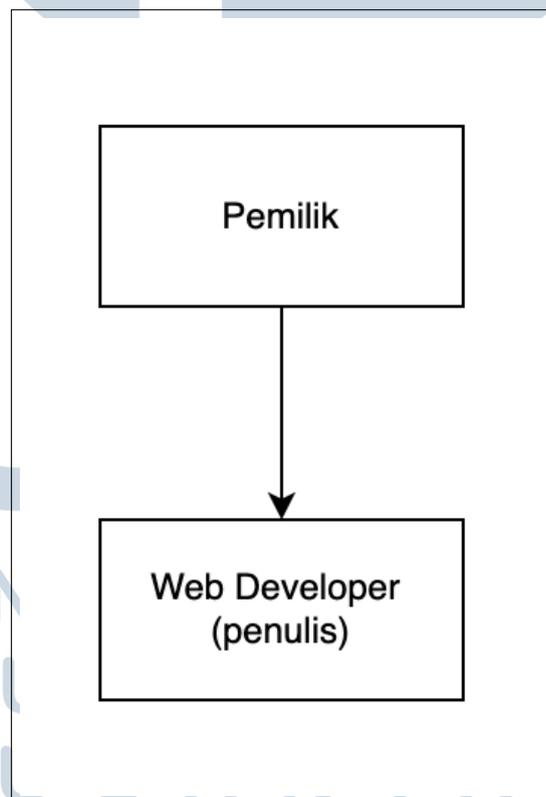


BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Kedudukan penulis pada struktur organisasi ini dapat terlihat pada Gambar 3.1. Penulis berada di bawah pengawasan langsung pemilik dari *Mustbuy Autohauz*, Bayu Riyantino dalam proses praktik kerja magang. Koordinasi pada praktik kerja magang ini dilakukan secara luring dan juga daring. Penulis berkoordinasi secara langsung dengan pemilik *Mustbuy Autohauz*. *Requirement* dari proyek yang dikerjakan diberikan langsung juga oleh pemilik usaha. Penulis juga memberikan laporan berupa *progress* mingguan yang dapat dilakukan secara luring atau daring melalui *Whatsapp*. Penulis berada pada posisi web developer yang melakukan pekerjaan pembangunan sistem *back-end* yang digunakan pada *website Mustbuy Autohauz*.



Gambar 3.1. Kedudukan dan organisasi UMKM *Mustbuy Autohauz*

3.2 Tugas yang Dilakukan

Pada *Mustbuy Autohauz* penulis bekerja sebagai *web developer* yang memiliki tugas utama untuk melakukan pembangunan sistem *back-end*. *Back-end* tersebut diperlukan sebagai sistem utama dari *website Mustbuy Autohauz*. Sistem *back-end* yang diperlukan memiliki *requirement* sebagai berikut:

- Memiliki tampilan *admin page*.
- *Admin* dapat melihat *stock* mobil yang ada.
- *Admin* dapat melakukan penambahan mobil.
- *Admin* dapat melakukan penghapusan mobil.
- *Admin* dapat melakukan pembaharuan mobil.
- Pengguna biasa dapat melihat mobil yang ada beserta dengan info lengkapnya.

Tools yang digunakan dibebaskan kepada pilihan penulis. Selain itu penulis juga harus melakukan *update* kepada pemilik *Mustbuy Autohauz* setiap minggu.

3.3 Uraian Pelaksanaan Magang

Penulis melakukan praktik kerja magang selama 15 minggu. Praktik kerja magang berdurasi dari 7 Maret 2022 hingga 17 Juni 2022. Pelaksanaan kerja magang tiap minggunya diuraikan seperti pada Tabel 3.1.

U M W I N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

| Minggu Ke - | Pekerjaan yang dilakukan |
|-------------|---|
| 1 | Memahami cara kerja bisnis <i>Mustbuy Autohauz</i> . |
| 2 | Melakukan diskusi terhadap kebutuhan usaha. |
| 3 | Memberikan <i>requirement</i> sistem yang diperlukan berdasarkan hasil diskusi minggu sebelumnya. |
| 4 | Melakukan <i>research</i> untuk <i>tools</i> yang diperlukan. Menjelaskan alasan pemilihan <i>tools</i> yang digunakan. |
| 5 | Mempelajari teknologi yang digunakan untuk pembangunan sistem. |
| 6 | Melakukan pembuatan <i>API</i> yang dibutuhkan oleh sistem |
| 7 | Pembuatan tampilan sederhana untuk penggunaan <i>API</i> yang telah dibuat |
| 8 | Libur lebaran |
| 9 | Libur lebaran |
| 10 | Melakukan perubahan pada sistem dikarenakan adanya perubahan struktur data |
| 11 | Memastikan struktur data sudah benar dan melakukan penyesuaian terhadap tampilan. |
| 12 | Memastikan tidak ada <i>bug</i> pada sistem. |
| 13 | Melakukan dokumentasi terhadap sistem yang telah dibuat. |
| 14 | Libur (pemilik harus menghadiri <i>event</i>). |
| 15 | Mengurus dokumen penyelesaian magang. |

3.3.1 Sistem Back-end Mustbuy Autohauz

A. Skema Basis Data

Untuk pembangunan *back-end* yang digunakan pada usaha *Mustbuy Autohauz* diperlukan sebuah basis data. Basis data yang digunakan berfungsi untuk menyimpan data mobil yang diperlihatkan pada *website*. Pada Gambar 3.2 dapat terlihat struktur tabel yang digunakan pada basis data sistem ini. *NoSQL* merupakan basis data yang berbentuk *non relational database* sehingga tidak adanya hubungan antar tabel.

| Car's Table | |
|-------------|---------------|
| PK | <u>id</u> |
| | name |
| | engine |
| | transmission |
| | interiorColor |
| | exteriorColor |
| | type |
| | price |
| | description |

Gambar 3.2. Gambar tabel mobil

Terlihat pada Tabel 3.2 deskripsi lengkap dari properti yang digunakan pada sistem ini.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tabel 3.2. Skema basis data mobil

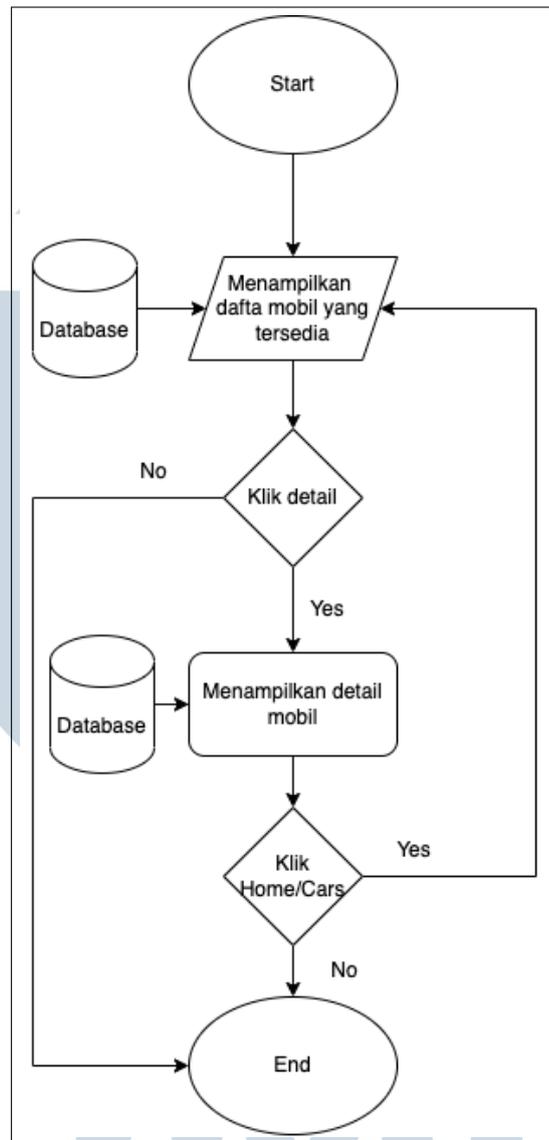
| Data | Tipe data | Keterangan |
|---------------|-----------|--|
| id | ObjectId | <i>Primary key</i> yang membedakan antara satu <i>item</i> dengan <i>item</i> lain |
| name | String | Menyimpan nama mobil |
| engine | String | Menyimpan data mesin mobil |
| transmission | String | Menyimpan data transmisi |
| interiorColor | String | Menyimpan warna <i>interior</i> mobil |
| exteriorColor | String | Menyimpan warna <i>exterior</i> mobil |
| type | String | Menyimpan tipe mobil |
| price | Int | Menyimpan harga mobil |
| year | Int | Menyimpan tahun mobil |
| description | String | Menyimpan detail lain mobil |

Sistem hanya memiliki satu tabel saja untuk menyimpan data mobil. Basis data pada sistem ini disimpan pada *MongoDB Atlas*. Penggunaan *MongoDB Atlas* sendiri dipilih dikarenakan teknologi *NoSQL* yang lebih efisien dibandingkan dengan basis data yang menggunakan *SQL* [5].

B. Diagram Flow Pengguna

Pengguna dari *website Mustbuy Autohauz* tergambar pada Gambar 3.3.





Gambar 3.3. *User flow* pengguna dalam menggunakan *website Mustbuy Autohaus*

Pengguna biasa dapat melihat stok mobil yang ada pada *Mustbuy Autohaus*. Pengguna disajikan gambar mobil beserta dengan nama dan tipe mobil. Jika pengguna tertarik maka pengguna dapat memencet tombol detail untuk dapat melihat informasi lengkap dari mobil tersebut. Informasi tersebut diambil dari basis data *website Mustbuy Autohaus*.

C. Teknologi yang digunakan

Pada projek ini terdapat beberapa teknologi yang digunakan untuk mewujudkan sistem yang dibangun. Teknologi yang digunakan pada praktik kerja magang ini adalah sebagai berikut:

1. *MongoDB Atlas*

MongoDB Atlas digunakan untuk melakukan penyimpanan data terhadap informasi mobil. *MongoDB Atlas* melakukan penyimpanan data tanpa menggunakan *NoSQL* dan menyimpan data secara daring pada *cloud*.

2. *Node*.

Node digunakan untuk melakukan konfigurasi terhadap server.

3. *Express JS*

Express JS digunakan untuk melakukan konfigurasi API pada server.

4. *React JS*

React JS digunakan untuk membuat *front-end* sederhana.

5. *Axios*

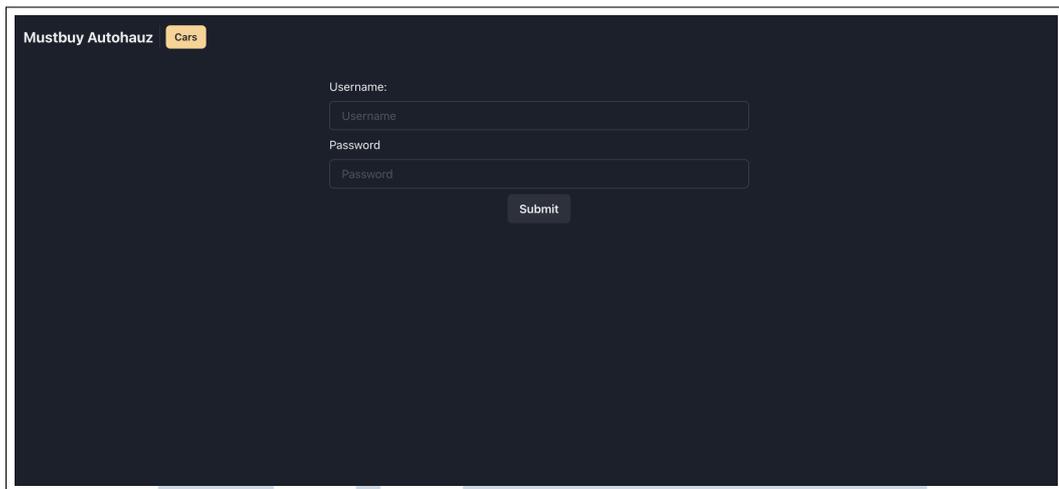
Axios digunakan untuk melakukan *HTTP Request* dari *front-end* ke API yang telah dibuat.

6. *Chakra UI*

Chakra-ui digunakan sebagai *library* untuk memperindah tampilan *website Mustbuy Autohauz*.

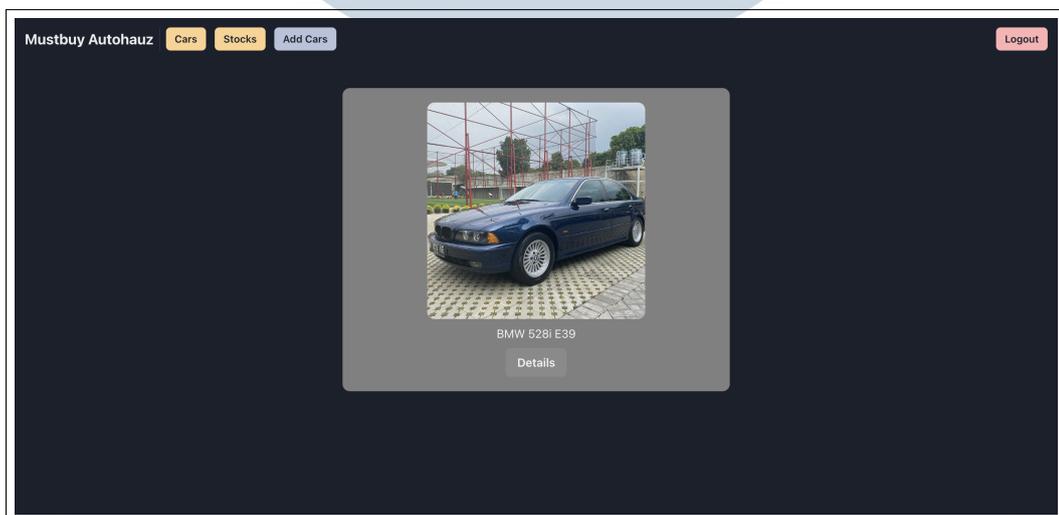
D. Implementasi

Implementasi sistem untuk *website Mustbuy Autohauz* berhasil dilakukan sesuai dengan *requirement* yang diperlukan. Bukti dari implementasi *login* dapat terlihat pada Gambar 3.4. Halaman ini dapat diakses dengan melakukan membuka */login*. *Login page* hanya dapat diakses melalui tautan agar para pengguna tidak merasa terganggu dengan adanya *login button* pada tampilan pengguna. Setelah *login* halaman *web* dialihkan ke laman utama dapat terlihat pada Gambar 3.5.



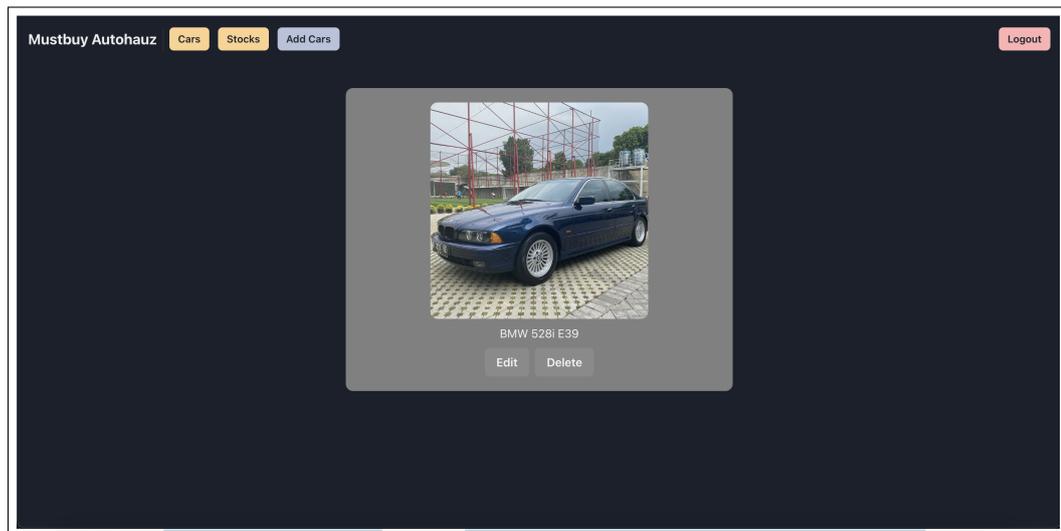
Gambar 3.4. Halaman *login*

Tampilan setelah *login* dibedakan melalui *navbar* yang memiliki tampilan lebih banyak. Tampilan *navbar* yang berbeda tersebut dapat membawa admin ke halaman untuk menunjukkan produk dan menambahkan mobil. *Navbar* juga menyediakan tombol *logout* agar admin dapat keluar dari tampilan admin tersebut.



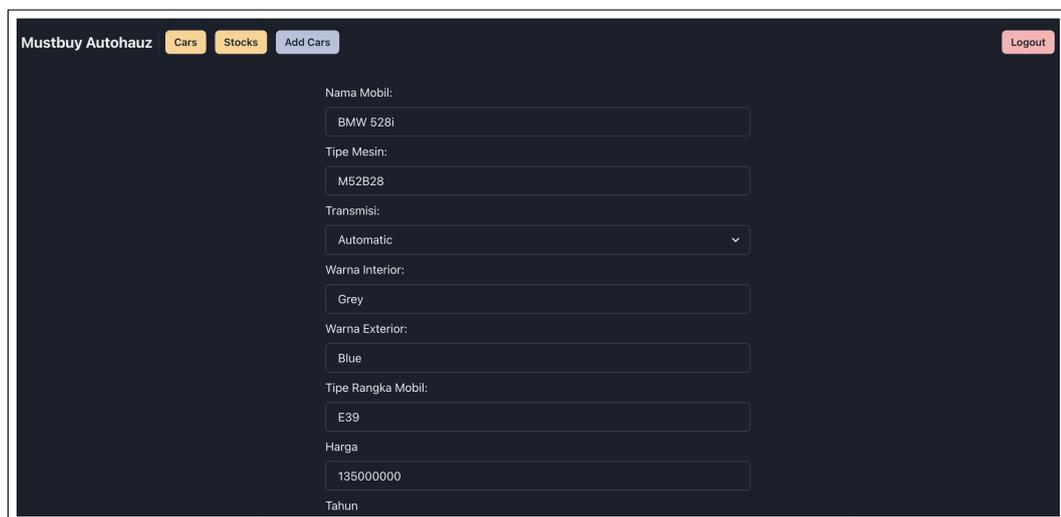
Gambar 3.5. Halaman setelah *login*

Pada Gambar 3.6 dapat terlihat tampilan admin terhadap mobil-mobil yang ada pada *website*. Perbedaan halaman *stocks* adalah dengan halaman *cars* adalah tombol yang tersedia. Pada halaman tersebut tombol yang tersedia adalah tombol *edit* dan tombol *delete*. Tombol *edit* bertujuan untuk melempar admin ke halaman *edit* yang menyediakan *detail item* yang dapat diperbaharui.



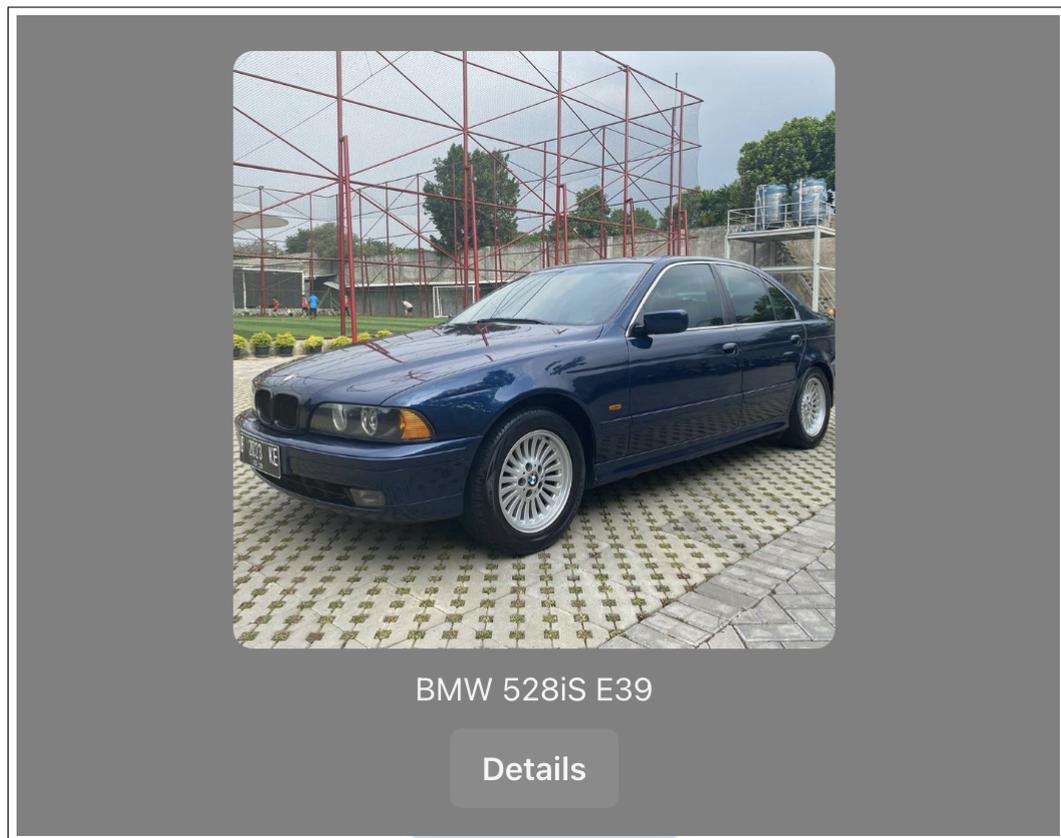
Gambar 3.6. Halaman *stocks*

Ketika melakukan *edit* pada *item* maka admin dialihkan ke halaman *edit* seperti yang terlihat pada Gambar 3.7. Informasi *detail* dari mobil tersebut menjadi *default value* pada halaman *edit*. Dapat terlihat *input* yang ada pada halaman tersebut sudah memiliki *value* yang sesuai dengan *default value* yang telah ditetapkan.



Gambar 3.7. Halaman *edit*

Pada halaman *edit* dilakukan pengetesan untuk melakukan pembaharuan data terhadap produk *BMW 528i E39*. Tes ini mengganti data nama mobil menjadi *BMW 528iS*. Tes tersebut berhasil dilakukan dapat terlihat pada Gambar 3.8. Nama dari mobil atau produk tersebut sudah berubah sesuai dengan nilai yang diberikan pada halaman sebelumnya.



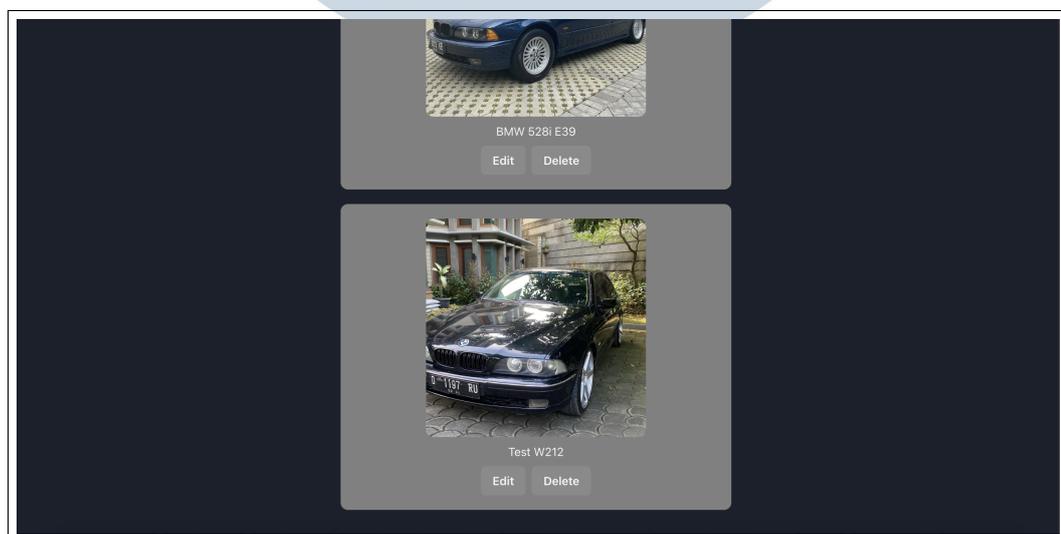
Gambar 3.8. Halaman setelah *edit*

Ada juga halaman *add cars* untuk melakukan penambahan produk oleh admin. Halaman tersebut dapat menerima *input* untuk penambahan mobil baru. Seluruh *input field* atau data yang ada pada halaman tersebut harus diisi sebelum dapat menambahkan produk. Halaman *add cars* yang telah diisi oleh data produk untuk tes dapat dilihat pada Gambar 3.9.

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

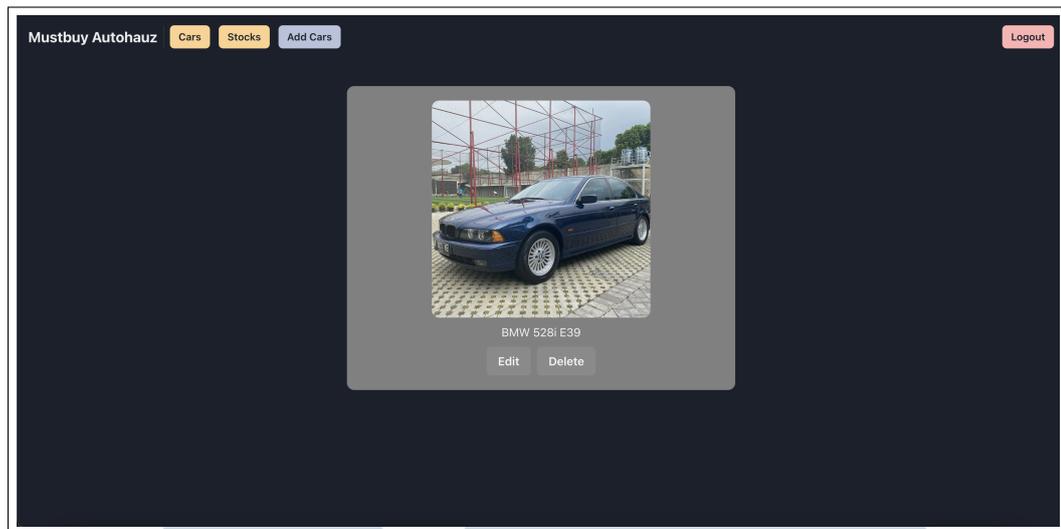
Gambar 3.9. Halaman *add*

Pada halaman *add cars* dilakukan pengetesan untuk menambahkan produk. Produk yang ditambahkan bernama *Test* dengan nomor rangka W212. Terlihat pada Gambar 3.10 bahwa produk tersebut berhasil ditambahkan.



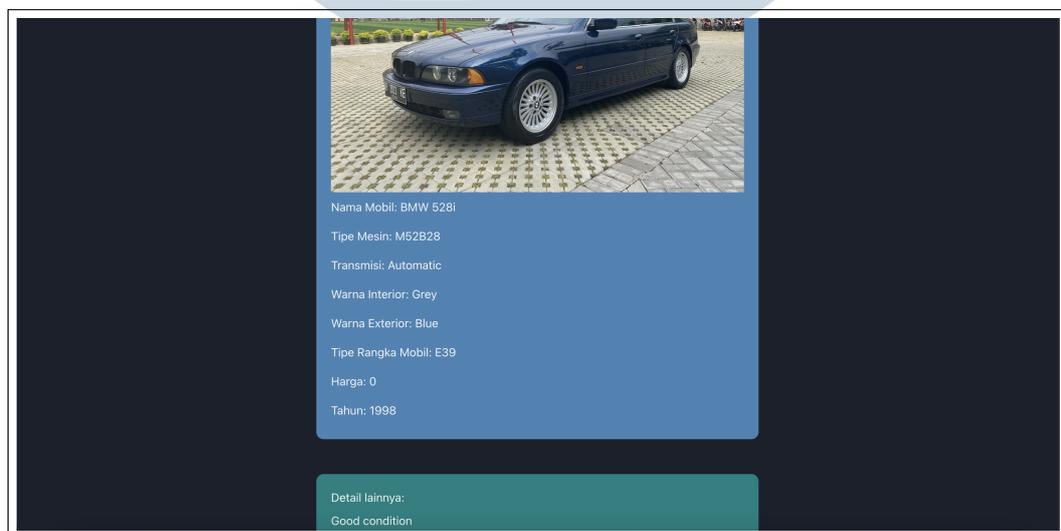
Gambar 3.10. Halaman setelah *add item*

Setelah mobil baru berhasil ditambahkan maka dilakukan pengetesan lain. Pengetesan yang dilakukan adalah melakukan *delete* terhadap *item* yang telah dibuat pada halaman sebelumnya. Tes ini melakukan penghapusan terhadap mobil dengan nama *Test W212*. Dapat terlihat pada Gambar 3.11 bahwa tampilan produk mobil dengan nama *Test W212* sudah berhasil dihapus.



Gambar 3.11. Halaman setelah *delete item*

Selain itu ada juga halaman *detail* yang dapat diakses oleh publik dan admin. Halaman ini terdiri dari detail produk mobil tersebut. Pada Gambar 3.12 dapat terlihat bahwa informasi lengkap dari mobil *BMW 528i E39*.



Gambar 3.12. Halaman *detail*

3.4 Kendala dan Solusi yang Ditemukan

Kendala yang ditemui oleh penulis dalam pembuatan program ini adalah:

1. Melakukan pengetesan API.

Penulis ketika melakukan pengetesan API memiliki masalah ketika menggunakan *Postman*. Hal ini terjadi ketika penulis ingin menambahkan

data namun data tersebut tidak berhasil ditambahkan ketika ingin melakukan *POST*. Namun permasalahan yang terjadi adalah penulis tidak teliti dalam mengirimkan data melalui *Postman*. Penulis mengirimkan data berupa *raw data* yang seharusnya berbentuk *JSON*.

2. Melakukan *HTTP Request*.

Penulis memiliki kendala ketika melakukan *HTTP Request*. Hal ini terjadi dikarenakan penulis yang belum memiliki pengalaman menggunakan API untuk berinteraksi dengan data. Penulis awalnya tidak menggunakan *library* untuk melakukan *HTTP Request*. Setelah melakukan pembacaan dan bertanya kepada teman yang memiliki pengalaman lebih maka penulis mendapatkan informasi bahwa harus menggunakan *library axios* untuk melakukan *HTTP Request*.

3. Melakukan *upload* gambar pada server.

Penulis memiliki kendala ketika ingin melakukan *upload* terhadap gambar yang diberikan oleh admin. Server tidak dapat menerima gambar dan menyimpan gambar tersebut pada server. Permasalahan tersebut berhasil ditangani dengan melakukan riset terhadap cara melakukan *upload* data ke server. Kesalahan yang terbesar adalah melakukan *upload* dan data lainnya sebagai *JSON* sedangkan bentuk data yang diperlukan adalah *FormData*. Sehingga penulis melakukan penyesuaian terhadap sistem baik secara *front-end* dan penyesuaian pada API.

4. Melakukan pengambilan gambar dari server.

Kendala penulis pada berhasil melakukan *upload* gambar pada server adalah pengambilan gambar. Hal ini dikarenakan *path* yang digunakan oleh penulis salah. Penulis membuat sistem *front-end* dengan membaca *path* menggunakan */public*. Sementara *path* yang seharusnya digunakan tidak menggunakan */public*.

Meskipun dalam praktik kerja magang ini penulis mengalami banyak kendala seluruh kendala tersebut berhasil diatasi. Penyelesaian masalah cukup menantang dikarenakan kekurangan pengalaman yang dimiliki oleh penulis. Namun dengan cukup waktu dan percobaan sistem berhasil dibuat.