

Aplikasi Tracker Maintenance Vehicle

Servilog

Kelompok 27

Anggota Kelompok

2206830340

Nahl Syareza
Rahidra

Darmawan
Hanif

2206829175

2206828071

Muhammad
Sesarafli Aljjagra

Salahuddin
Zidane A.

2206028200

2206059805

Muhammad
Lutfi Setiadi



Latar Belakang

Servilog adalah sebuah Platform tracking untuk pemeliharaan kendaraan. ServiLog dirancang untuk mencatat jarak tempuh kendaraan dan memantau umur dari setiap komponen penting dalam kendaraan. Dengan sistem notifikasi prediktif yang dikirimkan 7 hari sebelum sebuah part mencapai batas jarak penggunaannya, ServiLog membantu pengguna mencegah kerusakan mendadak dan biaya perbaikan yang tidak terduga. Cocok untuk pemilik kendaraan pribadi maupun pengelola armada untuk menjaga performa kendaraan dengan mudah dan teratur.



Stack

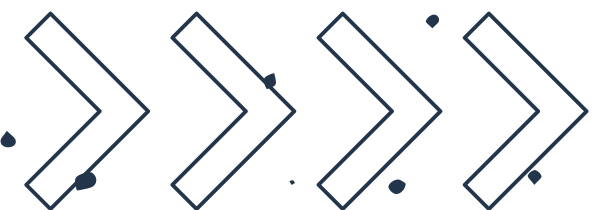
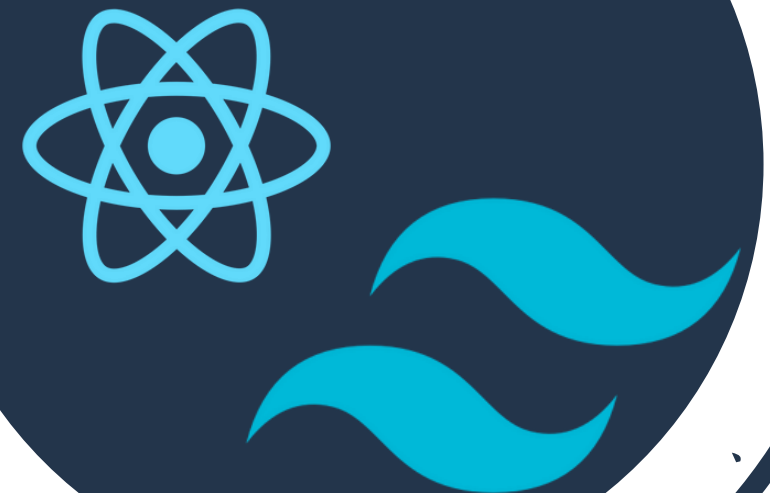
Back-end



Database

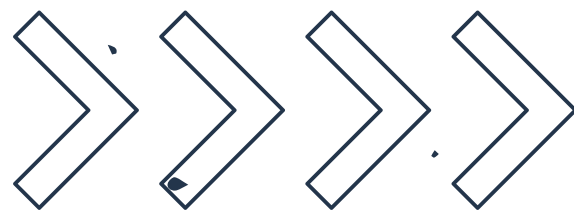


Front-end



Back-end

Alasan penggunaan Node Express sebenarnya dikarenakan lebih familiar.



```
src
├── configs
├── controllers
│   ├── accountController.js
│   ├── mileageController.js
│   ├── partController.js
│   └── vehicleController.js
├── middlewares
├── repositories
│   ├── accountRepository.js
│   ├── mileageRepository.js
│   ├── partRepository.js
│   └── vehicleRepository.js
├── routes
│   ├── accountRoutes.js
│   ├── mileageRoutes.js
│   ├── partRoutes.js
│   └── vehicleRoutes.js
├── services
├── utils
├── .env
├── .gitignore
├── dump.sql
└── index.js
```

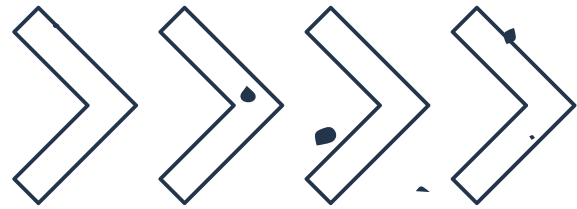

Database

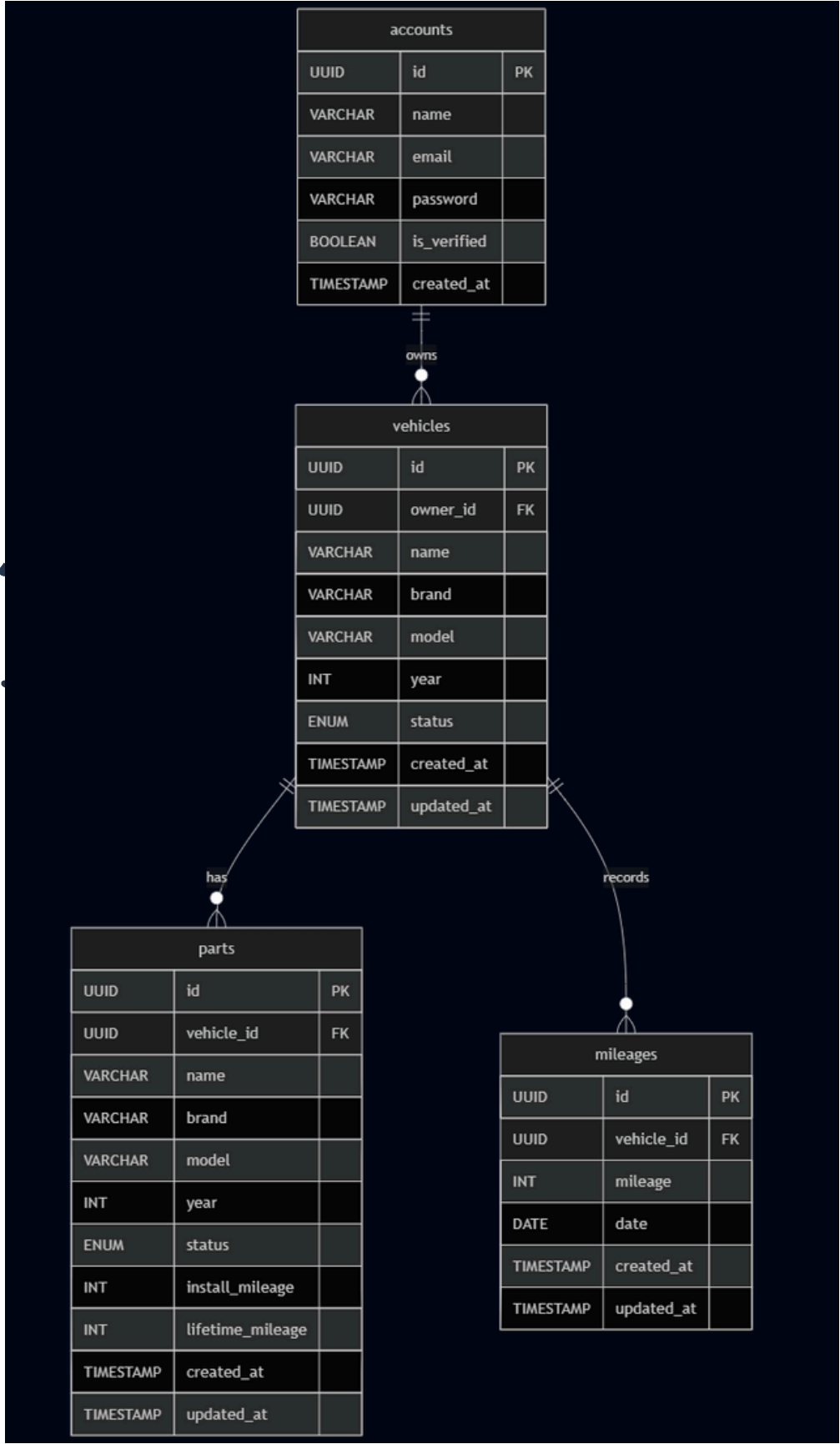
```
ServiLog=> select * from accounts;  
id | name | email | password | is_verified | created_at  
----+-----+-----+-----+-----+-----  
(0 rows)
```

```
ServiLog=> select * from mileages;  
id | vehicle_id | mileage | date | created_at | updated_at  
----+-----+-----+-----+-----+-----  
(0 rows)
```

```
ServiLog=> select * from vehicles;  
id | owner_id | name | brand | model | year | status | created_at | updated_at  
----+-----+-----+-----+-----+-----+-----+-----+-----  
(0 rows)
```

```
ServiLog=> select * from parts;  
id | vehicle_id | name | brand | model | year | status | install_mileage | lifetime_mileage | created_at | updated_at  
----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----  
(0 rows)
```





Database

Front-End

ServiLog

Dashboard Profile



Harian

Mazda - Mazda2 (2018)

Status: **Maintenance Due**

Created: 5/19/2025, 7:29:15 PM

Updated: 5/23/2025, 8:49:29 PM

Edit Vehicle

Delete Vehicle

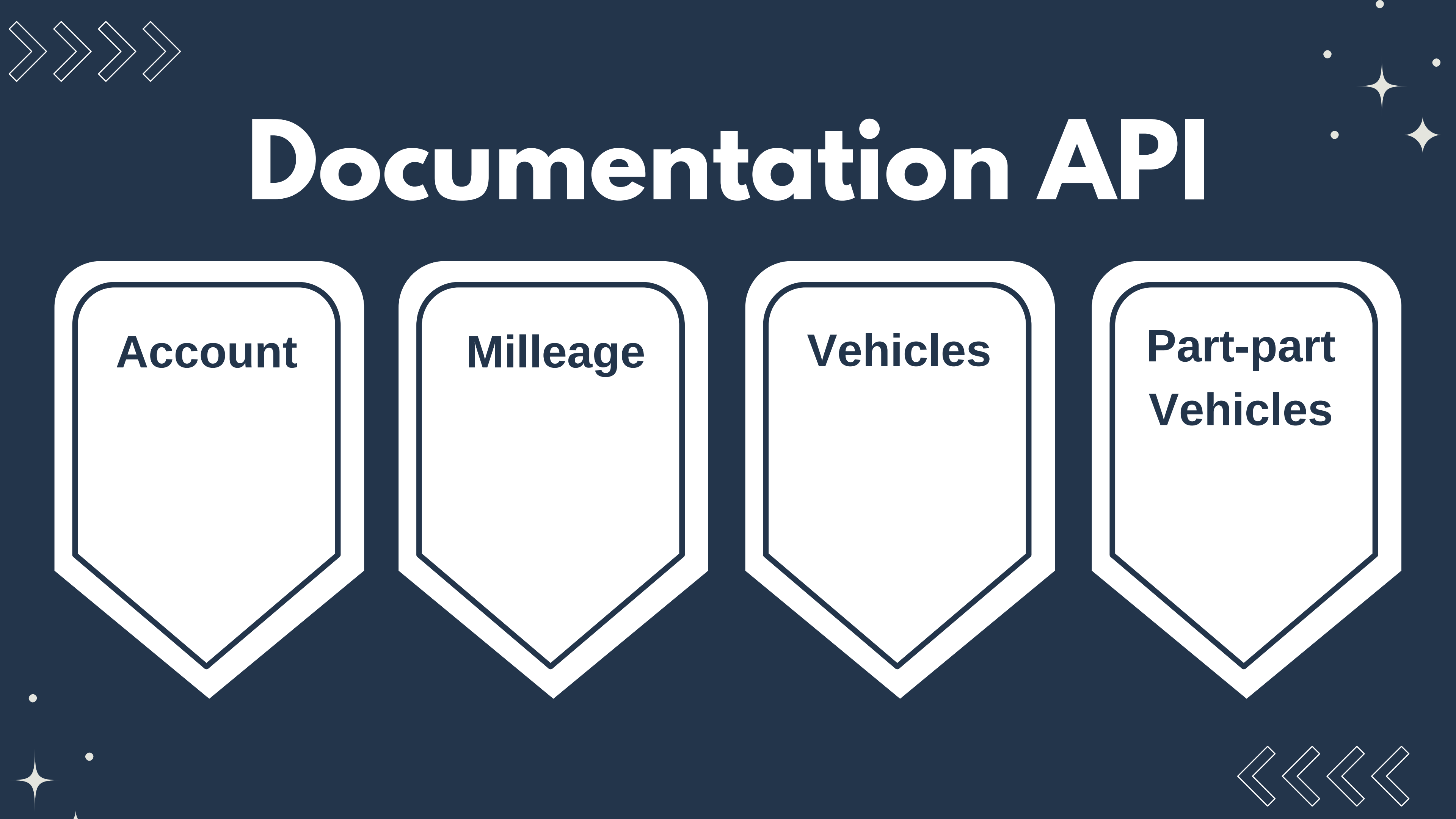
Parts

Mileages

Add Part

Name	Brand	Model	Year	Status	Install Mileage	Lifetime	Actions		
Engine Oil	Mazda	Mazda2	2018	MAINTAINED/REPLACED	1000	10000	Edit	Maintain/Replace	Delete
Clutch	Mazda	Mazda2	2018	Maintenance Due	6000	8000	Edit	Maintain/Replace	Delete

Selain React.js dan tailwind sebagai framework, kita juga menggunakan axios untuk fetching ke back-endnya



Documentation API

Account

Milleage

Vehicles

**Part-part
Vehicles**



Accounts



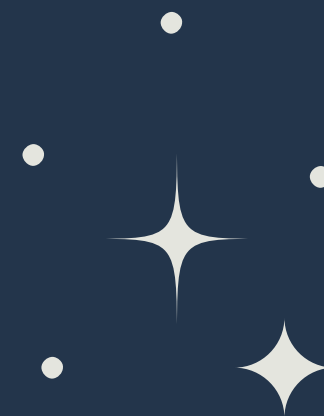


```
const express = require('express');
const accountController = require('...../controllers/accountController');
const authenticator = require('...../middlewares/Authenticator');
const router = express.Router();

router.post('/login', accountController.login);
router.post('/register', accountController.register);
router.post('/verify/:id', accountController.verifyAccount);
router.put('/:id', authenticator, accountController.updateAccount);
router.delete('/:id', authenticator, accountController.deleteAccount);

module.exports = router;
```

Route pada account lebih mengarah bagaimana kerja account pada umumnya yaitu seperti login, register, hapus akun dan verify untuk tokennya.



Milleage





```
const express = require('express');
const mileageController = require('../controllers/mileageController');
const authenticator = require('../middlewares/Authenticator');
const router = express.Router();

router.get('/id/:id', authenticator, mileageController.getMileagebyId);
router.get('/vehicle/:vehicleId', authenticator, mileageController.getMileagesbyVehicleId);
router.post('/', authenticator, mileageController.createMileage);
router.put('/:id', authenticator, mileageController.updateMileage);
router.delete('/:id', authenticator, mileageController.deleteMileage);


module.exports = router;
```

Sesuai nama,
Milleage akan
membahas
milleage melalui
ID pada vehicle
yang ada pada
garage yang
dimasukkan
oleh user juga



Parts






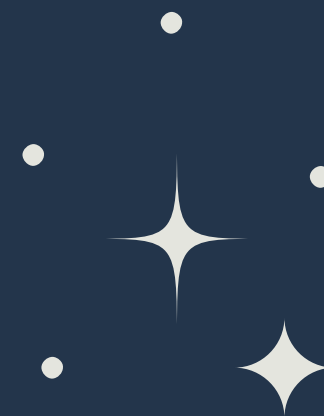
```
const express = require('express');
const partController = require('../controllers/partController');
const authenticator = require('../middlewares/Authenticator');
const router = express.Router();

router.get('/id/:id', authenticator, partController.getPartbyId);
router.get('/vehicle/:vehicleId', authenticator, partController.getPartsbyVehicleId);
router.post('/', authenticator, partController.createPart);
router.post('/maintain/:id', authenticator, partController.maintainPart);
router.put('/:id', authenticator, partController.updatePart);
router.delete('/:id', authenticator, partController.deletePart);

module.exports = router;
```



Part yang dimaksud adalah part pada vehicle. Dengan adanya bagian part ini menjelaskan kapan part-part harus direparasi atau diservice sesuai milleage



Vehicle





```
const express = require('express');
const vehicleController = require('.../controllers/vehicleController');
const authenticator = require('.../middlewares/Authenticator');
const router = express.Router();

router.get('/id/:id', authenticator, vehicleController.getVehiclebyId);
router.get('/account/:accountId', authenticator, vehicleController.getVehiclesbyAccountId);
router.post('/', authenticator, vehicleController.createVehicle);
router.put('/:id', authenticator, vehicleController.updateVehicle);
router.delete('/:id', authenticator, vehicleController.deleteVehicle);

module.exports = router;
```

Kembali lagi
sesuai nama,
Vehicle ini akan
menjadi kepala
di antara
milleage dan
part yang sudah
ditambah di
garagenya.



Deployment





Kami telah meng-deploy menggunakan Vercel, yang memungkinkan proses deployment secara cepat dan efisien.

Backend dari aplikasi yang sudah kami deploy dapat di akses melalui link:
<https://servilog-backend.vercel.app>

Frontend dari aplikasi yang sudah kami deploy dapat di akses melalui link:
<https://servilog.vercel.app>





Thanks

