

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pada pelaksanaan kerja magang ini, proses koordinasi dan pelaporan dilakukan secara langsung dengan Direktur yakni Bapak Jimmy Susantio karena di PT. Astrinusa Jaya Dharma sebelumnya tidak memiliki divisi IT & Website Development. Proses pelaporan dan koordinasi kerja magang kepada Bapak Jimmy Susantio sebagai pembimbing lapangan dilakukan setiap seminggu sekali melalui platform *Whatsapp* serta aplikasi *video conference* seperti Zoom apabila memang dibutuhkan dalam rangka pemberian *progress* dalam pembangunan sistem dan membahas tentang fitur-fitur yang sudah sesuai yang diinginkan oleh Bapak Jimmy Susantio.

3.2 Tugas dan Uraian Kerja Magang

3.2.1 Tugas Kerja Magang

Tugas yang telah dilakukan pada waktu pelaksanaan kerja magang adalah membuat sebuah sistem informasi berbasis *web* yang dapat mengelola stok persediaan barang dan juga beberapa data lainnya seperti data pemasok yang terdaftar, merek, kategori barang, karyawan. Rancangan basis data juga telah dibuat serta desain web dengan menggunakan *template* Bootstrap 5 yang bersifat *open-source*.

Berikut merupakan tabel yang berisi rincian tugas yang telah dilakukan selama 43 hari atau 9 minggu yang dapat dilihat pada Tabel 3.1.

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 Rincian Tugas Magang

Minggu Ke-	Tugas yang Dilakukan	Durasi
1	<ul style="list-style-type: none"> Berdiskusi dengan <i>supervisor</i> mengenai tugas-tugas yang diberikan, pemberian <i>deadline</i>. Belajar mandiri, memperdalam konsep penggunaan Object-Oriented Programming. Belajar mandiri untuk memperdalam konsep Model-View-Controller pada bahasa pemrograman PHP. 	24 Januari 2022 – 28 Januari 2022
2	<ul style="list-style-type: none"> Melakukan penginstalan Composer, Laragon, MySQL Workbench dan Laravel versi 8 di komputer. Belajar mandiri mengenai dasar-dasar dalam penggunaan Framework Laravel 8 (Routing, Blade Template, Eloquent, Tinker, Database Seeder, Factory, Faker, Pagination, Fitur CRUD (Create, Read, Update, Delete), Image Upload, Middleware, dan lain-lain). 	31 Januari 2022 – 04 Februari 2022
3	<ul style="list-style-type: none"> Menginstalasi project di sistem dan server lokal. Melakukan desain database, membuat database dengan bantuan tools MySQL Workbench 	07 Februari 2022 – 11 Februari 2022

Minggu Ke-	Tugas yang Dilakukan	Durasi
	<ul style="list-style-type: none"> • Membuat tabel-tabel di database yang sesuai dengan kebutuhan sistem dengan Laravel Model dan Migrations. • Melakukan pengujian memasukkan data awal dengan menggunakan <i>Database Seeder</i> ke dalam database. 	
4	<ul style="list-style-type: none"> • Pemilihan <i>template</i> Bootstrap 5 bersifat <i>open-source</i> yang meliputi <i>template</i> admin berisikan komponen, layout, form, widget, dan lainnya. • Melakukan instalasi tema kepada project yang digunakan. • Membuat beberapa Blade Template untuk halaman-halaman profil website, halaman login administrator, halaman <i>dashboard</i> administrator. 	14 Februari 2022 – 18 Februari 2022
5	<ul style="list-style-type: none"> • Membuat sistem login dan logout administrator dengan <i>middleware</i>. • Melanjutkan Membuat Blade Template untuk halaman dashboard, pengelolaan barang, kategori barang, merek, penambahan barang masuk, barang keluar (penjualan), dan <i>supplier</i>. 	21 Februari 2022 – 25 Februari 2022

Minggu Ke-	Tugas yang Dilakukan	Durasi
6	<ul style="list-style-type: none"> Membuat fitur CRUD untuk data pengelolaan merek, kategori, dan supplier. Memperbaiki beberapa <i>bug</i> yang muncul. 	28 Februari 2022 – 04 Maret 2022
7	<ul style="list-style-type: none"> Membuat fitur CRUD untuk data pengelolaan barang. Memperbaiki beberapa <i>bug</i> yang muncul. 	07 Maret 2022 – 11 Maret 2022
8	<ul style="list-style-type: none"> Membuat fitur CRUD untuk data barang masuk, dan data barang keluar (penjualan). 	14 Maret 2022 – 18 Maret 2022
9	<ul style="list-style-type: none"> Membuat dokumen <i>UAT</i> (<i>User Acceptance Test</i>). 	21 Maret 2022 – 25 Maret 2022

3.2.2 Uraian Kerja Magang

- Minggu Pertama (24 Januari 2022 - 28 Januari 2022)**

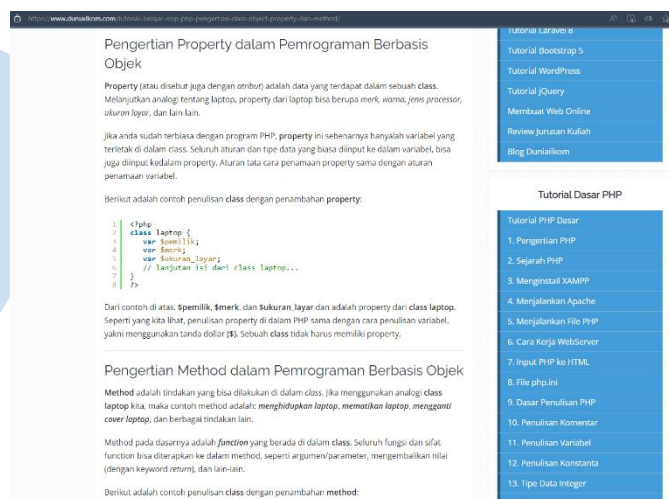
Pada minggu pertama, peserta magang melakukan diskusi dengan *supervisor* terkait dengan ketentuan *jobdesk*, kebutuhan sistem seperti fitur, penggunaan *framework* dan bahasa, serta *deadline* tiap masing-masing tugasnya perminggu.

Setelah melakukan pembahasan dengan pembimbing lapangan, selanjutnya adalah proses kegiatan pembelajaran mandiri melalui internet seperti media pembelajaran YouTube, dokumentasi bahasa PHP, *E-Book*, dan lainnya. Pembelajaran pertama adalah

mempelajari konsep *OOP (Object-Oriented Programming)* pada penggunaan bahasa PHP. Alasan untuk mempelajari konsep *OOP* adalah karena dapat menghemat waktu dan lebih efektif, serta kode yang lebih terstruktur. Konsep *OOP* juga memberikan kemudahan dalam melakukan *maintenance* terhadap kode-kode yang telah dibuat dan kode bersifat *reusable*, artinya penggunaan atau pemanggilan *property* atau *function* yang dapat dilakukan berulang kali[4].

Selain itu, karena proyek yang dibangun akan menggunakan *framework Laravel* yang di mana *framework* tersebut menggunakan konsep *Object-Oriented Programming*, maka harus memiliki pemahaman terkait dasar-dasar konsep penggunaannya, seperti penggunaan *Object*, *Class*, *Property*, *Method*, metode *Inheritance*, dan lainnya.

Proses mempelajari *tutorial-tutorial* mengenai konsep *OOP* dilakukan melalui situs *web* seperti Dunia Ilkom (www.duniailkom.com).



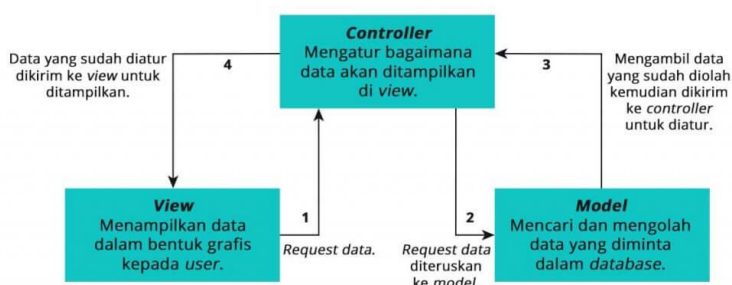
Gambar 3.1 Tampilan Situs Dunia Ilkom

Gambar 3.1 merupakan tampilan situs Dunia Ilkom yang memberikan instruksi-instruksi serta arahan pembelajaran dan pemahaman terkait dengan pemrograman berbasis objek. Setelah

mempelajari konsep *OOP*, selanjutnya adalah pembelajaran mengenai konsep bernama struktur *MVC* (*Model-View-Controller*) yang menggunakan bahasa PHP. Konsep struktur *MVC* ini merupakan konsep yang membagi aplikasi menjadi 3 bagian, yaitu *Model*, *View*, dan *Controller*. Bagian *Model* digunakan untuk mengelola dan mengandung data yang nantinya akan ditampilkan pada bagian *View*. Bagian *View* sendiri berfungsi untuk menampilkan data dari *Model*, dan juga komponen-komponen lainnya seperti gambar, elemen-elemen HTML, dan sebagainya. Lalu bagian *Controller* yang bertugas untuk menghubungkan data yang telah diolah di bagian *Model* dan memberikannya ke bagian *View* untuk dapat ditampilkan di halaman *web* kepada *user* [5].

Tujuan dari penggunaan konsep struktur *MVC* adalah agar membuat struktur kode menjadi lebih rapi, dan dapat melakukan proses pengembangan proyek menjadi lebih efisien karena aplikasi dibagi menjadi 3 bagian sehingga pengembang dapat membagi tugasnya sesuai dengan perannya seperti *Front-End* atau *Back-End*. Selain itu, dapat juga mempermudah *maintenance* kode ataupun melakukan perbaikan pada *bug* dikarenakan lebih rapi dan terstruktur [5].

Berikut adalah diagram struktur dari *MVC* yang dapat dilihat pada gambar 3.2.



Gambar 3.2 Struktur Model-View-Controller

Gambar 3.2 menjelaskan mengenai struktur *MVC* yang terbagi menjadi 3 bagian yaitu *Model*, *View*, dan *Controller*. Pada bagian model merupakan bagian untuk mencari dan mengolah data yang telah di-request oleh basis data yang mengambil data yang telah diolah dan dikirimkan kepada bagian *Controller*. Bagian *Controller* akan mengatur bagaimana data akan ditampilkan dalam bagian *View*. Bagian *View* merupakan tampilan yang dapat dilihat oleh *user* dalam bentuk aplikasi yang dapat menampilkan data dan lainnya.

- **Minggu Kedua (31 Januari 2022 - 04 Februari 2022)**

Pada minggu kedua pelaksanaan tugas magang, proses kegiatan pembelajaran mandiri kembali dilakukan dengan materi yang berbeda yaitu mengenai *Framework Laravel* versi 8 melalui tutorial dari kanal Web Programming UNPAS, dan juga melihat dokumentasi-dokumentasi di *web* *Laravel* serta membeli buku dari Dunia Ilkom yang berjudul *Laravel In Depth*.

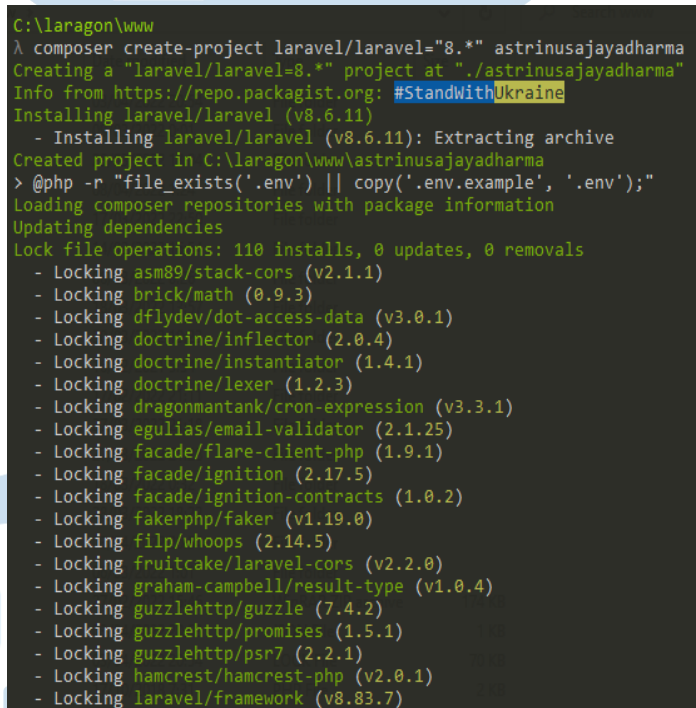
Dari materi-materi tersebut telah dipelajari tentang penggunaan *Framework Laravel* versi 8 yang meliputi :

1. Pengenalan *framework Laravel* versi 8 yang menggunakan konsep *OOP* dan *MVC*.
2. Struktur folder dari *framework* tentang bagian *Model*, *View*, dan *Controller*.
3. Penggunaan *Blade Template Engine* untuk bagian *View*.
4. Cara melakukan *database migration* dan penggunaan *Eloquent Relationship*.
5. Cara penggunaan *Tinker*, *Seeder*, *Factory*, *Faker*.
6. Cara membuat fitur *Create*, *Read*, *Update*, *Delete* pada data.
7. Penggunaan fitur *pagination*.

Sementara melakukan pembelajaran mandiri, proses instalasi *Framework Laravel* juga dilakukan pada laptop pribadi dengan cara menginstal program-program yang dibutuhkan sebelumnya terlebih dahulu seperti *Composer* yang digunakan untuk menginstal *library* dari luar untuk bahasa *PHP*. *Composer* ini digunakan untuk menginstal *Framework Laravel* versi 8. Selain itu dilakukannya instalasi *Laragon* sebagai *web server* di *environment* lokal, dan *MySQL Workbench* sebagai alat untuk mengelola basis data *MySQL*.

- **Minggu Ketiga (07 Februari 2022 - 11 Februari 2022)**

Pada minggu ketiga, dilakukannya proses instalasi proyek baru dan menginstal framework *Laravel* di dalamnya pada *web server* lokal, serta merancang sistem basis data yang diperlukan untuk proyek ini dan menghubungkannya dengan *Laravel*.



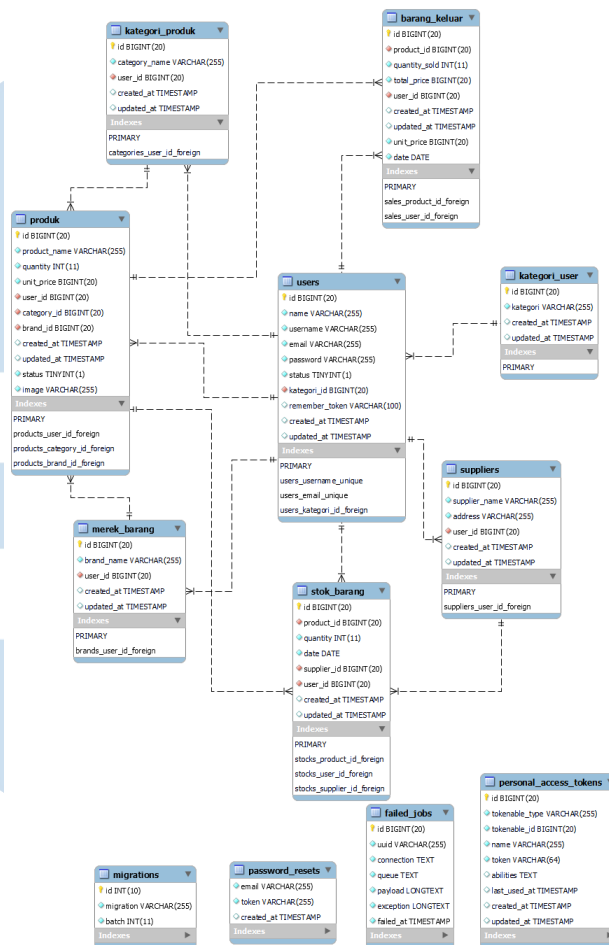
```
C:\laragon\www
λ composer create-project laravel/laravel="8.*" astrinusajayadharma
Creating a "laravel/laravel=8.*" project at ".\astrinusajayadharma"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v8.6.11)
- Installing laravel/laravel (v8.6.11): Extracting archive
Created project in C:\laragon\www\astrinusajayadharma
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 110 installs, 0 updates, 0 removals
- Locking asm89/stack-cors (v2.1.1)
- Locking brick/math (0.9.3)
- Locking dflydev/dot-access-data (v3.0.1)
- Locking doctrine/inflector (2.0.4)
- Locking doctrine/instantiator (1.4.1)
- Locking doctrine/lexer (1.2.3)
- Locking dragonmantank/cron-expression (v3.3.1)
- Locking egulias/email-validator (2.1.25)
- Locking facade/flare-client-php (1.9.1)
- Locking facade/ignition (2.17.5)
- Locking facade/ignition-contracts (1.0.2)
- Locking fakerphp/faker (v1.19.0)
- Locking filp/whoops (2.14.5)
- Locking fruitcake/laravel-cors (v2.2.0)
- Locking graham-campbell/result-type (v1.0.4)
- Locking guzzlehttp/guzzle (7.4.2)
- Locking guzzlehttp/promises (1.5.1)
- Locking guzzlehttp/psr7 (2.2.1)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v8.83.7)
```

Gambar 3.3 Proses Instalasi Proyek Menggunakan Laravel dan Composer

Pada Gambar 3.3 memperlihatkan proses instalasi menggunakan *Laravel* dan *Composer* untuk proyek sistem inventaris pada PT. Astrinusa Jaya Dharma dengan menggunakan *Laragon* dan *Terminal*.

Selanjutnya adalah proses rancangan struktur basis data yang terdiri dari tabel *user*, kategori *user*, produk, kategori_produk, barang_keluar, stok_barang, merek, *supplier* yang digunakan untuk sistem ini yang dapat dilihat pada gambar 3.4.

Tabel *User* dibuat terdiri dari beberapa kolom seperti *ID User* yang menjadi *primary key*, Nama *User*, *Username*, *Password*, *Email*, *Status*, serta *ID Kategori User*. Tabel *Kategori User* dibuat terdiri dari beberapa kolom seperti *ID Kategori* yang menjadi *primary key* dan kolom *Kategori*. Tabel *Produk* berisi kolom-kolom untuk data seperti *ID Produk* sebagai *primary key*, nama produk, kuantitas produk, harga satuan produk, kategori produk yang terhubung dengan tabel kategori produk, dan kolom merek yang terhubung dengan tabel merek. Selanjutnya dibuat tabel *Kategori Produk* yang berisi kolom untuk data *ID Kategori* sebagai *primary key*, serta nama kategori produk, tabel *Merek Produk* yang berisi kolom untuk data *ID Merek* sebagai *primary key* serta nama merek produk, lalu tabel *Supplier* yang berisi kolom untuk data seperti *ID Supplier* sebagai *primary key* serta nama *supplier* dan alamatnya, lalu terdapat tabel *Stok* yang berisi kolom untuk data tanggal produk masuk, jumlah produk yang masuk, dan kolom *supplier*, serta tabel *Sales* untuk barang keluar yang berisi kolom untuk data tanggal produk keluar, jumlah produk yang keluar, serta harga total dari produk yang telah keluar atau dijual.



Gambar 3.4 Desain Sistem Basis Data

Setelah itu, dimasukkan beberapa data *dummy* untuk percobaan memasukkan data ke dalam tabel basis data, dengan menggunakan *Database Seeder* dan *Faker* yang dimiliki oleh *Laravel*. *Database Seeder* berguna untuk melakukan proses pengisian data awal ke dalam basis data. Berikut adalah isi dari *script* kode untuk melakukan *seeding* yang dapat dilihat pada Gambar 3.5 sampai Gambar 3.8.

```

database > seeders > DatabaseSeeder.php > ...
1  <?php
2  namespace Database\Seeders;
3  use App\Models\User;
4  use App\Models\Brand;
5  use App\Models\Category;
6  use App\Models\Product;
7  use App\Models\Supplier;
8  use App\Models\UserCategory;
9  use Illuminate\Database\Seeder;
10
11 class DatabaseSeeder extends Seeder
12 {
13     /**
14      * Seed the application's database.
15      *
16      * @return void
17      */
18     public function run()
19     {
20
21         UserCategory::create([
22             'id' => '1',
23             'kategori' => 'Admin',
24         ]);
25
26         UserCategory::create([
27             'id' => '2',
28             'kategori' => 'Direktur',
29         ]);
30
31         User::create([
32             'name' => 'Admin',
33             'username' => 'admin1',
34             'email' => 'astrinusajayadharma@gmail.com',
35             'password' => bcrypt('12345'),
36             'kategori_id' => '1'
37         ]);
38
39         User::create([
40             'name' => 'Jimmy Susantio',
41             'username' => 'jimmysusantio',
42             'email' => 'jimmysusantio@astrinusajayadharma.com',
43             'password' => bcrypt('12345678'),
44             'kategori_id' => '2'
45         ]);
46
47         Brand::create([
48             'id' => '1',
49             'brand_name' => 'Scania',
50             'user_id' => '1'
51         ]);
52
53         Brand::create([
54             'id' => '2',
55             'brand_name' => 'Volvo',
56             'user_id' => '1'
57         ]);
58     }
59 }

```

Gambar 3.5 Script Seeding untuk tabel kategori user

Script pada Gambar 3.5 digunakan untuk menambah contoh data pada data kategori *user* seperti “Admin”, dan “Direktur”.

```

User::create([
    'name' => 'Admin',
    'username' => 'admin1',
    'email' => 'astrinusajayadharma@gmail.com',
    'password' => bcrypt('12345'),
    'kategori_id' => '1'
]);

User::create([
    'name' => 'Jimmy Susantio',
    'username' => 'jimmysusantio',
    'email' => 'jimmysusantio@astrinusajayadharma.com',
    'password' => bcrypt('12345678'),
    'kategori_id' => '2'
]);

Brand::create([
    'id' => '1',
    'brand_name' => 'Scania',
    'user_id' => '1'
]);

Brand::create([
    'id' => '2',
    'brand_name' => 'Volvo',
    'user_id' => '1'
]);

```

Gambar 3.6 Script Seeding untuk tabel user dan merek produk

Script pada Gambar 3.6 digunakan untuk menambah contoh data pada data user seperti nama user, *username*, *password*, lalu data merek produk seperti nama merek.

```
Category::create([
    'id' => '1',
    'category_name' => 'Body Parts',
    'user_id' => '1'
]);

Category::create([
    'id' => '2',
    'category_name' => 'Engine',
    'user_id' => '1'
]);

Supplier::create([
    'id' => '1',
    'supplier_name' => 'PT. Supplier Test 1',
    'address' => 'Jalan Test 1',
    'user_id' => '1'
]);

Supplier::create([
    'id' => '2',
    'supplier_name' => 'PT. Supplier Test 2',
    'address' => 'Jalan Test 1',
    'user_id' => '1'
]);
```

Gambar 3.7 Script Seeding untuk kategori produk dan supplier

Script pada Gambar 3.7 digunakan untuk menambah contoh data pada data kategori produk seperti “Body Parts”, dan “Engine” serta contoh data *supplier* yang berisi nama *supplier* dan alamatnya.

```
Product::create([
    'id' => '1',
    'product_name' => 'Sania Fog Lamp',
    'quantity' => '10',
    'unit_price' => '400000',
    'user_id' => '1',
    'category_id' => '1',
    'brand_id' => '1',
    'image' => 'https://sc04.alicdn.com/kf/HTB1n0UL0hTpK1RjSZFKq6y2wXaQ.jpg'
]);

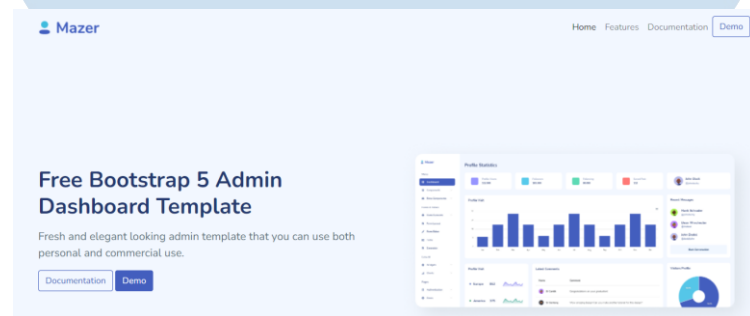
Product::create([
    'id' => '2',
    'product_name' => 'Volvo AC Compressor',
    'quantity' => '10',
    'unit_price' => '200000',
    'user_id' => '1',
    'category_id' => '2',
    'brand_id' => '2',
    'image' => 'https://www.vlvaautoparts.com/vlvaautoparts/images/produits/compresseur_de_clim_volvo_ref_36000231.jpg'
]);
```

Gambar 3.8 Script Seeding untuk data produk

Script pada Gambar 3.8 digunakan untuk menambah contoh data pada data produk seperti yang berisi nama produk, jumlah stok saat data tersebut ditambah, harga satuannya, ID kategori dan merek produk yang terhubung dengan tabel kategori dan merek produk, serta gambar produk.

- **Minggu Keempat (14 Februari 2022 - 18 Februari 2022)**

Pada minggu keempat, dilakukannya pemilihan *template Bootstrap 5* sebagai tampilan untuk sistem admin dan *web* nantinya. Beberapa daftar *template* yang telah dikumpulkan kepada pembimbing lapangan agar dapat dipilih sesuai dengan pembimbing lapangan inginkan. Template yang digunakan adalah *template Bootstrap 5* yang bersifat *open-source* bernama **Mazer**.



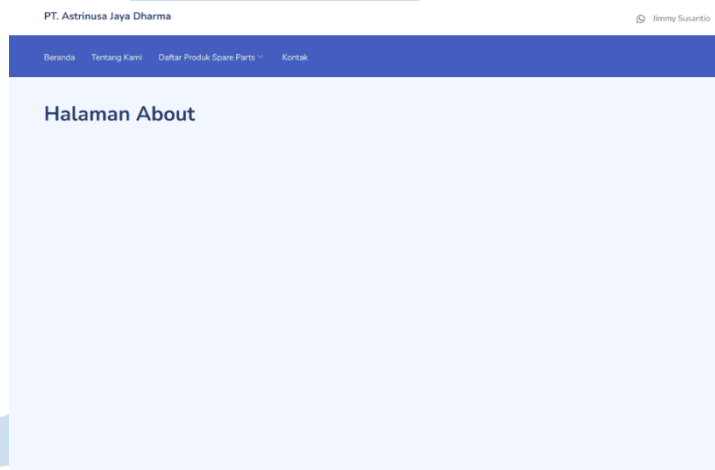
Gambar 3.9 Template Bootstrap 5 Mazer

Setelah itu, dibuatlah beberapa halaman menggunakan fitur *Blade Template Engine* serta *routing*-nya untuk halaman depan yang nantinya dapat dikerjakan oleh bagian *Front-End* seperti halaman beranda, kontak, tentang kami. Lalu dibuat juga halaman untuk sistem adminnya seperti halaman login khusus admin, dan halaman *dashboard* untuk sistem manajemen.



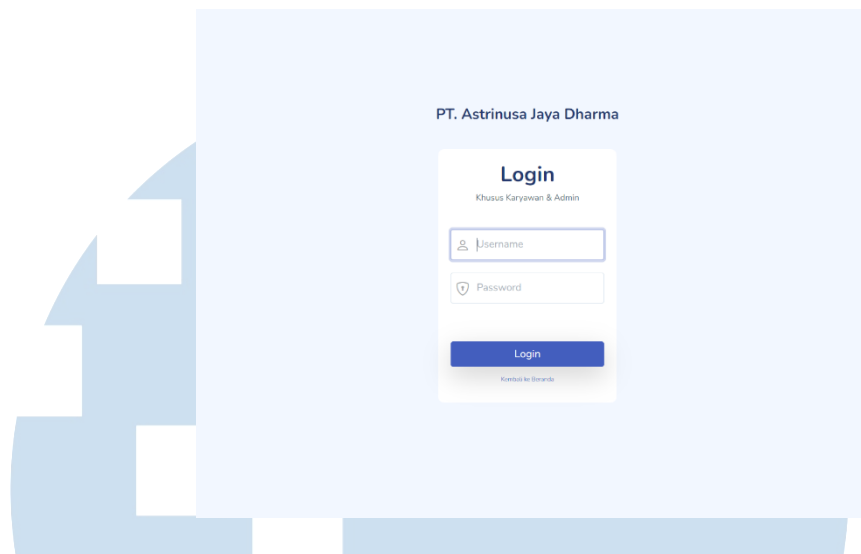
Gambar 3.10 Tampilan Halaman Beranda

Gambar 3.10 merupakan tampilan halaman yang telah dibuat untuk menampilkan halaman bagian beranda ketika pelanggan pertama kali membuka situs perusahaan.



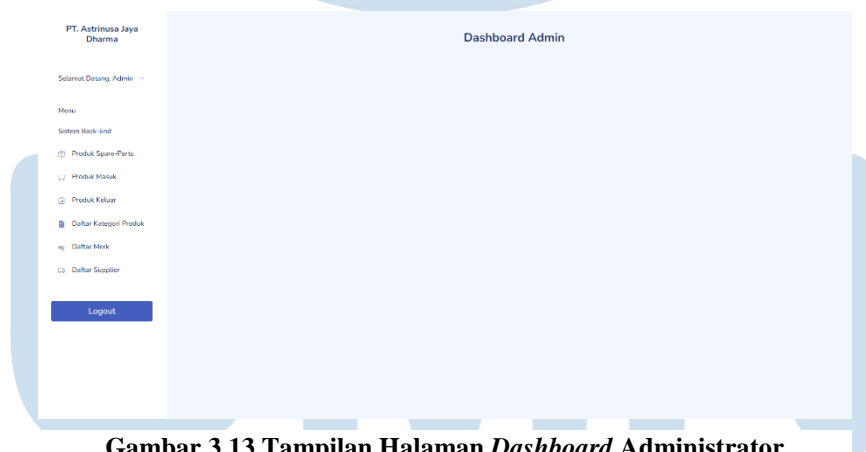
Gambar 3.11 Tampilan Halaman Tentang Kami

Gambar 3.11 merupakan tampilan halaman yang telah dibuat untuk bagian tentang perusahaan yang nantinya bisa diisi untuk informasi-informasi seperti sejarah perusahaan, visi misi, dan lainnya.



Gambar 3.12 Tampilan Halaman *Login* untuk Administrator

Gambar 3.12 merupakan tampilan halaman yang telah dibuat untuk bagian *login* yang hanya dapat dilakukan oleh bagian administrator, yang nantinya akan masuk ke bagian sistem informasi manajemen milik perusahaan.



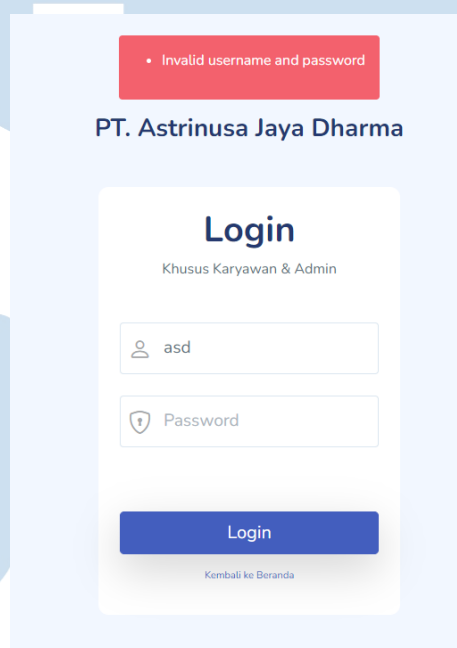
Gambar 3.13 Tampilan Halaman *Dashboard* Administrator

Gambar 3.13 merupakan tampilan halaman yang telah dibuat untuk bagian *dashboard* ketika administrator telah berhasil login ke dalam sistem informasi manajemen dan nantinya menampilkan informasi-informasi penting seperti jumlah total produk, jumlah *supplier*, jumlah kategori, dan lainnya. Serta menu *navigation bar* di samping kiri yang terdapat nama

perusahaan, dan daftar menu yang nantinya dapat diakses, serta tombol logout untuk keluar dari sistem.

- **Minggu Kelima (21 Februari 2022 - 25 Februari 2022)**

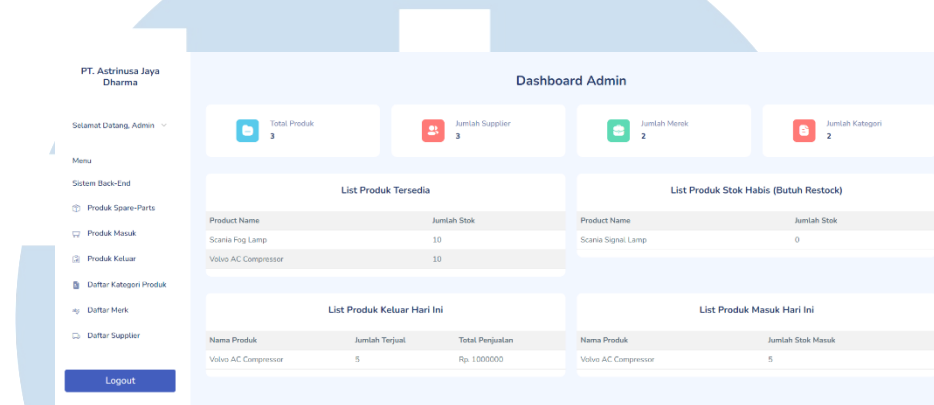
Pada minggu kelima, telah dibuatnya proses sistem *login* dan *logout* untuk sistem informasi manajemen yang hanya dapat dilakukan oleh administrator, setelah itu melakukan pengujiannya menggunakan data yang sudah dibuat pada proses *seeding* minggu sebelumnya, serta membuat notifikasi apabila *username* dan *password* yang dimasukkan salah dan tidak terdaftar pada basis data. Tampilan notifikasi tersebut dapat dilihat pada Gambar 3.14



Gambar 3.14 Notifikasi *username* dan *password* salah

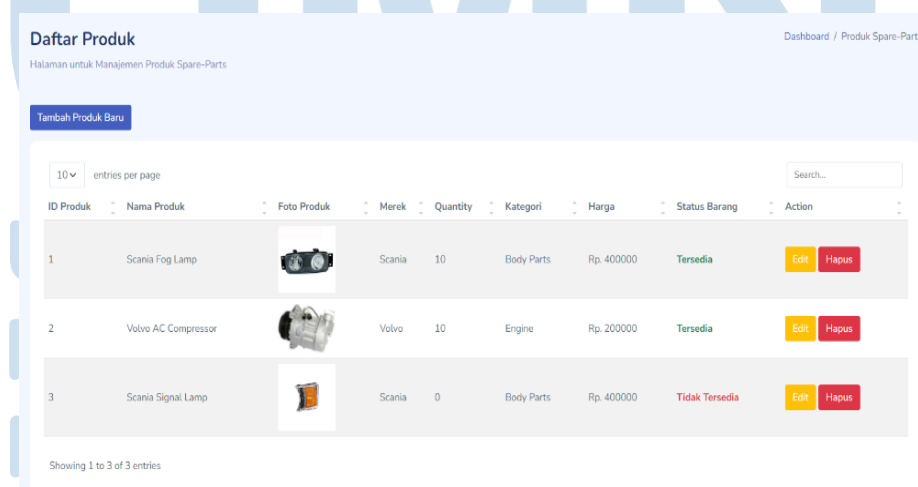
Setelah itu, dibuatlah isi halaman *dashboard* yang berisikan tampilan jumlah produk, *supplier*, merek, kategori produk, dalam bentuk *Card* menggunakan *Bootstrap 5*, serta daftar tabel yang berisi daftar produk

yang memiliki stok dan daftar produk yang stoknya telah habis, serta daftar produk yang masuk dan keluar berdasarkan tanggal tersebut yang dapat dilihat pada Gambar 3.15.



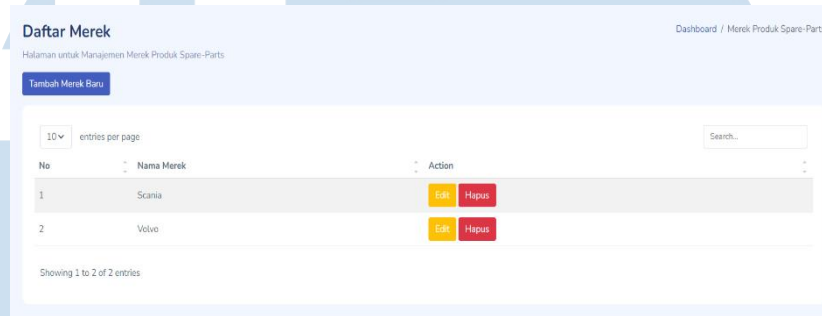
Gambar 3.15 Isi Tampilan Halaman Dashboard

Lalu, dibuat juga halaman-halaman untuk pengelolaan produk yang dapat menampilkan tabel menggunakan *DataTables* untuk data-data produk seperti ID produk, nama produk, foto produk, merek, jumlah stok, kategori produk, harga, dan statusnya (tersedia atau tidak tersedia berdasarkan jumlah stok). Berikut adalah tampilan halaman Produk *Spare-Parts* yang dapat dilihat pada gambar 3.16.



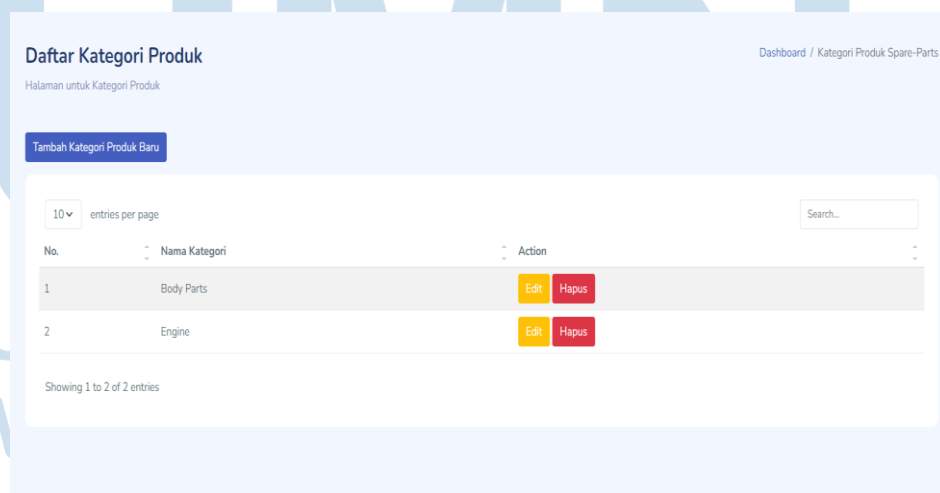
Gambar 3.16 Isi Tampilan Halaman Produk Spare-Parts

Selanjutnya adalah membuat halaman untuk mengelola merek produk yang dapat menampilkan data berupa nama-nama merek yang telah terdaftar pada sistem (contohnya seperti *Scania* dan *Volvo*), yang dapat dilihat pada Gambar 3.17.



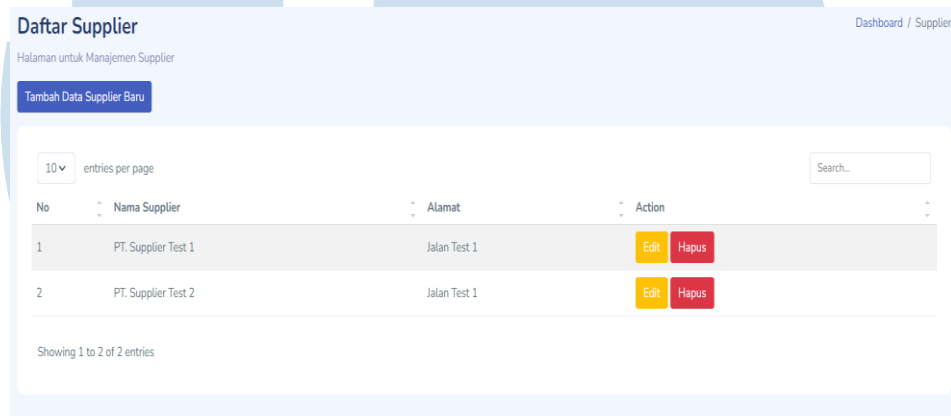
Gambar 3.17 Isi Tampilan Halaman Merek Produk *Spare-Parts*

Selanjutnya adalah membuat halaman untuk mengelola kategori-kategori produk, yang berisikan nama-nama kategori produk seperti *body parts*, *engine*, *filters*, dan lain-lain. Data yang ditampilkan berupa nama kategori yang telah terdaftar pada sistem yang dapat dilihat pada Gambar 3.18.



Gambar 3.18 Isi Tampilan Halaman Kategori Produk *Spare-Parts*

Setelah membuat halaman kategori, selanjutnya dibuatlah halaman *supplier* yang dapat menampilkan data *supplier* yang telah terdaftar pada sistem dan telah bekerja sama dengan perusahaan, data *supplier* ini akan terhubung dengan data produk masuk nantinya. Data yang ditampilkan adalah data nama *supplier* dan alamatnya yang dapat dilihat pada Gambar 3.19.



Gambar 3.19 Isi Tampilan Halaman *Supplier* Produk *Spare-Parts*

Terakhir adalah pembuatan halaman untuk daftar produk yang masuk dan produk yang keluar. Daftar produk masuk artinya ada penambahan stok berdasarkan tanggal tertentu, dan daftar produk keluar artinya ada pengeluaran atau penjualan terhadap produk sehingga stoknya akan berkurang. Berikut adalah gambar halaman daftar produk masuk dan produk keluar yang dapat dilihat pada Gambar 3.20 dan Gambar 3.21.



Gambar 3.20 Isi Tampilan Halaman Stok Produk Masuk

Daftar Produk Keluar
Halaman untuk Manajemen Stok Produk Spare-Parts yang Keluar

Dashboard / Stok Keluar Produk Spare-Parts

Tambah Produk Keluar

10 entries per page Search...

No.	Nama Produk	Jumlah Terjual	Harga Satuan	Total Pendapatan	Tanggal	Action
1	Volvo AC Compressor	5	Rp. 200000	Rp. 1000000	2022-04-11	Edit

Showing 1 to 1 of 1 entries

Gambar 3.21 Isi Tampilan Halaman Stok Produk Keluar

- **Minggu Keenam (28 Februari 2022 - 04 Maret 2022)**

Pada minggu keenam, dibuatnya fitur *CRUD* untuk data 3 macam pengelolaan yaitu kategori produk, merek produk, serta *supplier*. Untuk semua fitur penambahan data menggunakan *form* di halaman yang terpisah dan meminta administrator untuk memasukkan data-data yang diperlukan seperti :

1. Untuk penambahan data kategori produk dan merek produk, sistem akan hanya meminta data yaitu nama kategori dan kategori.
2. Untuk penambahan data *supplier*, sistem akan meminta data nama *supplier* dan alamatnya.

Berikut adalah halaman *form* untuk penambahan data kategori produk yang akan meminta *input* nama kategori produk yang baru. Halaman dapat dilihat pada Gambar 3.22 .

Tambah Kategori Produk Spare-Parts Dashboard / Kategori Produk Spare-Parts / Tambah Kategori Produk

Halaman untuk Menambah Kategori Produk Spare-Parts

Form Input Data Kategori Produk

Masukkan Nama Kategori Baru :

Add Category

Gambar 3.22 Form Penambahan Kategori Produk *Spare-Parts*

Berikut adalah halaman *form* untuk penambahan data merek *supplier* yang akan meminta *input* nama *supplier* dan alamatnya. Halaman pada dilihat pada Gambar 3.23.

Tambah Supplier Produk Spare-Parts Dashboard / Supplier Produk Spare-Parts / Tambah Supplier

Halaman untuk Menambah Supplier Produk Spare-Parts

Form Input Data Supplier

Masukkan Nama Supplier Baru :

Masukkan Alamat Supplier

Tambah Supplier

Gambar 3.23 Form Penambahan Kategori Produk Spare-Parts

Berikut adalah halaman *form* untuk penambahan data merek produk yang akan meminta *input* nama merek produk yang baru. Halaman pada dilihat pada Gambar 3.24.

Gambar 3.24 Form Penambahan Merek Produk Spare-Parts

Setelah membuat fitur penambahan data, selanjutnya adalah membuat fitur untuk dapat memperbarui data (*Update*) pada ketiganya apabila administrator ingin mengubah atau memperbarui data dikarenakan ada kesalahan pada proses *input*. Fitur ini dapat digunakan melalui tombol *Edit* yang ada pada halaman daftar di dalam kolom *Action*.

Berikut adalah halaman untuk melakukan pembaruan data pada kategori produk yang akan meminta *input* nama kategori produk yang baru. Halaman pembaruan pada dilihat pada Gambar 3.25.

Gambar 3.25 Form Pembaruan Data Kategori Produk Spare-Parts

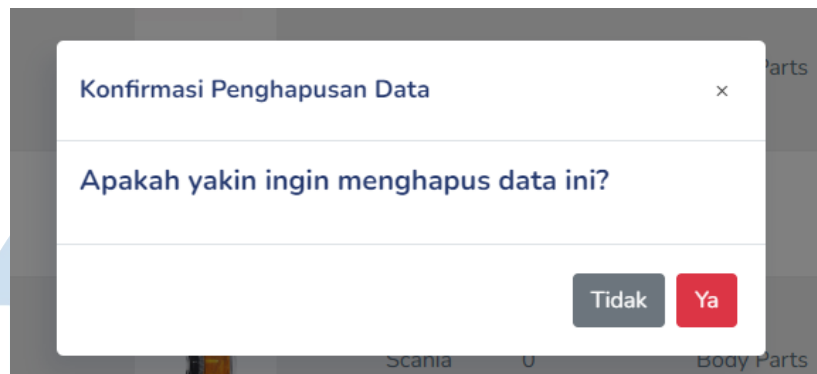
Berikut adalah halaman untuk melakukan pembaruan data pada merek produk yang akan meminta *input* nama merek produk yang baru. Halaman pembaruan pada dilihat pada Gambar 3.26.

Gambar 3.26 Form Pembaruan Data Merek Produk Spare-Parts

Berikut adalah halaman untuk melakukan pembaruan data pada bagian *supplier* yang akan meminta *input* nama *supplier* dan alamat yang baru. Halaman pembaruan pada dilihat pada Gambar 3.27.

Gambar 3.27 Form Pembaruan Data *Supplier* Produk Spare-Parts

Setelah membuat fitur memperbarui/mengubah data, terakhir adalah membuat fitur penghapusan data yang dapat digunakan melalui tombol Hapus yang di samping tombol *Edit*. Tombol hapus akan memunculkan *modal* menggunakan *Bootstrap 5* yang akan mengkonfirmasi apabila ingin benar-benar menghapus data.



Gambar 3.28 Modal untuk Konfirmasi Penghapusan Data

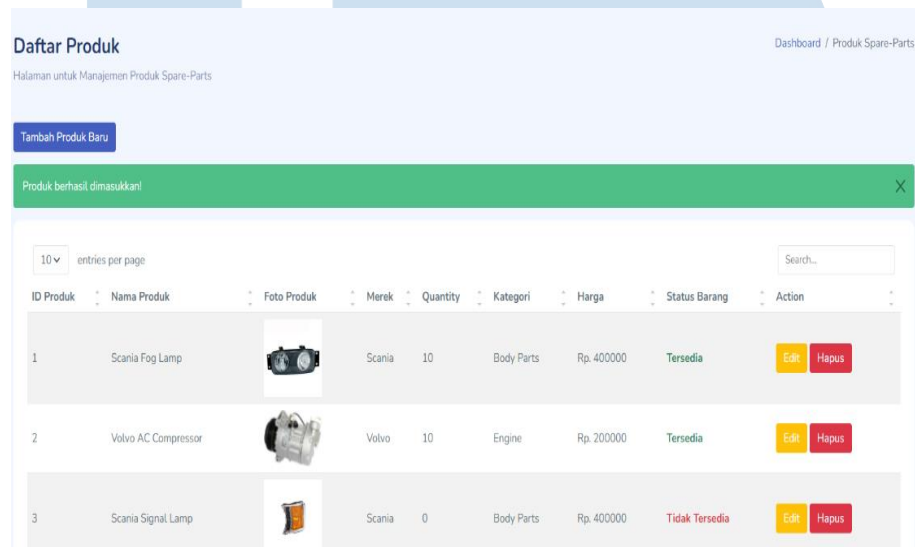
- **Minggu Ketujuh (07 Maret 2022 - 11 Maret 2022)**

Pada minggu ketujuh, dibuatnya fitur *CRUD* untuk data pengelolaan produk *spare-parts*. Sehingga administrator nantinya dapat menambahkan produk baru, mengubah data produk apabila terdapat kesalahan, dan menghapus data produk. Untuk fitur penambahan data produk, telah dibuatkan halaman baru yang berisikan *form* yang nantinya akan diisi oleh administrator tentang data-data produk seperti nama produk, foto produk, kategori produk, merek produk, harga satuannya, serta jumlah stok saat data produk akan dimasukkan. Halaman tambah produk dapat dilihat pada Gambar 3.29.

Gambar 3.29 Isi Tampilan Halaman Form Penambahan Produk

Setelah data-data produk dimasukkan, sistem mengarahkan administrator ke halaman daftar produk dan akan memberikan notifikasi bahwa produk telah berhasil dimasukkan ke dalam daftar produk.

Berikut adalah gambar tampilan notifikasi yang diberikan sistem ketika produk berhasil dimasukkan yang dapat dilihat pada Gambar 3.30.



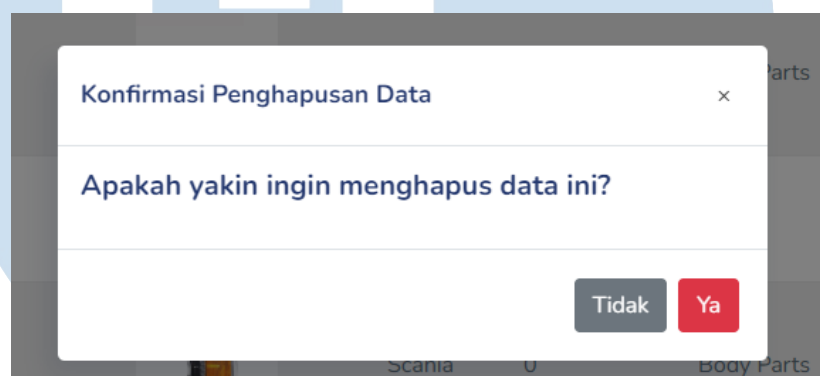
Gambar 3.30 Notifikasi Berhasil Menambah Data Produk

Setelah itu, telah dibuatnya fitur *Update* untuk memperbarui data produk yang sudah terdaftar apabila terdapat kesalahan pada data produk. Fitur *Update* akan menggunakan halaman baru yang terpisah (sama seperti fitur *Create*). Berikut adalah gambar halaman untuk fitur *Update* data pada produk yang dapat dilihat pada Gambar 3.31



Gambar 3.31 Isi Tampilan Halaman Form Update Produk

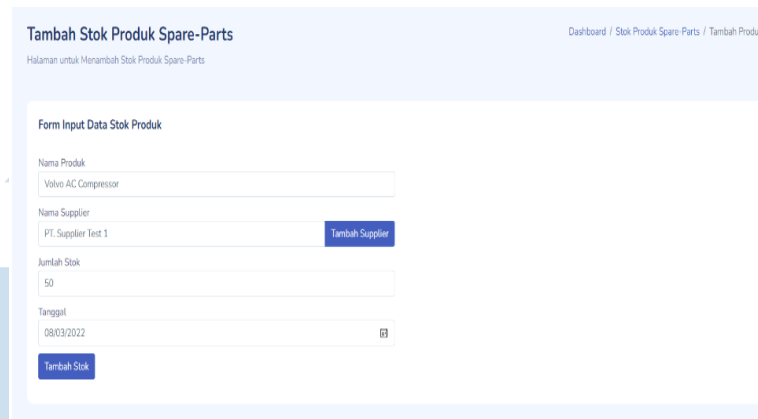
Terakhir adalah fitur untuk menghapus data produk apabila produk memang sudah tidak ingin terdaftar pada sistem, fitur menghapus produk akan menampilkan *modal* dari *Bootstrap 5* untuk mengkonfirmasi apakah ingin benar-benar melakukan penghapusan produk. Berikut adalah tampilan *modal* dapat dilihat pada Gambar 3.32 untuk mengkonfirmasi penghapusan data.



Gambar 3.32 Tampilan *Modal* untuk Menghapus Data Produk

- **Minggu Kedelapan (14 Maret 2022 - 18 Maret 2022)**

Pada minggu kedelapan, telah dilakukannya pembuatan fitur *CRUD* untuk bagian produk masuk (penambahan stok). Apabila terdapat stok barang yang baru masuk ke gudang, administrator dapat menambah data berupa nama barang yang ingin ditambah stoknya, jumlah stok yang masuk, *supplier* yang mengirim stok, serta tanggal stok barang masuk. Berikut adalah *form* yang digunakan untuk menambahkan data penambahan stok barang. Tampilan *form* yang telah dibuat dapat dilihat pada Gambar 3.33



Tambah Stok Produk Spare-Parts Dashboard / Stok Produk Spare-Parts / Tambah Produk

Halaman untuk Menambah Stok Produk Spare-Parts

Form Input Data Stok Produk

Nama Produk
Volvo AC Compressor

Nama Supplier
PT. Supplier Test 1 Tambah Supplier

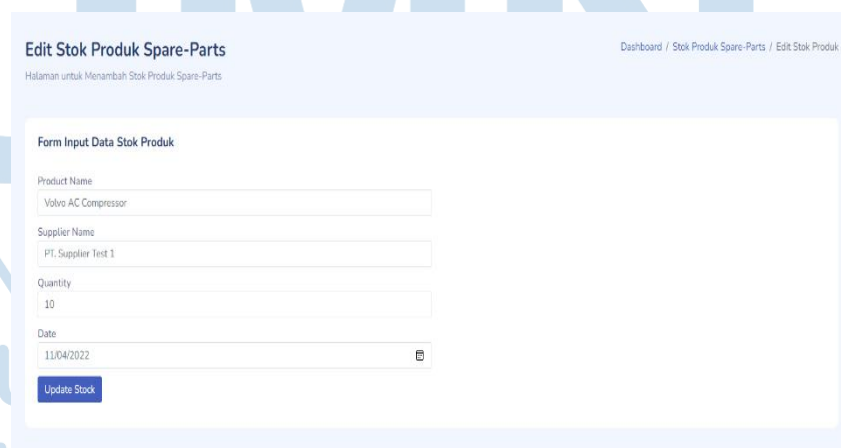
Jumlah Stok
50

Tanggal
08/03/2022

Tambah Stok

Gambar 3.33 Tampilan Form untuk Menambah Data Stok Produk Masuk

Apabila data stok produk masuk telah berhasil ditambahkan, maka stok awal produk akan otomatis ditambahkan dengan stok produk yang masuk. Selanjutnya adalah fitur pembaruan atau *Update/Edit* untuk bagian penambahan stok apabila administrator ingin mengubah atau memperbarui data stok produk masuk yang sudah ada. Fitur ini dapat diakses melalui tombol *Edit* pada kolom *Action* pada halaman Daftar Stok Produk Masuk. Tampilan untuk fitur *Update* dapat dilihat pada Gambar 3.34.



Edit Stok Produk Spare-Parts Dashboard / Stok Produk Spare-Parts / Edit Stok Produk

Halaman untuk Menambah Stok Produk Spare-Parts

Form Input Data Stok Produk

Product Name
Volvo AC Compressor

Supplier Name
PT. Supplier Test 1

Quantity
10

Date
11/04/2022

Update Stock

Gambar 3.34 Tampilan Form untuk Mengubah atau Memperbarui Data Stok Produk Masuk

Setelah itu, telah dilakukannya pembuatan fitur *CRUD* untuk bagian produk yang keluar (pengurangan stok) apabila terjadi penjualan atau retur kepada *supplier*. Administrator dapat menambah data pengurangan stok produk dengan mengisi *form* yang membutuhkan data-data seperti nama produk, jumlah stok yang keluar, harga satuannya, dan tanggal produk keluar. Administrator dapat mengisi kembali harga satuan yang tidak harus sesuai dengan data produk apabila terdapat diskon atau harga dilebihkan, dan jika kasusnya adalah barang yang akan diretur, maka tidak perlu mengisi harga satuan. Berikut adalah tampilan *form* untuk mengisi data stok produk keluar yang dapat dilihat pada Gambar 3.35.



Gambar 3.35 Tampilan Form untuk Menambah Data Stok Produk Keluar

Apabila penambahan data untuk pengeluaran stok produk berhasil dilakukan, maka stok produk akan otomatis dikurangi dengan jumlah yang keluar. Selanjutnya adalah fitur pembaruan atau *Update/Edit* untuk bagian pengeluaran stok apabila administrator ingin mengubah atau memperbarui data stok produk keluar yang sudah ada. Fitur ini dapat diakses melalui tombol *Edit* pada kolom *Action* pada halaman Daftar Stok Produk Masuk. Tampilan untuk fitur *Update* dapat dilihat pada Gambar 3.36.

Gambar 3.36 Tampilan Form untuk Mengubah atau Memperbarui Data Stok Produk Keluar

- **Minggu Kesembilan (21 Maret 2022 - 25 Maret 2022)**

Pada minggu kesembilan, telah dibuatnya dokumen *User Acceptance Test* (UAT) agar sistem nantinya dapat diuji langsung oleh divisi bagian *Inventory* yang nantinya akan menjadi administrator untuk sistem informasi manajemen *inventory* ini. Dokumen *UAT* yang dibuat berbentuk tabel yang berisikan beberapa skenario pengujian. Berikut adalah gambar dokumen *UAT* yang telah dibuat yang dapat dilihat pada Gambar 3.37 dan 3.38.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

USER ACCEPTANCE TEST		PT. ASTRINUSA JAYA DHARMA	
Proyek : Sistem Informasi Manajemen Inventory Produk Spare Parts		Penguji :	Admin Divisi Inventory
		Hasil Pengujian (Berhasil/Gagal)	Tanggal Pengujian
No	Kasus Pengujian		
Skenario Login/Logout dan Mengakses Dashboard			
1	Administrator melakukan Login khusus Administrator di Halaman Login Khusus Administrator		
2	Administrator diarahkan ke halaman Dashboard Administrator untuk Sistem Manajemen Inventory		
3	Administrator melakukan Logout melalui tombol Logout di menu sebelah kiri		
Skenario Manajemen Data Produk			
4	Administrator mengakses halaman manajemen produk melalui menu di sebelah kiri		
5	Administrator menambah data produk baru melalui tombol tambah produk		
6	Administrator mengubah data produk melalui tombol Edit		
7	Administrator menghapus data produk melalui tombol hapus		
Skenario Manajemen Data Kategori			
8	Administrator mengakses halaman manajemen kategori produk melalui menu di sebelah kiri		
9	Administrator menambah data kategori produk baru melalui tombol tambah produk		
10	Administrator mengubah data kategori produk melalui tombol Edit		
11	Administrator menghapus data kategori produk melalui tombol hapus		
Skenario Manajemen Data Merek			
12	Administrator mengakses halaman manajemen merek produk melalui menu di sebelah kiri		
13	Administrator menambah data kategori merek baru melalui tombol tambah produk		
14	Administrator mengubah data merek produk melalui tombol Edit		

Gambar 3.37 Dokumen User Acceptance Test

15	Administrator menghapus data merek produk melalui tombol hapus		
Skenario Manajemen Data Supplier			
16	Administrator mengakses halaman manajemen supplier melalui menu di sebelah kiri		
17	Administrator menambah data supplier baru melalui tombol tambah produk		
18	Administrator mengubah data supplier melalui tombol Edit		
19	Administrator menghapus data supplier melalui tombol hapus		
Skenario Manajemen Data Stok Masuk			
20	Administrator mengakses halaman manajemen tambah stok produk melalui menu di sebelah kiri		
21	Administrator menambah data stok masuk baru melalui tombol tambah produk		
22	Administrator mengubah data stok masuk melalui tombol Edit		
23	Administrator menghapus data stok masuk melalui tombol hapus		
Skenario Manajemen Data Stok Keluar			
24	Administrator mengakses halaman manajemen stok keluar produk melalui menu di sebelah kiri		
25	Administrator menambah data stok keluar baru melalui tombol tambah produk		
26	Administrator mengubah data stok keluar melalui tombol Edit		
27	Administrator menghapus data stok keluar melalui tombol hapus		

Gambar 3.38 Dokumen User Acceptance Test (Lanjutan)

3.2.3 Kendala yang Ditemukan

Saat melakukan pelaksanaan kerja magang pada PT. Astrinusa Jaya Dharma, ditemukannya beberapa kendala seperti berikut :

1. Dikarenakan pelaksanaan praktik kerja magang dilakukan secara daring atau *Work From Home* sepenuhnya, maka proses koordinasi cukup minim dan susah untuk berkomunikasi secara langsung dengan pembimbing lapangan.
2. Pada saat awal pelaksanaan magang masih belum memahami dan mempunyai keahlian dalam menggunakan *Framework Laravel* versi 8.
3. Pelaksanaan kerja magang yang secara individu sehingga mengalami cukup kesulitan ketika menemukan *bug* atau *error* pada saat pembuatan sistem.

3.2.4 Solusi Atas Kendala yang Dihadapi

Dari kendala-kendala yang ditemukan selama proses pelaksanaan magang di PT. Astrinusa Jaya Dharma, maka solusi yang ditemukan dan diterapkan ketika magang adalah sebagai berikut :

1. Koordinasi dengan pembimbing dan pemberian *update* setiap minggunya dilakukan melalui kiriman pesan dalam platform *Whatsapp* dan menggunakan *Zoom Meeting* ketika ingin memperlihatkan fitur-fitur yang telah dibuat.
2. Mencari dan mengikuti tutorial-tutorial dari berbagai sumber internet dan membeli *E-Book* untuk mempelajari tentang syarat-syarat yang diperlukan dalam pembuatan sistem.
3. Mencari di internet terkait *bug* atau *error* yang serupa serta bergabung dengan komunitas seperti PHP Indonesia dan Laravel Indonesia pada platform *Facebook* untuk bertanya ataupun berdiskusi.