o **type** (*small/big/articulated*)
  - o **fuel** (*petrol/gas/diesel/electric*)

Response:

- If success:

Header: response status: 200

Body: { *message: update bus success* }
- If input validation failed:

Header: response status: 422

Body: { *message: invalid field* }
- If accessed without token:

Header: response status: 403

Body: { *message: forbidden* }



**c. Delete bus**

URL: [domain]/v1/bus/{id}

Description: Untuk admin menghapus armada yang ada berdasarkan id.

Method: DELETE

Header:

Authorization: Bearer {TOKEN}

Response:

- If success:

Header: response status: 200

Body: { message: delete bus success }

- If accessed without token:

Header: response status: 403

Body: { message: forbidden }

**d. Get all bus**

URL: [domain]/v1/bus

Description: Untuk admin mendapatkan semua data armada.

Method: GET

Header:

Authorization: Bearer {TOKEN}

Response:

- If success:

Header: response status: 200

Body: all bus data in json { id, plate\_number, brand, type, fuel }



- If accessed without token:

Header: response status: 403

Body: { *message: forbidden* }

### 3. Driver

#### a. Create new driver

URL: [domain]/v1/driver

Description: Untuk admin menambahkan supir baru.

Method: POST

Header:

Authorization: Bearer {*TOKEN*}

Request Parameter:

- Body:
  - o *driver\_id* (string; unique; required)
  - o *name* (string; required)
  - o *age* (integer; min:18; required)
  - o *gender* (male/female; required)
  - o *phone\_number* (string; size:20; required)
  - o *address* (string; required)
  - o *email* (email; required)

Response:

- If success:

Header: response status: 200

Body: { *message: create driver success* }
- If input validation failed:

Header: response status: 422





Body: { message: invalid field }

- If accessed without token:

Header: response status: 403

Body: { message: forbidden }

#### b. Update driver

URL: [domain]/v1/driver/{**driver\_id**}

Description: Untuk admin memperbarui supir berdasarkan **driver\_id**.

Method: PUT

Header:

Authorization: Bearer {**TOKEN**}

Request Parameter:

- Body:
  - o **driver\_id** (string)
  - o **name** (string)
  - o **age** (integer; min:18)
  - o **gender** (male/female)
  - o **phone\_number** (string; size:20)
  - o **address** (string)
  - o **email** (email)

Response:

- If success:

Header: response status: 200

Body: { message: update driver success }

- If input validation failed:

Header: response status: 422



Body: { *message: invalid field* }

- If accessed without token:

Header: response status: 403

Body: { *message: forbidden* }

**c. Delete driver**

URL: [domain]/v1/driver/{**driver\_id**}

Description: Untuk admin menghapus supir berdasarkan **driver\_id**.

Method: DELETE

Header:

Authorization: Bearer {*TOKEN*}

Response:

- If success:

Header: response status: 200

Body: { *message: delete driver success* }

- If accessed without token:

Header: response status: 403

Body: { *message: forbidden* }

**d. Get all driver**

URL: [domain]/v1/driver

Description: Untuk admin mendapatkan semua data supir.

Method: GET

Header:

Authorization: Bearer {*TOKEN*}





Response:

- If success:

Header: response status: 200

Body: all driver data in json { ***driver\_id, name, age, gender, phone\_number, address, email*** }

- If accessed without token:

Header: response status: 403

Body: { *message: forbidden* }

#### 4. Corridor

##### a. Create new corridor

URL: [domain]/v1/corridor

Description: Untuk admin menambahkan data koridor. Dalam penambahan data, harus dipastikan bahwa bus dan driver yang diinput tersedia di tanggal yang sama, sehingga tidak terjadi konflik (**tidak terdapat lebih dari satu koridor untuk bus atau driver di tanggal yang sama**). Misal Bus A dan Driver B sudah ditugaskan untuk di koridor 9, maka Bus A atau Driver B tidak bisa ditugaskan lagi di koridor lain dalam tanggal yang sama. Kemudian untuk waktu dinas (**duty\_time\_in\_minutes**) dihitung berdasarkan waktu mulai dinas dan waktu selesai dinas.

Method: POST

Header:

Authorization: Bearer {*TOKEN*}

Request Parameter:

- Body:
  - o **corridor\_code** (*string; required*)
  - o **driver\_id** (*string; required*)



- o **bus\_id** (*integer; required*)
- o **duty\_date** (*date; required*)
- o **start\_time** (*time; required; format:H:i*)
- o **finish\_time** (*time; required; format:H:i*)

Response:

- If success:

Header: response status: 200

Body: corridor data in json with relationship of **bus** and **driver** { **id**, **corridor\_code**, **driver\_id**, **driver\_name**, **bus\_id**, **bus\_plate\_number**, **bus\_brand**, **duty\_date**, **start\_time**, **finish\_time**, **duty\_time\_in\_minute** }

- If input validation failed:

Header: response status: 422

Body: { *message: invalid field* }

- If corridor cannot be fulfilled because conflicting corridor or no available bus or driver:

Header: response status: 401

Body: { *message: conflicting corridor or bus or driver is not available* }

- If accessed without token:

Header: response status: 403

Body: { *message: forbidden* }

#### b. Delete corridor

URL: [domain]/v1/corridor/{**id**}

Description: Untuk admin menghapus koridor berdasarkan id.

Method: DELETE



Header:

Authorization: Bearer {TOKEN}

Response:

- If success:

Header: response status: 200

Body: { *message: delete corridor success* }

- If accessed without token:

Header: response status: 403

Body: { *message: forbidden* }

#### c. Get all corridors

URL: [domain]/v1/corridor

Description: Untuk admin mendapatkan semua data koridor.

Method: GET

Header:

Authorization: Bearer {TOKEN}

Response:

- If success:

Header: response status: 200

Body: corridor data in json with relationship of **bus** and **driver** { *id, corridor\_code , driver\_id , driver\_name, bus\_id, bus\_plate\_number, bus\_brand, duty\_date, start\_time, finish\_time, duty\_time\_in\_minute* }

If accessed without token:

Header: response status: 403

Body: { *message: forbidden* }





Sistem manajemen koridor Transjakarta yang lengkap harus melengkapi ketentuan-ketentuan dibawah ini.

Menu	Detail
Login	<ul style="list-style-type: none"> <li>• User bisa melakukan login (dan logout) ke system pada aplikasi.</li> <li>• Setelah Login, user mendapatkan akses ke menu Bus, Supir, dan Koridor.</li> <li>• Jika user sudah melakukan login, halaman login tidak dapat di akses.</li> <li>• Pastikan sistem mencegah unauthorized user ke menu.</li> <li>• Tidak ada fitur register.</li> </ul>
Bus	<ul style="list-style-type: none"> <li>• Pada menu bus, User dapat melakukan create bus.</li> <li>• User dapat melakukan update dan delete terhadap bus yang sudah ada.</li> <li>• Semua form harus terisi pada saat create.</li> <li>• <b>Plate Number</b> harus merupakan <i>string</i>.</li> <li>• <b>Brand</b> harus merupakan string</li> <li>• <b>Type</b> harus merupakan salah satu dari small/big/articulated.</li> <li>• <b>Fuel</b> harus merupakan salah satu dari <i>petrol/gas/diesel/electric</i>.</li> </ul>
Driver	<ul style="list-style-type: none"> <li>• Pada menu driver, User dapat melakukan create driver.</li> <li>• User dapat melakukan update dan delete terhadap driver yang sudah ada berdasarkan driver_id.</li> <li>• Semua form harus terisi pada saat create.</li> <li>• <b>Driver ID</b> harus merupakan sebuah string dan primary key.</li> <li>• <b>Name</b> harus merupakan sebuah string.</li> <li>• <b>Age</b> harus merupakan sebuah string dan minimal 18 tahun.</li> <li>• <b>Gender</b> harus merupakan salah satu dari male/female.</li> <li>• <b>Phone Number</b> harus merupakan sebuah string dan digit.</li> <li>• <b>Address</b> harus merupakan sebuah string.</li> <li>• <b>Email</b> harus merupakan sebuah email.</li> </ul>
Corridor	<ul style="list-style-type: none"> <li>• Pada menu corridor, User dapat melakukan create corridor.</li> <li>• User dapat melakukan delete terhadap corridor yang sudah ada.</li> <li>• Semua form harus terisi pada saat create.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Corridor Code</b> harus merupakan sebuah string.</li> <li>• <b>Driver ID</b> harus merupakan sebuah string.</li> <li>• <b>Bus Plate Number</b> harus merupakan sebuah string.</li> <li>• <b>Duty Date</b> harus merupakan sebuah date.</li> <li>• <b>Start Duty Time</b> harus merupakan sebuah time.</li> <li>• <b>Finish Duty Time</b> harus merupakan sebuah time</li> <li>• Sistem akan menghitung waktu lamanya driver melakukan dinas dalam satuan menit.</li> <li>• Tidak ada Driver atau Bus yang bertabrakan di koridor dan tanggal yang sama.</li> </ul>
--	--

## ERD

ERD di bawah ini bisa dijadikan referensi untuk pembuatan database.

