

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam pelaksanaan kerja magang di UMKM Mr.Calories, penulis memegang jabatan sebagai Front-end Web Developer, dibawah bimbingan Bapak Ricky Juslim, selaku Owner Mr.Calories. Bapak Ricky berperan sebagai memberikan informasi dan bimbingan dengan baik melalui bimbingan tatap muka maupun via aplikasi pesan singkat *Whatsapp*. Setiap terjadi perubahan dalam *design catalog page* dikonsultasikan dengan pembimbing dan tim projek.

3.2 Tugas yang Dilakukan

Dalam melaksanakan kerja magang di UMKM Mr.Calories, tugas yang dilakukan yaitu mengembangkan *Catalog page* dengan menggunakan framework VueJS yang berfungsi sebagai framework pada *front-end Catalog page*. Pada rancang bangun *Catalog page* ini dibagi berdasarkan fitur yang dibuat, yaitu.

1. Merancang tampilan *wireframe* halaman produk dan tentang Mr.Calories.
2. Membuat tampilan halaman produk – produk yang tersedia pada *Catalog page*.
3. Membuat tampilan halaman tentang UMKM Mr.Calories.
4. Membuat *Create, Read, Update, Delete* pada halaman produk *Admin*.

3.3 Uraian Pelaksanaan Kerja Magang

Proses pelaksanaan kerja magang dibagi menjadi tiga bagian yaitu proses pelaksanaan, kendala yang ditemukan, dan solusi atas kendala yang ditemukan.

3.3.1 Proses Pelaksanaan

Proses pelaksanaan rancang bangun *catalog page* pada UMKM Mr.Calories membutuhkan perangkat lunak dan perangkat keras. Perangkat lunak yang digunakan untuk merancang dan membangun *catalog page* adalah sebagai berikut.

1. Visual Studio Code versi 1.51.1
2. Github versi 2.6.0
3. NPM versi 6.13.4
4. PHP versi 7.4.9
5. XAMPP versi 3.2.4
6. Opera GX Lvl2 Core:71.0.3770.323 64-bit
7. Operating System Windows 10 Home Single Language 64-bit

Perangkat keras yang digunakan untuk merancang dan membangun *catalog page* adalah Laptop Acer Predator AN715-51N dengan spesifikasi sebagai berikut.

1. Intel Core i7-9750H Hexa-core (6Core) 2.60Ghz
2. Nvidia GTX 1660Ti VRAM 6GB GDDR6
3. RAM DDR4 16GB Dual Channel 2666Mhz

Proses pengerjaan berdasarkan tugas dilakukan penulis yaitu, rancangan tampilan *wireframe* halaman produk dan tentang Mr.Calories, tampilan halaman produk, tampilan tentang Mr.Calories, *create,read,update,delete* pada halaman produk *admin*.

A. Wireframe

Rancangan tampilan akan dijelaskan ke dalam *Wireframe*. *Wireframe-wireframe* ini dibuat untuk merancang tampilan halaman produk dan tampilan tentang Mr.Calories.

A.1 Wireframe Halaman Tampilan Produk

Halaman produk adalah halaman yang akan dilihat oleh konsumen atau pengguna, halaman ini bertujuan untuk menampilkan produk-produk yang tersedia di Mr.Calories.



Gambar 3.1 *Wireframe* tampilan halaman produk

Gambar 3.1 merupakan *wireframe* yang menampilkan hasil rancangan pada halaman produk. Dalam tampilan produk tersebut memiliki *navigation bar* yang digunakan untuk mempermudah pengguna untuk pindah ke halaman halaman yang terdapat di *catalog page* Mr.Calories. Setelah *navigation bar*, dibawahnya terdapat *header* yang digunakan untuk menunjukkan pengguna berada di halaman produk.

Pada tampilan produk di bagi ke dalam 4 *section* agar dapat melihat produk apa saja yang dapat di sediakan. *Section* tersebut terdiri dari *section main product*, *section style*, *section topping*, dan *section dressing*. Pada *section* pertama yaitu, *section main product* yang menampilkan produk-produk utama dari Mr.Calories, setiap produk tersebut terdapat gambar produk, nama produk, harga produk, dan deskripsi produk. Pada setiap produk utama di tampilkan berlawanan arah agar konsumen dapat melihat produk dengan mudah.

Pada *section* kedua yaitu, *section style* pada produk Mr.Calories, *section style* ini bertujuan untuk menampilkan tampilan produk yang di inginkan oleh konsumen. *Section* ini menampilkan *subheader*, gambar *style*, nama *style*, dan harga *style*. *Subheader* pada *section style* bertujuan agar dapat mengetahui jenis produk yang ditampilkan. Pada *section style* tampilan gambar produk lebih di utamakan agar dapat melihat dengan jelas *style* apa saja yang ditampilkan oleh Mr.Calories, setelah itu, dibawah tampilan gambar menampilkan nama *style*, dan harga agar diketahui oleh pengguna atau konsumen harga pada setiap *style* tersebut.

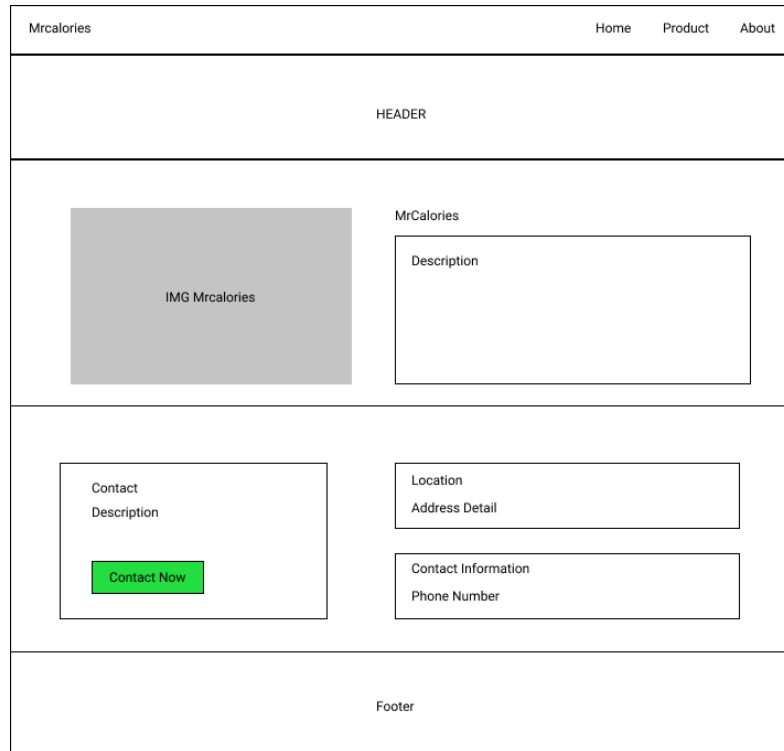
Pada *section* ketiga yaitu, *section topping* yang disediakan oleh Mr.Calories. *Section topping* ini menampilkan tampilan *topping* yang Mr.Calories gunakan pada produk utamanya. *Section ini* menampilkan *subheader topping*, gambar *topping*, nama *topping*, dan harga *topping*. *Subheader* pada *section topping* bertujuan agar

dapat mengetahui jenis produk yang ditampilkan. Gambar *topping* lebih diutamakan agar pengguna atau konsumen dapat melihat bentuk atau *topping* yang akan dipilih. Setelah itu, dibawah gambar *topping* menampilkan nama *topping*, dan harga *topping*.

Pada *section* keempat yaitu, *section dressing* yang menampilkan *dressing-dressing* yang disediakan oleh Mr.Calories. *Section dressing* ini menampilkan *subheader dressing*, gambar *dressing*, nama *dressing*, dan harga *dressing*. *Subheader dressing* diletakkan dibawah *section topping* agar mengetahui bagian yang ditampilkan adalah bagian *dressing*. Setelah itu, dibawah *subheader* menampilkan gambar *dressing* untuk memperlihatkan *dressing-dressing* yang tersedia. Dibawah gambar *dressing* menampilkan nama *dressing* dan harga *dressing*.

A.2 Wireframe Halaman Tampilan Tentang Mr.Calories

Halaman tentang Mr.Calories merupakan halaman yang berisikan gambar *offline store* Mr.Calories agar konsumen mengetahui bentuk *offline store* Mr.Calories, deskripsi Mr.Calories agar menjelaskan awal berdirinya Mr.Calories, kontak atau nomor telepon dari Mr.Calories agar konsumen dapat menanyakan informasi lebih lanjut, dan Alamat Mr.Calories agar konsumen mengetahui lokasi *offline store* berada.



Gambar 3.2 *Wireframe* tampilan halaman tentang Mr.Calories

Gambar 3.2 merupakan *wireframe* yang menampilkan hasil rancangan halaman tentang Mr.Calories. Pada *wireframe* ini terdapat *navigation bar*, *header*, *main section*. *Navigation bar* diletakkan paling atas layar, agar mempermudah pengguna / konsumen akses halaman lain. *Navigation bar* menampilkan nama Mr.Calories dibagian kiri agar terpisah dari navigasi untuk ke halaman lain, dan navigasi ke *home*, *product*, dan *about*. *Header* diletakkan dibawah *navigation bar* agar mempermudah pengguna atau konsumen mengetahui halaman yang sedang di akses adalah halaman *About*.

Pada *main section* menampilkan gambar Mr.Calories, nama Mr.Calories, deskripsi Mr.Calories, deskripsi kontak, alamat, nomor telepon. Gambar Mr.Calories diletakkan pada bagian kiri agar terlihat dengan jelas, dan disamping gambar tersebut ada nama Mr.Calories dan deskripsi Mr.Calories yang menjelaskan tentang Mr.Calories. Pada bagian kontak terdapat deskripsi tentang pertanyaan

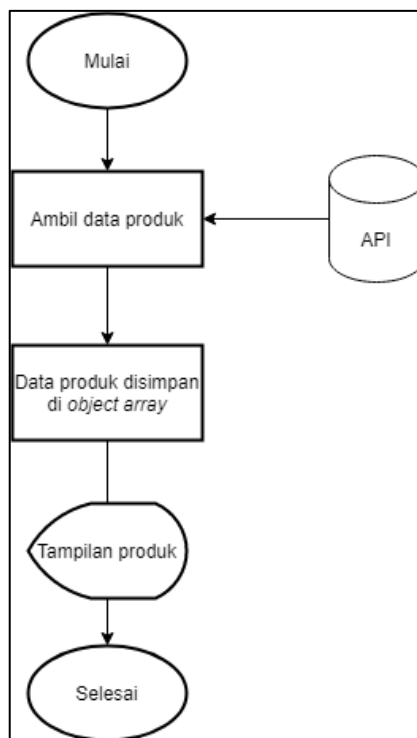
lebih lanjut, dan dibawah deskripsi terdapat tombol “Contact Now”, jika ditekan oleh pengguna atau user dapat langsung bertanya kepada *Owner Mr.Calories* via *Whatsapp*. Pada bagian lokasi menjelaskan lokasi *stand Mr.Calories*, dan bagian kontak informasi, ditampilkan nomor telepon Mr.Calories jika ingin informasi lebih lanjut.

B. Flowchart

Alur kerja catalog page akan dijelaskan ke dalam *flowchart*. *Flowchart* – *flowchart* ini dibuat sesuai dengan cara kerja pada halaman masing – masing.

B.1 Flowchart Tampilan Halaman Produk

Gambar 3.3 merupakan *flowchart* tampilan halaman produk yang akan mengambil data produk dari *API* yang sudah di sediakan.

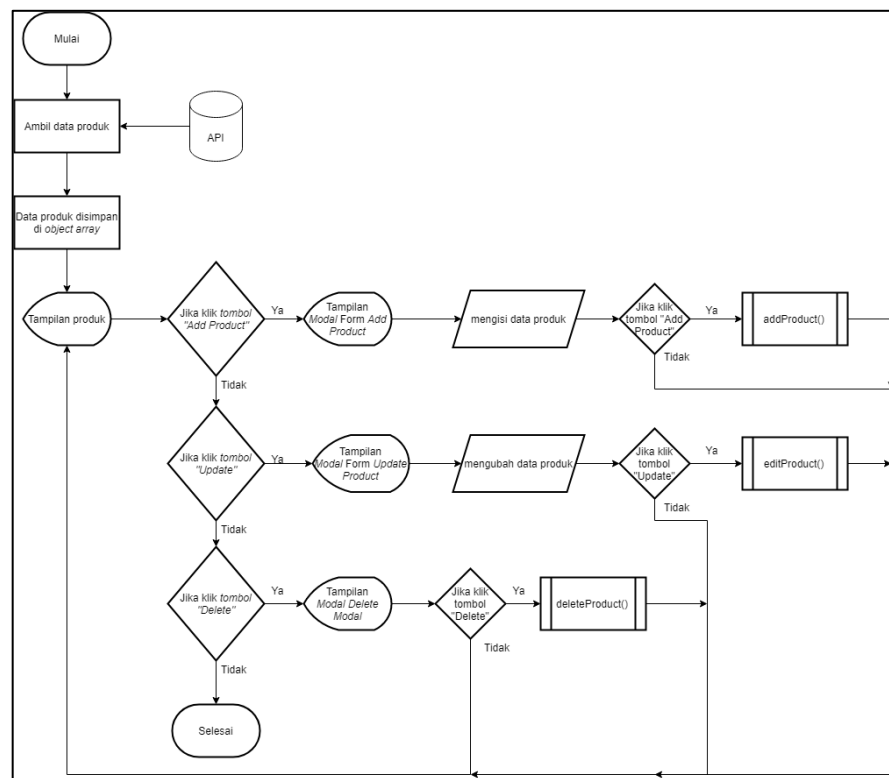


Gambar 3.3 Flowchart tampilan halaman produk

Data produk yang sudah di dapat oleh API akan dimasukkan ke dalam *object array* dan menampilkan tampilan produk yang berisi produk-produk di *object array* sebelumnya.

B.2 Flowchart Tampilan Halaman Produk Admin

Flowchart halaman produk pada gambar 3.3 saling terhubung dikarenakan data produk yang diambil melalui API yang telah dimasukan datanya dari halaman produk admin.



Gambar 3.4 *Flowchart* tampilan halaman produk admin

Gambar 3.4 merupakan *flowchart* pada tampilan halaman produk admin. Sebelum menampilkan tampilan produk, melakukan pengambilan data produk dari *API*. Setelah itu memasukkan data yang sudah di dapat kedalam *object array* dan

menampilkan tampilan yang sudah berisi produk – produk. Pada *flowchart* ini terdapat 3 tombol yaitu *add product*, *update*, *delete*.

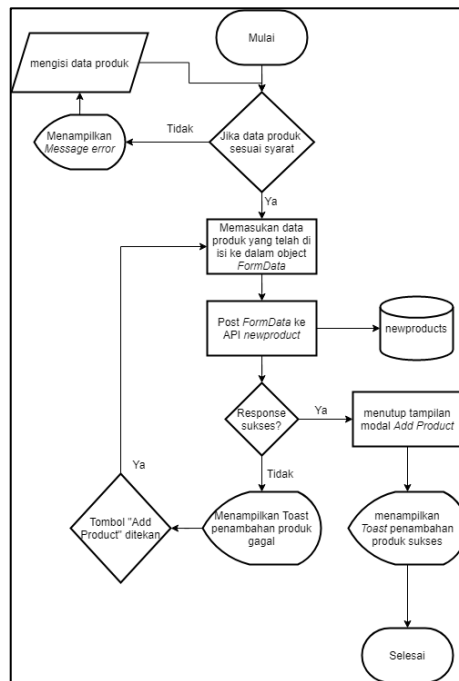
Pengguna dapat menambah produk dengan menekan tombol *add product* akan memunculkan tampilan *modal form add product*, dan mengisi data yang sesuai dengan *form* tersebut. Setelah selesai mengisi *form*, jika pengguna menekan tombol *Add Product* akan memanggil modul *addProduct()*, dan jika menekan tombol *cancel* akan menutup *modal form add product* dan kembali ke tampilan produk admin.

Pengguna dapat mengubah produk yang sudah di masukkan ke dalam *database* sebelumnya dengan menekan tombol *update*. Tombol *update* tersebut akan menampilkan *modal form update product* yang setiap isi form sudah menampilkan data sebelumnya. Setelah pengguna selesai mengisi *form*, pengguna dapat menekan tombol *update product* untuk mengubah data produk yang sesuai dengan pengguna inginkan. Pengguna dapat membatalkan perubahan dengan menekan tombol *cancel* akan menutup tampilan *modal form update product* dan kembali ke tampilan halaman produk admin.

Pengguna dapat menghapus data produk yang tidak diinginkan dengan menekan tombol *Delete* akan menampilkan *modal delete* yang berisi 2 tombol yaitu, *delete* dan *cancel*. Pengguna dapat menekan tombol *delete* untuk menghapus data produk. Pengguna dapat membatalkan penghapusan data produk dengan menekan tombol *cancel* yang akan menutup tampilan *modal form delete product* dan kembali ke tampilan halaman produk admin.

B.3 Flowchart Create Produk Admin

Jika pengguna ingin menambahkan produk dari halaman produk admin pada gambar 3.4, akan terhubung dengan *flowchart create* produk agar data produk yang ingin ditambahkan dapat disimpan didalam API.

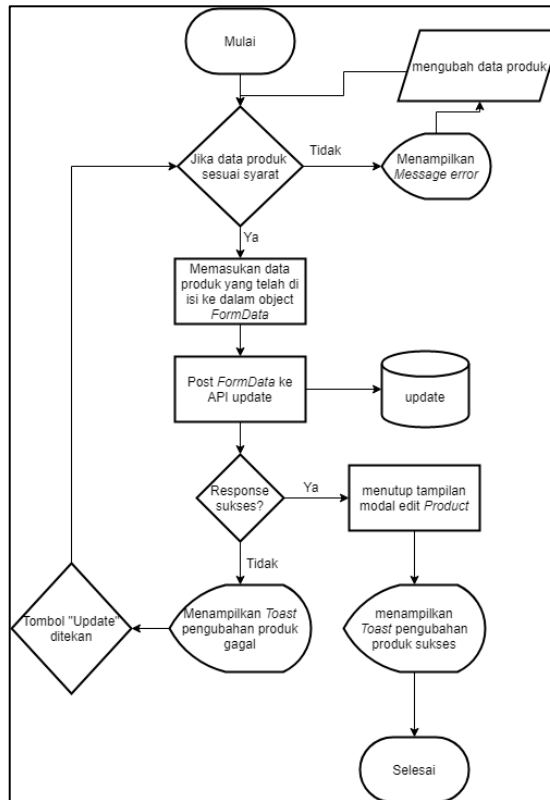


Gambar 3.5 *Flowchart create* produk admin

Gambar 3.5 merupakan *flowchart create* produk admin, pengguna telah mengisi *form addmodal*, kemudian data tersebut akan di cek apakah sesuai dengan syarat. Setelah data sesuai dengan syarat akan memasukkan data ke dalam *FormData* dan melakukan *Http request POST* ke API *newproduct* dan akan mengirim kembali *response*. *Response* tersebut akan di cek apakah sukses atau tidak, jika sukses akan menampilkan *Toast* penambahan produk sukses, dan jika tidak akan menampilkan *Toast* penambahan produk gagal.

B.4 Flowchart Edit Produk Admin

Pengguna dapat melakukan perubahan data pada produk dengan menekan tombol *update* yang ada di gambar 3.4, saat pengguna menekan tombol *update* akan melakukan alur kerja pada gambar 3.6.

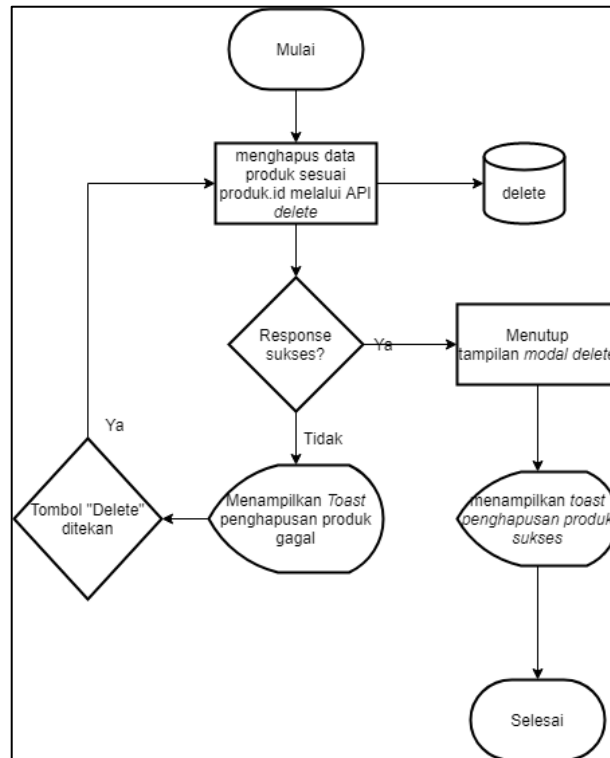


Gambar 3.6 Flowchart *edit* produk admin

Gambar 3.6 merupakan *flowchart edit* produk admin, pengguna telah mengubah data produk pada *form update modal*, dan melakukan pengecekan pada data apakah sesuai dengan syarat. Setelah sesuai dengan syarat akan di masukkan ke dalam *FormData* dan melakukan *Http Request POST* ke *API update*. *Http Request* tersebut akan mengirimkan sebuah *response*, setelah itu melakukan pengecekan apakah sukses atau tidak. Jika *response* sukses akan menutup tampilan *modal editproduct* dan menampilkan *toast* pengubahan produk sukses. Jika *response* gagal akan menampilkan *toast* pengubahan produk gagal.

B.5 Flowchart Delete Produk Admin

klien dapat melakukan penghapusan produk dengan menekan tombol *delete* pada gambar 3.4, jika *delete* ditekan akan memanggil alur pada gambar 3.7.



Gambar 3.7 *Flowchart delete* produk admin

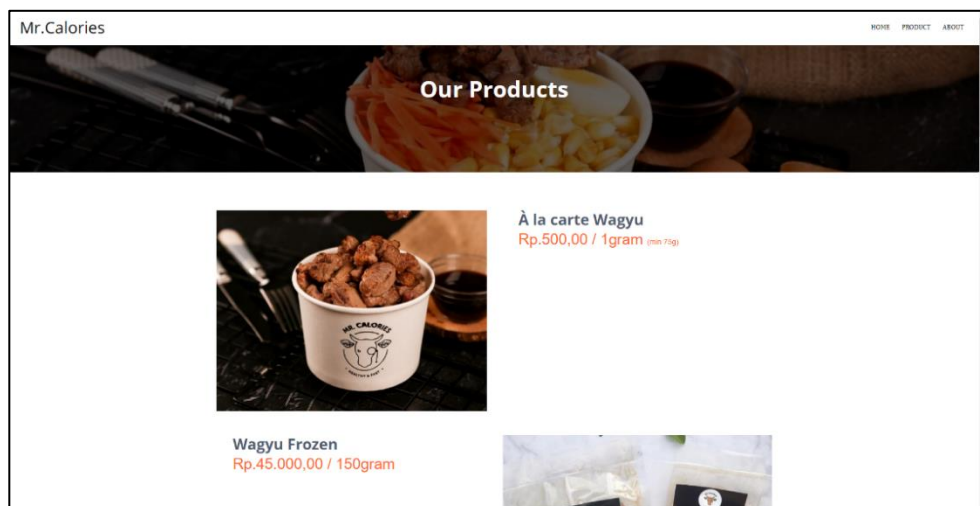
Gambar 3.7 merupakan *flowchart delete* produk, pada saat pengguna menekan tombol *Delete* pada *modal delete* akan menghapus data produk sesuai dengan *id* produk dan melakukan *Http Request Delete* ke API *delete*. *Http Request* akan mengirimkan sebuah *response*, setelah mendapat *response* akan di cek apakah sukses atau tidak. Jika *response* sukses akan menutup *modal delete* dan menampilkan *Toast* penghapusan produk sukses. Jika *response* gagal akan menampilkan *Toast* penghapusan produk gagal.

C. Implementasi

Berikut adalah hasil dari implementasi *catalog page* untuk Mr.Calories. Implementasi ini dibuat sesuai dengan rancangan dari *flowchart* dan *wireframe*.

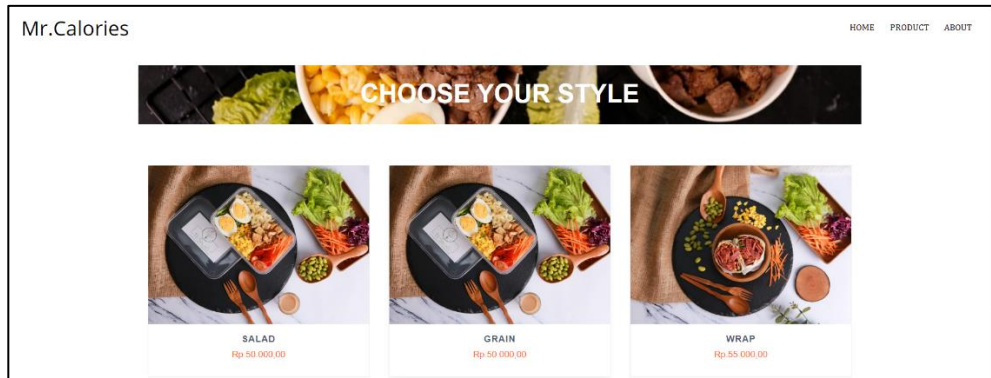
C.1 Implementasi Halaman Produk

Pada implementasi halaman produk berisikan tampilan produk-produk yang disediakan oleh Mr.Calories. Implementasi halaman produk ini berisikan 4 *section* yaitu, *section main product*, *section style*, *section topping*, dan *section dressing*.



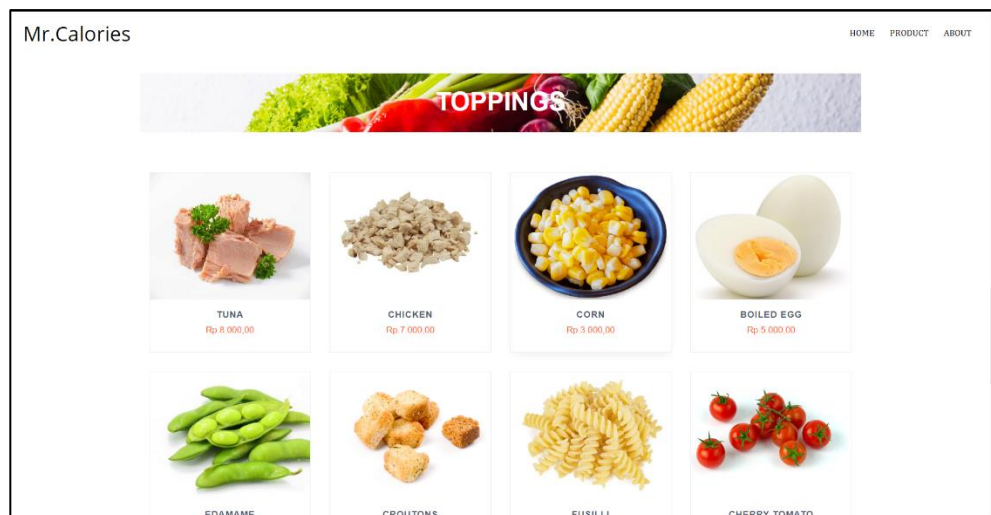
Gambar 3.8.1 Implementasi *section main product*

Gambar 3.8.1 merupakan implementasi *section main product* pada halaman produk. Pada *section main product* berisikan tampilan gambar produk, nama produk, deskripsi produk, dan harga produk. Tampilan setiap produk sengaja ditampilkan dengan berlawanan arah, agar dapat memudahkan pengguna atau konsumen melihat produk.



Gambar 3.8.2 Implementasi *section style*

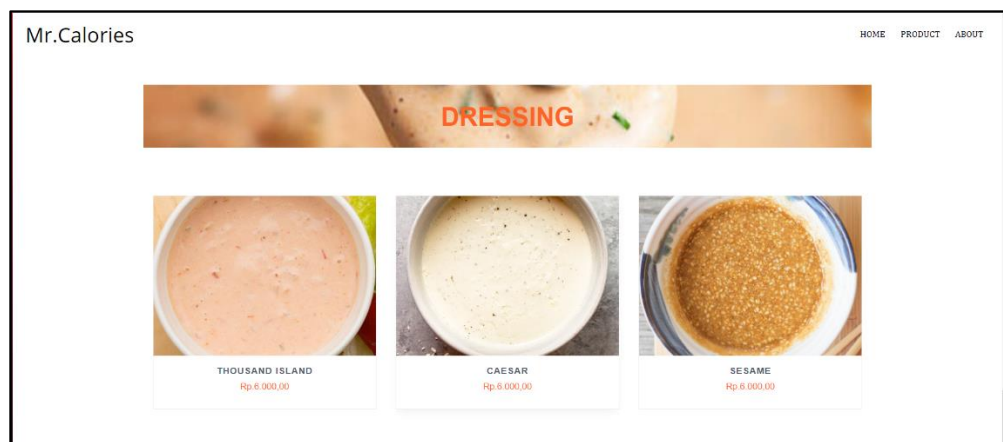
Gambar 3.8.2 merupakan implementasi *section style* yang berisikan *subheader*, gambar *style*, nama *style*, dan harga *style*. *Subheader* diletakkan sebelum produk *style* agar pengguna atau konsumen dapat mengetahui tampilan produk dibawah *subheader* tersebut. Gambar *style* dibuat lebih besar dari tampilan nama dan harga agar mempermudah pengguna atau user melihat *style* yang berbeda-beda dari Mr. Calories.



Gambar 3.8.3 Implementasi *section topping*

Gambar 3.8.3 merupakan implementasi dari *section topping* yang menampilkan produk *topping-topping* yang disediakan Mr. Calories. *Section topping* ini berisikan *subheader*, gambar *topping*, nama *topping*, dan harga *topping*.

Subheader diletakkan sebelum tampilan produk *topping* agar mempermudah mengetahui jenis produk dibawahnya. Setiap *topping* dibuat dalam 1 kotak atau kolom dan diberikan sedikit *shadow* pada kolom tersebut agar pengguna atau konsumen dapat melihat bagian dari setiap produk *topping*.

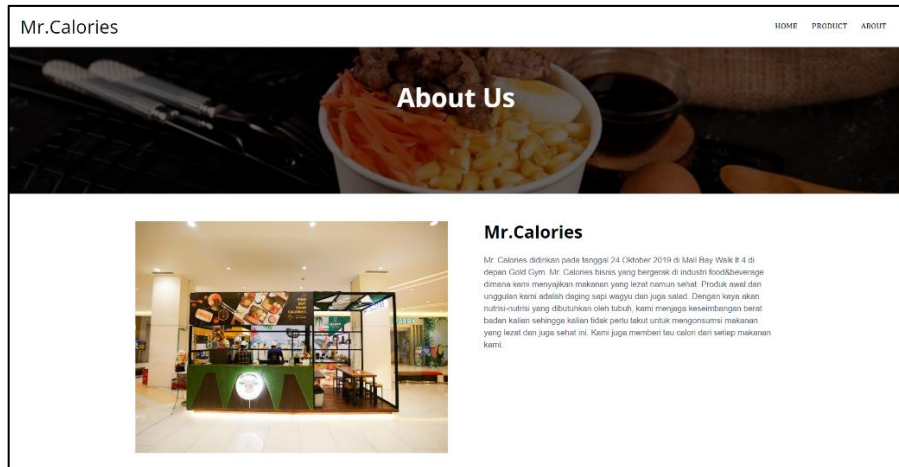


Gambar 3.8.4 Implementasi *section dressing*

Gambar 3.8.4 merupakan implementasi *section dressing*, *section dressing* menampilkan produk yang tersedia dari Mr. Calories. *Section dressing* berisi *subheader*, gambar *dressing*, nama *dressing*, dan harga *dressing*. *Subheader* diletakkan sebelum tampilan produk *dressing* agar mengetahui jenis produk yang ditampilkan. Gambar *Dressing* lebih besar daripada nama dan harga *dressing*, agar pengguna atau konsumen dapat melihat lebih jelas gambar produk.

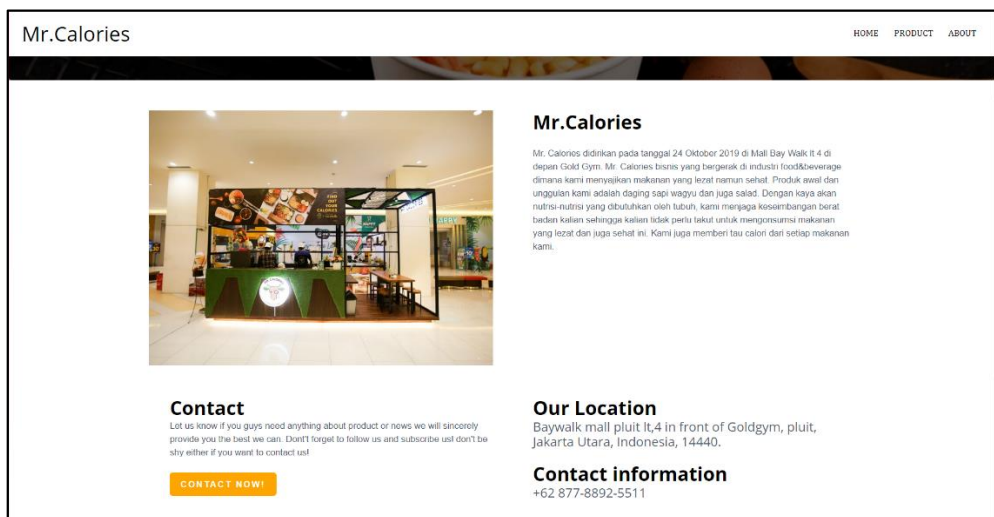
C.2 Implementasi Halaman Tentang Mr. Calories

Halaman tentang Mr. Calories bertujuan untuk memberikan informasi tentang Mr. Calories agar pengguna atau konsumen dapat mengetahui latar belakang sejarah singkat Mr. Calories.



Gambar 3.9 Implementasi *header* halaman tentang Mr.Calories

Gambar 3.9 merupakan implementasi *header* pada halaman tentang Mr.Calories. Tampilan *header* tersebut di letakan paling atas untuk mempermudah pengguna atau konsumen mengetahui halaman yang sedang di akses. Tampilan *Background header* menggunakan salah satu produk Mr.Calories agar *header* tidak terlalu polos.



Gambar 3.10 Implementasi *section main*

Gambar 3.10 merupakan implementasi *section main* yang berisi gambar Mr.Calories, nama Mr.Calories, deskripsi, kontak, lokasi, nomor telepon. Gambar Mr.Calories diletakan di bagian kiri agar terlihat lebih besar, dan lebih jelas. Nama

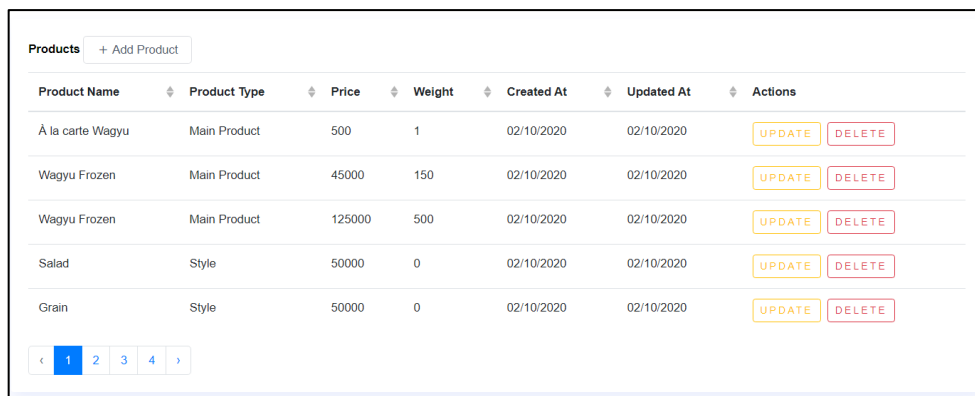
Mr.Calories dan deksripsi diletakan bagian kanan gambar. Pada bagian *Contact* terdapat deskripsi dan tombol *Contact Now* untuk melakukan *chat* secara langsung dengan *Owner* Mr.Calories via *Whatsapp*. Bagian *our location* terdapat alamat lengkap dari *stand* Mr.Calories, dan bagian *contact information* terdapat nomor telepon dari *Owner* Mr.Calories jika ingin menanyakan informasi lebih lanjut.

C.3 Implementasi Produk Admin

Klien dapat melakukan *maintenance* pada data produk seperti *add product*, *update product*, *delete product*, dan *read product* pada hasil implementasi halaman produk admin. Hasil implementasi ini sesuai dengan *flowchart*.

C.3.1 Implementasi Produk Tabel

Implementasi dibawah ini adalah tampilan pertama saat klien mengakses halaman produk admin, halaman produk admin ini menampilkan data yang telah di masukan kedalam API.



Product Name	Product Type	Price	Weight	Created At	Updated At	Actions
À la carte Wagyu	Main Product	500	1	02/10/2020	02/10/2020	<button>UPDATE</button> <button>DELETE</button>
Wagyu Frozen	Main Product	45000	150	02/10/2020	02/10/2020	<button>UPDATE</button> <button>DELETE</button>
Wagyu Frozen	Main Product	125000	500	02/10/2020	02/10/2020	<button>UPDATE</button> <button>DELETE</button>
Salad	Style	50000	0	02/10/2020	02/10/2020	<button>UPDATE</button> <button>DELETE</button>
Grain	Style	50000	0	02/10/2020	02/10/2020	<button>UPDATE</button> <button>DELETE</button>

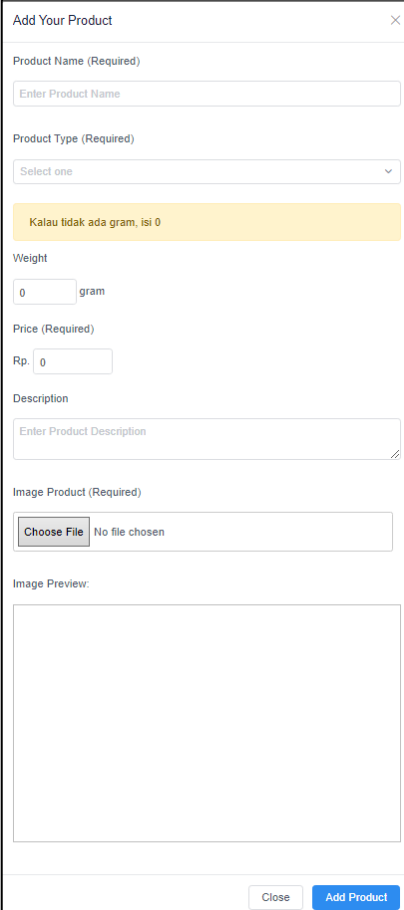
Gambar 3.11 Implementasi produk tabel

Gambar 3.11 merupakan implementasi produk tabel, pada tampilan produk tabel ini terdapat *pagination bar*, dan 3 tombol yaitu, *add product*, *update*, *delete*.

Dalam tabel produk memiliki 7 kolom yaitu, *product name*, *product type*, *price*, *weight*, *createdAt*, *updatedAt*, dan *Action*. Setiap kolom dalam melakukan pengurutan secara *ascending* dan *descending*.

C.3.2 Implementasi Modal Form Add Produk

Implementasi *modal form* dibawah ini ditampilkan saat klien ingin menambah produk. *Form* ini diisi dengan data produk yang ingin ditambahkan dan data produk tersebut dimasukan kedalam API agar dapat ditampilkan di Halaman produk.



The image shows a modal form titled "Add Your Product" with a close button (X) in the top right corner. The form contains the following fields and elements:

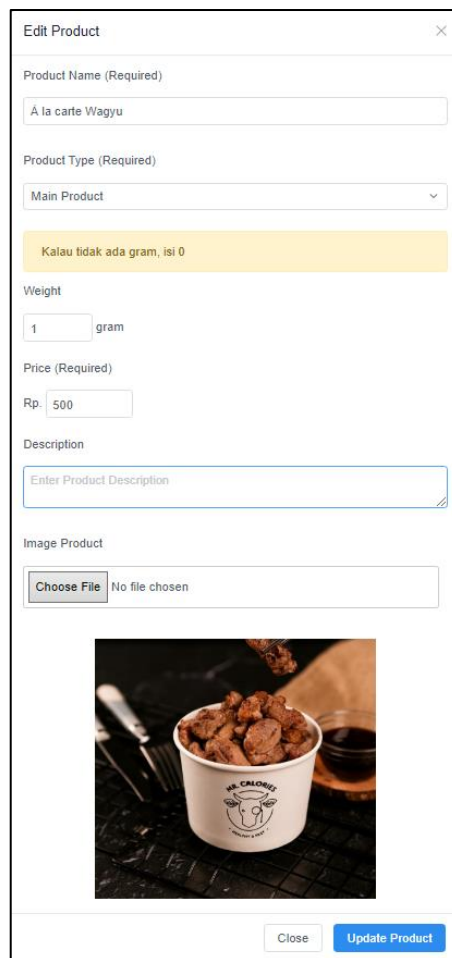
- Product Name (Required):** A text input field with the placeholder "Enter Product Name".
- Product Type (Required):** A dropdown menu with "Select one" and a downward arrow.
- Weight:** A text input field with "0" and the unit "gram".
- Price (Required):** A text input field with "Rp. 0".
- Description:** A text area with the placeholder "Enter Product Description".
- Image Product (Required):** A file upload field with a "Choose File" button and the text "No file chosen".
- Image Preview:** A large empty rectangular area for displaying the selected image.
- Buttons:** "Close" and "Add Product" buttons at the bottom right.

Gambar 3.12 Implementasi *modal form add* produk

Gambar 3.12 merupakan implementasi *modal form add* produk. Pada *modal form* pengguna dapat mengisi data produk sesuai *label form*. Pengguna harus menambahkan gambar untuk menambahkan produk. Pengguna dapat mengetahui gambar yang dipilih benar atau tidak, karena dalam *modal form* menampilkan *image preview*.

C.3.3 Implementasi Modal Edit Produk

Klien dapat mengubah data produk yang sebelumnya sudah data produk sudah ditambahkan dan terjadi kesalahan input dengan menekan tombol *update* yang dapat dilihat gambar 3.11.



The image shows a modal window titled "Edit Product" with a close button in the top right corner. The form contains the following elements:

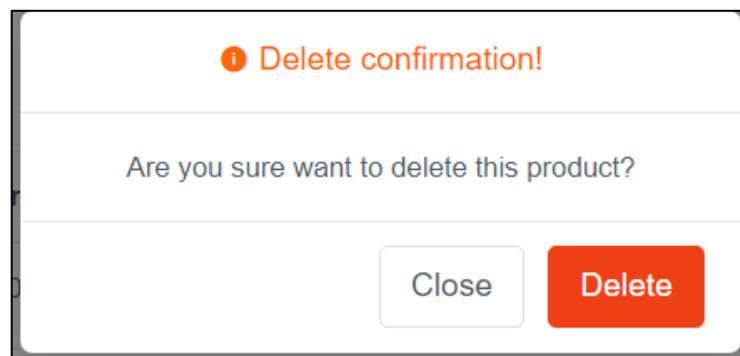
- Product Name (Required):** A text input field containing "A la carte Wagyu".
- Product Type (Required):** A dropdown menu currently showing "Main Product".
- Warning:** A yellow highlighted box with the text "Kalau tidak ada gram, isi 0".
- Weight:** A text input field containing "1" followed by "gram".
- Price (Required):** A text input field containing "Rp. 500".
- Description:** A text area with the placeholder "Enter Product Description".
- Image Product:** A section with a "Choose File" button and the text "No file chosen".
- Image Preview:** A photograph of a white cup filled with fried food, with a logo that says "MR. CALORIE".
- Buttons:** "Close" and "Update Product" buttons at the bottom right.

Gambar 3.13 Implementasi *modal form edit* produk

Gambar 3.13 merupakan implementasi *modal form edit* produk. Untuk menampilkan *Modal form edit* pengguna harus menekan tombol *update* pada data produk yang ingin di ubah. Saat *modal form edit* muncul, *form* telah di isi oleh data produk dan tampilan gambar produk yang digunakan sebelumnya.

C.3.4 Implementasi Modal Delete

Klien dapat menghapus data produk yang tidak diinginkan, pada gambar 3.11 terdapat tombol *delete* yang akan menampilkan *modal delete* seperti dibawah ini.



Gambar 3.14 Implementasi *modal delete* produk

Gambar 3.14 merupakan implementasi *modal delete* produk, pengguna dapat menekan tombol *delete* pada produk yang ingin di hapus. Setelah menekan tombol *delete*, *modal delete* akan muncul untuk meminta pengguna memastikan sekali lagi sebelum data produk dihapus. Jika tombol *close* ditekan akan menutup *modal delete* produk dan menampilkan produk tabel kembali.

3.3.2 Kendala yang Ditemukan

Dalam proses pelaksanaan kerja magang, terdapat beberapa kendala yang dihadapi. Kendala-kendala yang ditemukan adalah antara lain sebagai berikut:

- Kurangnya sumber pembelajaran (dokumentasi, video *tutorial*) tentang *framework* VueJS yang digunakan.
- Kurangnya data yang dibutuhkan saat proses pembuatan *catalog page*.

3.3.3 Solusi Atas Kendala yang Ditemukan

Dari kendala-kendala yang ditemukan selama proses kerja magang, terdapat beberapa solusi yang ditemukan untuk menyelesaikan kendala tersebut agar tidak menghambat pengerjaan tugas kerja magang. Solusi dari kendala-kendala yang ditemukan adalah sebagai berikut:

- Membaca forum tentang *framework* VueJS yang digunakan.
- Melakukan pencarian gambar yang digunakan sebagai pengganti sementara.