

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam pelaksanaan kerja magang di CV Guna Motor penulis berkedudukan sebagai *web application developer intern* (IT *intern*). Pekerjaan magang diberikan dan diarahkan oleh Bapak Herman selaku IT *Manager* dalam perusahaan. Dalam perencanaan kerja magang ini ditetapkan estimasi penyelesaian selama kurang lebih 2 bulan. Dalam *project* dibimbing dan diarahkan oleh Bapak Herman selaku IT *Manager* selama dalam masa *development*.

Progress/perkembangan daripada *project* ini setiap harinya akan dilaporkan kepada Bapak Herman apabila terjadi ketidak sesuaian ataupun kesalahan dapat langsung diperbaiki. Proses *development* terjadi seterusnya sampai *project* ini selesai dan sesuai dengan yang diinginkan. Dalam *development project* ini tidak lepas daripada revisi, dan ketika dalam tahap *development* dapat dilakukan revisi sehingga banyak terjadi perubahan-perubahan pada fitur dan desain daripada *project* ini.

3.2 Tugas yang Dilakukan

Pekerjaan yang dilakukan adalah merancang dan membangun *information system* yang berbasis *web application* untuk CV Guna Motor (divisi Toko Model). Dalam sistem ini terdapat berbagai fitur, diantaranya *login sistem*, *product*, artikel, *banner*, dan CMS (*Content Management System*).

Tugas-tugas yang diberikan dan dikerjakan memiliki fungsi masing-masing, yaitu sebagai berikut:

- a. Modul *login* sistem, berfungsi untuk memberikan akses masuk sebagai admin untuk melakukan akses ke dalam CMS, untuk melakukan *maintain* tampilan daripada sistem,
- b. Modul *product*, dapat melakukan perubahan terhadap data *product*, baik itu penambahan *product* baru, perubahan *product* yang sudah ada, maupun menghapus *product* yang sudah ada,
- c. Modul artikel, sama seperti modul *product*, hanya berbeda tujuan. Modul ini bertujuan/berfungsi untuk melakukan perubahan terhadap data artikel,
- d. Modul *banner*, berfungsi untuk memilih dan mengunggah gambar *banner* yang akan ditampilkan dalam laman utama *web application*,
- e. Modul CMS, berfungsi untuk melakukan *maintain* daripada *web application* oleh admin.

3.3 Uraian Pelaksanaan Kerja Magang

Tabel 3.1 Deskripsi Tugas Per Minggu

Minggu ke	Deskripsi
1	Melakukan adaptasi dengan lingkungan kerja, melakukan perancangan awal daripada sistem, serta melakukan perancangan DFD (<i>Data Flow Diagram</i>), ERD (<i>Entity Relationship Diagram</i>), dan analisa struktur tabel.
2	Melakukan perancangan awal <i>User Interface</i> sebagai tampilan daripada sistem, serta melakukan implementasi awal pembuatan rancangan.
3	Melakukan <i>Coding</i> PHP dengan Laravel 4.2.
4	Melakukan <i>Coding</i> PHP dengan Laravel 4.2.
5	Melakukan <i>Coding</i> PHP dengan Laravel 4.2.
6	Melakukan <i>Coding</i> PHP dengan Laravel 4.2.

Tabel 3.1 Deskripsi Tugas Per Minggu (lanjutan)

Minggu ke	Deskripsi
7	Melakukan <i>Coding</i> PHP dengan Laravel 4.2.
8	Tahapan <i>testing</i> dan <i>maintenance</i> .
9	Tahapan <i>testing</i> dan <i>maintenance</i> .

Uraian kerja magang dapat dilihat pada Tabel 3.1. Pada minggu pertama dilakukan pengenalan dan adaptasi dengan lingkungan kerja terlebih dahulu, kemudian di minggu kedua merupakan tahap perancangan awal daripada *user interface* sebagai tampilan daripada sistem, serta melakukan implementasi awal pembuatan rancangan, serta melakukan perancangan DFD (*Data Flow Diagram*) yang merupakan alur data daripada sistem, serta dilakukan analisa struktur tabel serta ERD (*Entity Relationship Diagram*) untuk kebutuhan penyimpanan data daripada sistem. Minggu ketiga hingga minggu ketujuh merupakan tahapan implementasi daripada sistem, dalam tahapan ini dilakukan *coding* atau proses pemrograman sistem. Kemudian pada tahapan akhir, yaitu minggu kedelapan dan minggu kesembilan dilakukan *testing* dan *maintenance* yang bertujuan menemukan *error* serta *bug* yang ada dalam sistem kemudian memperbaikinya dalam tahapan *maintenance* sistem.

3.3.1 Proses Pelaksanaan

Sistem yang dibangun ini dikembangkan dengan menggunakan *framework* Laravel 4.2 menggunakan bahasa PHP dan untuk keperluan penyimpanan data menggunakan MySQL sebagai *database*. Proses pembuatan proyek diawali dengan pengenalan dan pemahaman terlebih dahulu dengan *framework* Laravel 4.2,

sehingga ketika memasuki proses pembangunan sistem dengan menggunakan *framework* ini dapat berjalan dengan lebih baik dan lancar.

A. Perancangan Sistem

Perancangan sistem di dalam penyusunan kerja magang terbagi menjadi 3 (tiga) tahapan. Tahapan-tahapan tersebut dijelaskan lebih lanjut pada poin-poin di bawah ini.

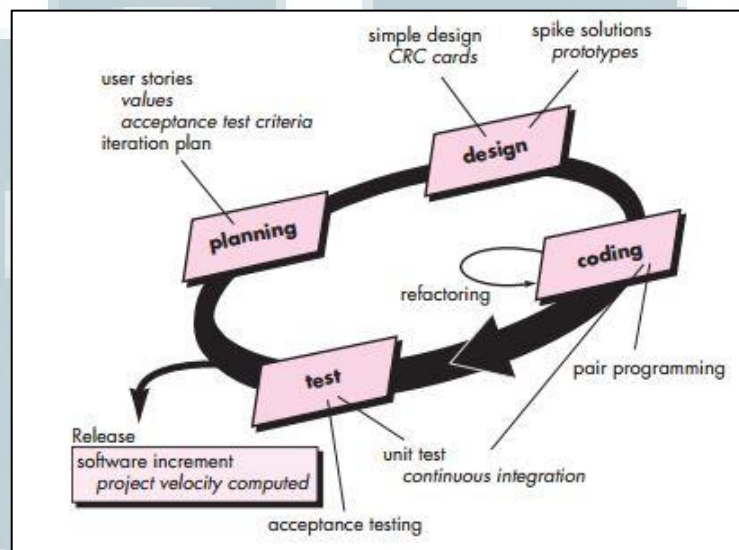
A.1 Metode Pengembangan Program

Metode pengembangan program yang digunakan dalam pembangunan sistem ini adalah *agile method*. Proses pembangunan program diawali dengan tahapan *planning* atau perencanaan, dengan mengetahui kebutuhan dalam pembangunan sistem, kemudian masuk ke dalam tahapan *design*, yaitu membuat suatu rancangan awal daripada sistem, dilanjutkan dengan *coding* yaitu tahapan pemrograman dan merupakan tahapan inti, yang dilanjutkan dengan tahap *testing*. Metode *agile* sendiri memungkinkan prosesnya untuk melakukan iterasi, jika terjadi perubahan dapat disesuaikan dalam pembangunan sistem.

Beberapa kelebihan dari metode *agile* yang cukup menjadi alasan untuk digunakan adalah proses pengembangan sistem yang relatif cepat, mengurangi resiko kegagalan dalam implementasi perangkat lunak dari segi non teknis. Apabila terjadi suatu kegagalan saat pengembangan sistem, kerugian dari segi materi relatif kecil (Pressman, 2005).

Iterasi berjalan pada 1 minggu pengerjaan sistem, kemudian dilakukan pengecekan terhadap sistem yang telah dikembangkan, apakah kriteria dan

kebutuhan sudah tercapai atau belum. Jika kriteria dan kebutuhan sudah terpenuhi, maka proses pengembangan akan dilanjutkan, seandainya belum terpenuhi maka dilakukan iterasi kedua sampai kriteria dan kebutuhan tercapai, dan seterusnya. Dalam implementasi kerja magang proses iterasi dilakukan dengan diawali proses perancangan sistem sesuai dengan arahan IT *Manager* sampai dengan *testing* dan *maintenance*. Hal ini digambarkan pada gambar 3.1.

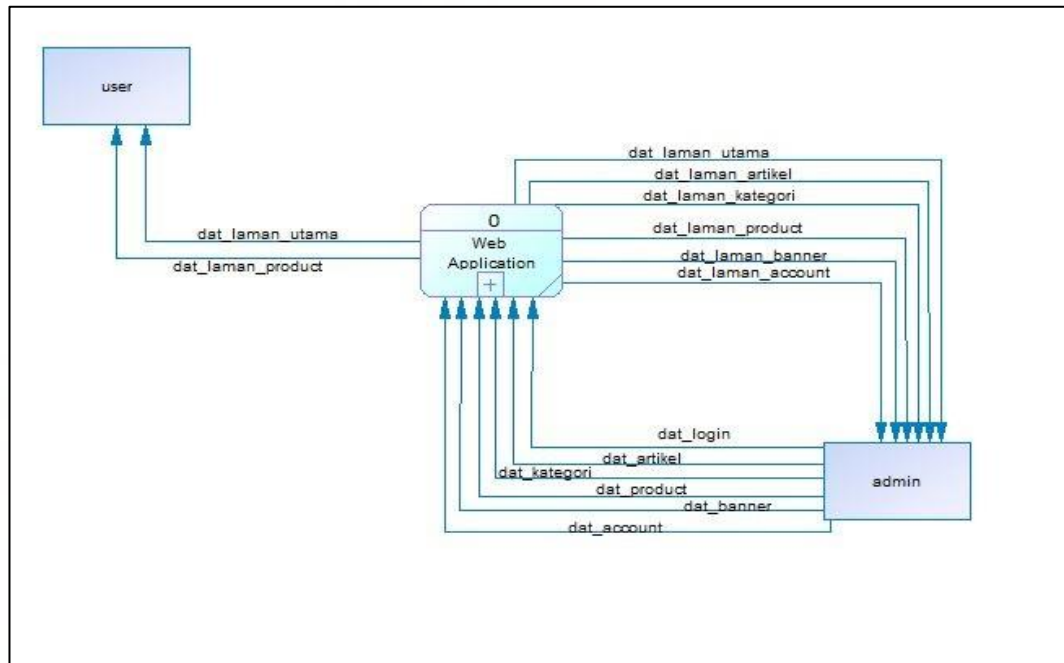


Gambar 3.1 Agile Method (Sumber: Pressman, 2005)

A.2 Data Flow Diagram

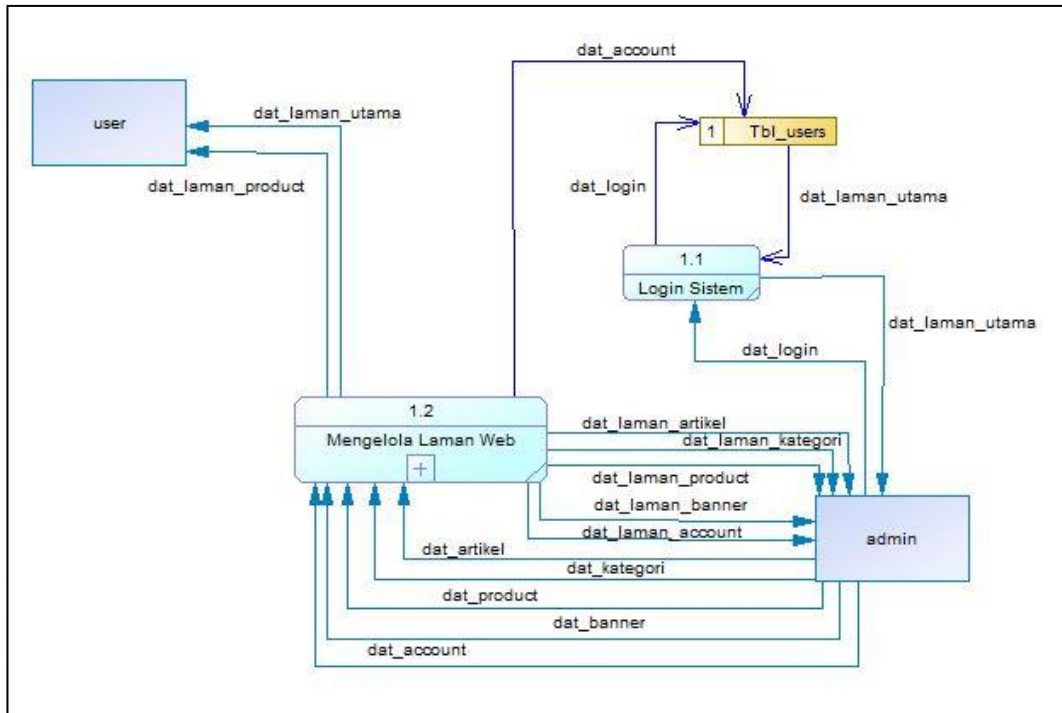
DFD (*Data Flow Diagram*) merupakan suatu alat perancangan model yang memungkinkan *designer* untuk menggambarkan suatu sistem sebagai suatu jaringan proses yang saling berhubungan satu sama lain lengkap dengan alur datanya. DFD terbagi menjadi 3 bagian, yaitu diagram konteks, merupakan diagram paling sederhana, karena menggambarkan sistem secara garis besar yang dapat mewakili seluruh proses yang terdapat dalam sebuah sistem serta diagram konteks merupakan tingkat tertinggi dalam DFD, diagram selanjutnya adalah diagram 0 merupakan suatu lingkaran besar yang mewakili lingkaran kecil yang

terdapat di dalamnya. Diagram 0 merupakan perpecahan daripada diagram konteks, pada diagram ini sudah memuat penyimpanan data, serta yang terakhir adalah diagram rinci yang menjelaskan proses apa yang terdapat pada diagram 0.

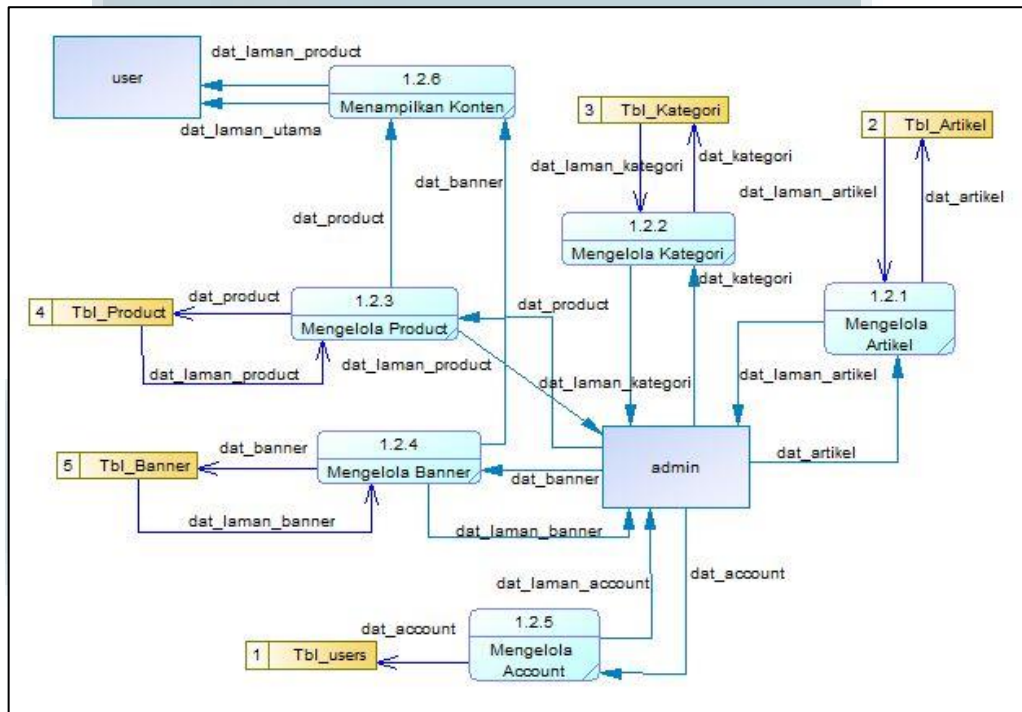


Gambar 3.2 Diagram Konteks GM Web Application

Gambar 3.2 merupakan diagram konteks yang menggambarkan . proses yang terdapat dalam GM Web Application, terdiri dari 2 entitas, yaitu *admin* dan *user*, serta memiliki sebuah proses yang menggambarkan keseluruhan sistem ini. Kemudian pada gambar 3.3 selanjutnya menggambarkan diagram level 0, menjelaskan tentang subproses dari proses diagram konteks sebelumnya, terdiri dari 2 buah proses yaitu, *login* sistem, dan mengelola laman *web*.



Gambar 3.3 Diagram *Level 0 GM Web Application*



Gambar 3.4 *Data Flow Diagram Level 1 Proses Mengelola Laman Web*

Gambar 3.4 terdapat 6 buah subproses dari proses mengelola laman *web*, yaitu mengelola artikel yang berfungsi untuk menambahkan, merubah, dan menghapus artikel, mengelola kategori, yang berfungsi untuk menambahkan, dan merubah kategori. Kemudian proses mengelola *product*, berfungsi untuk menambahkan, merubah, dan menghapus *product*. Selanjutnya proses mengelola *banner*, mengelola *account*, yaitu untuk menambahkan *account* baru, serta yang terakhir adalah menampilkan konten kepada *user*.

A.3 Flowchart

Flowchart merupakan alur logika yang diterapkan pada sebuah sistem, yang tujuan utamanya adalah untuk menjelaskan cara kerja dari suatu program yang akan dibuat agar lebih mudah dipahami. Oleh sebab itu dalam tahap pembangunan *web application* terlebih dahulu digambarkan dengan *flowchart* agar dapat mudah dipahami sehingga mempermudah proses pengembangan sistem, *flowchart* yang diciptakan untuk kebutuhan sistem ini diantaranya adalah:

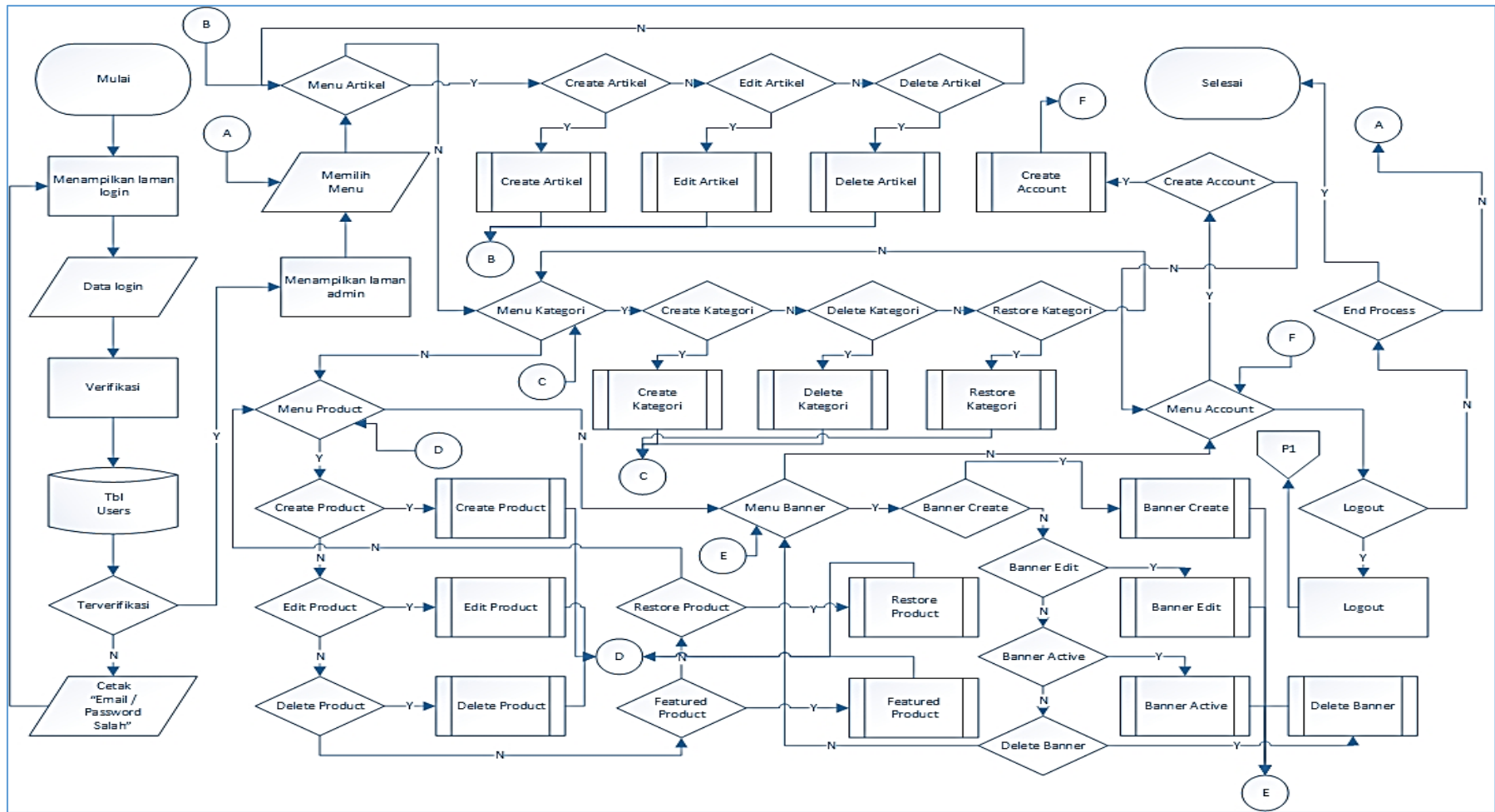
1) *Flowchart* CMS

Menjelaskan alur proses daripada CMS, dimulai dengan menampilkan laman *login*, admin perlu memasukkan data *login* ke dalam sistem, kemudian dilakukan proses verifikasi oleh sistem, apabila data *login* tersebut tidak terverifikasi maka sistem akan mengeluarkan *output* “Email/Password Salah”, sedangkan jika terverifikasi oleh sistem maka sistem akan menampilkan laman *admin*. Kemudian *admin* perlu melakukan *input* untuk memilih menu, terdapat 6 pilihan menu yaitu, menu artikel, menu kategori, menu *product*, menu *banner*, menu *account*, dan *logout*. Pada setiap menu

terdapat subproses daripada masing-masing menu tersebut yang dijelaskan lebih rinci pada *flowchart* setelah ini. Apabila *admin* tidak memilih submenu, *admin* akan dikembalikan pada tampilan untuk memilih menu tersebut. Kemudian pada menu *logout admin* akan dikembalikan ke laman utama *user* (hal ini digambarkan dengan konektor P1 yang akan tersambung ke dalam *flowchart web application*), tetapi apabila *admin* tidak melakukan pilihan menu dan tidak memilih untuk mengakhiri proses maka alur proses akan kembali lagi ke *input* untuk memilih menu. Hal ini dijelaskan pada gambar 3.5.



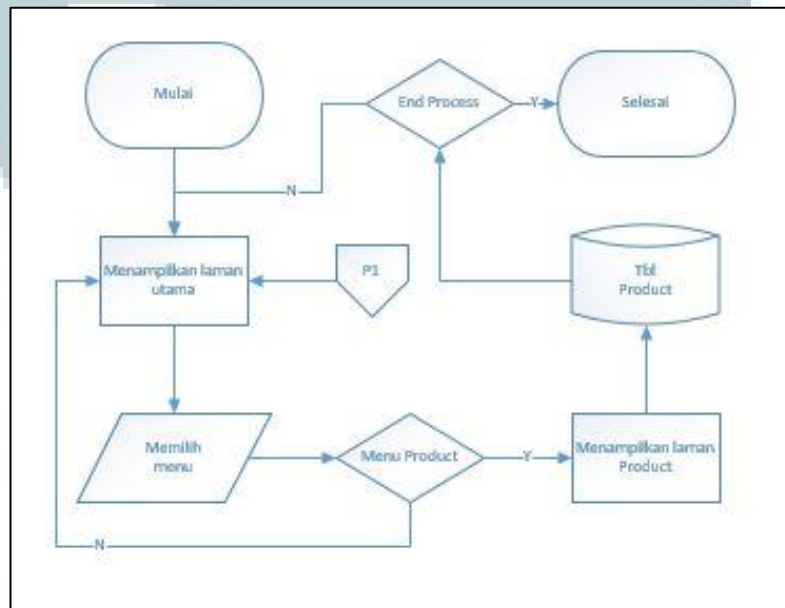
UMN



Gambar 3.5 Flowchart CMS

2) *Flowchart web application*

Flowchart ini menjelaskan alur proses daripada tampilan *web application* yang diakses oleh *user*. Alur proses dimulai dengan menampilkan laman utama, yang juga merupakan proses yang ditampilkan setelah *admin* melakukan *logout* yang menjadi konektor P1 dari *flowchart* sebelumnya. Setelah itu *user* perlu melakukan *input* untuk memilih menu, dan baru terdapat 1 menu, yaitu menu *product*, jika menu *product* dipilih maka sistem akan menampilkan laman *product*, setelah itu apabila *user* memilih untuk menyelesaikan proses maka alur proses ini akan selesai, namun jika tidak maka akan kembali lagi ke menampilkan laman utama. Dijelaskan pada gambar 3.6.

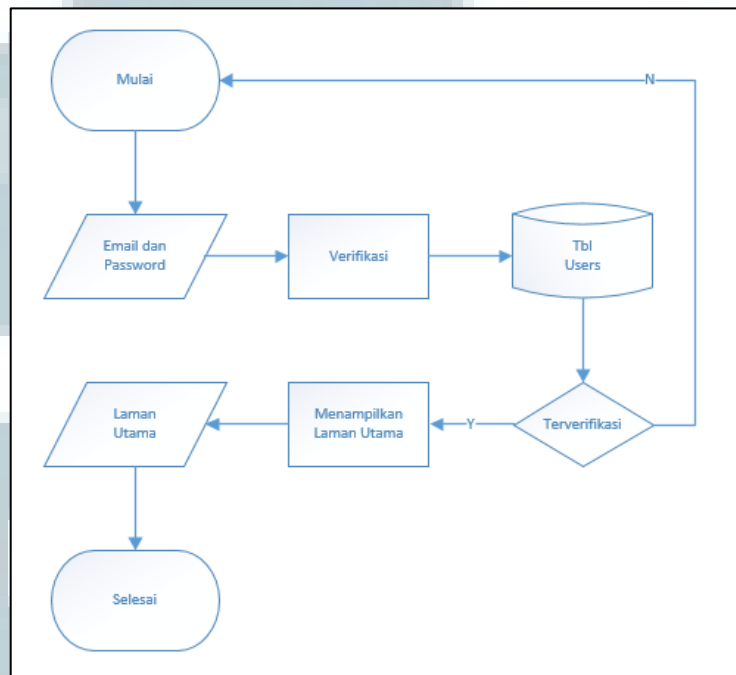


Gambar 3.6 *Flowchart web application*

3) *Flowchart login sistem*

Flowchart ini menjelaskan alur masuk ke sistem sebagai admin, pertama pengguna harus memasukkan *email* dan *password* ke dalam sistem, kemudian

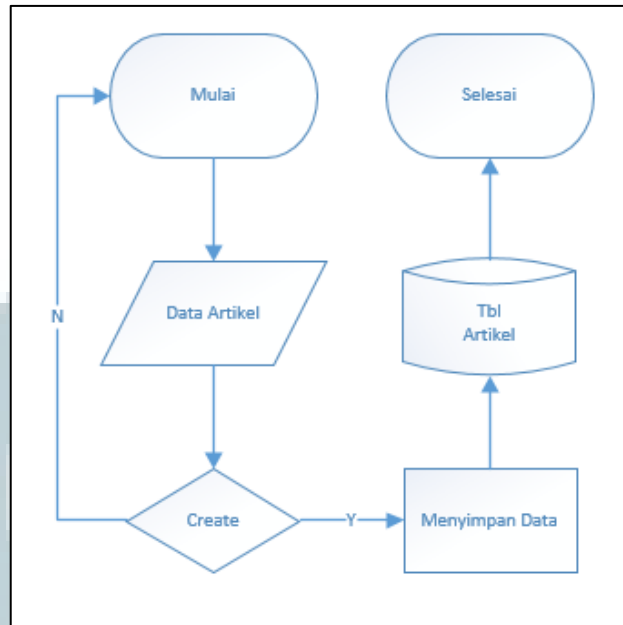
sistem melakukan verifikasi. Jika *email* dan *password* yang dimasukkan lolos verifikasi, maka akan ditampilkan laman utama daripada sistem *admin*, tetapi apabila salah, maka akan kembali ke menu *login* sistem, dapat dilihat pada gambar 3.7.



Gambar 3.7 *Flowchart login sistem*

4) *Flowchart create artikel*

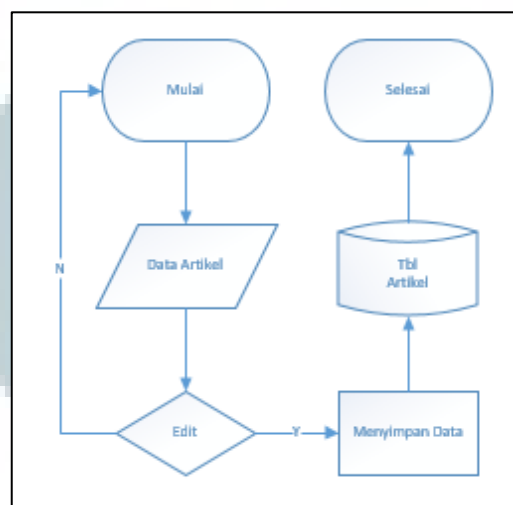
Fitur untuk melakukan *create* artikel baru, *admin* perlu untuk memasukkan data artikel. Kemudian apabila *button create* ditekan, maka data artikel baru tersebut akan disimpan ke dalam tabel artikel, dapat dilihat pada gambar 3.8.



Gambar 3.8 *Flowchart create artikel*

5) *Flowchart edit artikel*

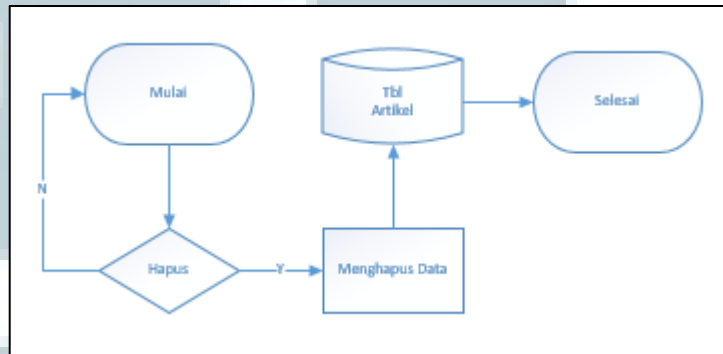
Fitur untuk melakukan *edit* terhadap data artikel yang telah dibuat sebelumnya, pertama *admin* perlu untuk memasukkan data artikel yang ingin dirubah. Kemudian apabila nilai *edit* di *trigger*, maka sistem akan menyimpan data ke dalam tabel artikel, dapat dilihat pada gambar 3.9.



Gambar 3.9 *Flowchart edit artikel*

6) *Flowchart delete* artikel

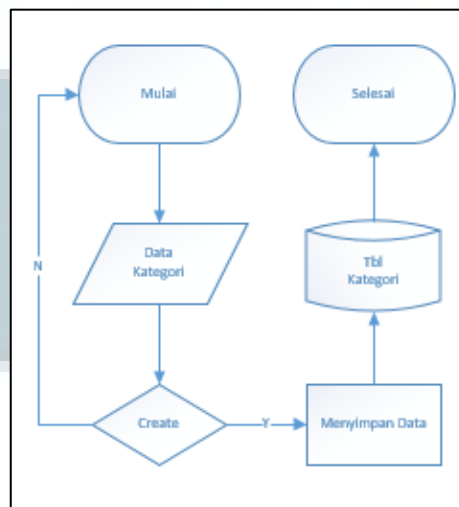
Fitur untuk melakukan *delete* terhadap data artikel yang telah dibuat, kemudian *admin* perlu melakukan *trigger* hapus untuk data artikel yang dipilih. Apabila nilai *trigger* diterima sistem, maka sistem akan menghapus data yang tersimpan di dalam tabel artikel, dapat dilihat pada gambar 3.10.



Gambar 3.10 *Flowchart delete* artikel

7) *Flowchart create* kategori

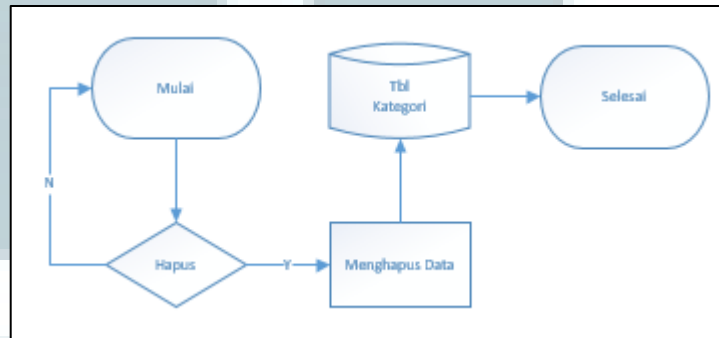
Fitur untuk membuat kategori baru, *admin* perlu memasukkan data kategori ke dalam sistem. Kemudian apabila nilai *create* di *trigger*, maka sistem akan menyimpan data baru tersebut, dapat dilihat pada gambar 3.11.



Gambar 3.11 *Flowchart create* kategori

8) *Flowchart delete kategori*

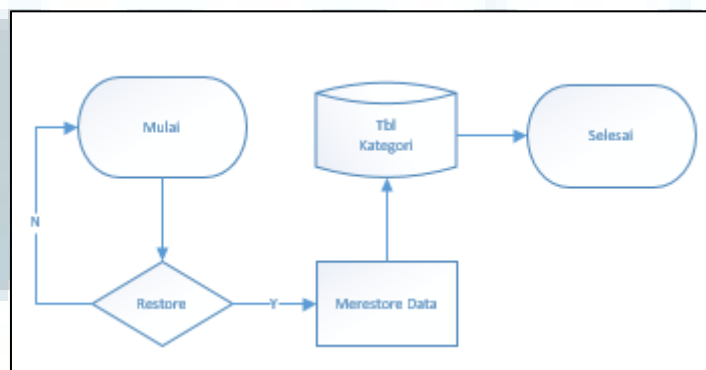
Fitur untuk menghapus kategori yang telah dibuat sebelumnya, *admin* perlu melakukan *trigger* hapus terhadap data kategori yang dipilih. Apabila *trigger* diterima sistem maka sistem akan menghapus data, dapat dilihat pada gambar 3.12.



Gambar 3.12 *Flowchart delete kategori*

9) *Flowchart restore kategori*

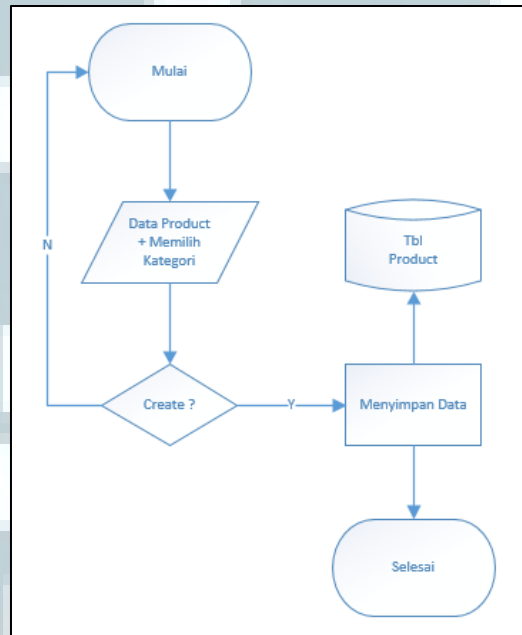
Fitur untuk melakukan *restore* daripada kategori yang telah dihapus sebelumnya. *Admin* perlu melakukan *trigger restore*, apabila nilai *trigger* diterima oleh sistem, maka sistem akan mengembalikan data kategori tersebut, dapat dilihat pada gambar 3.13.



Gambar 3.13 *Flowchart restore kategori*

10) *Flowchart create product*

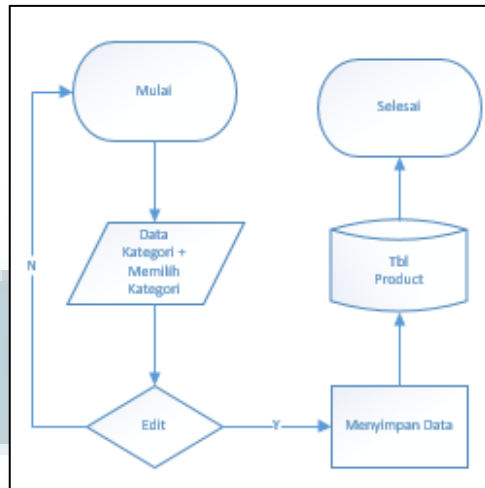
Fitur untuk membuat *product* yang baru. *Admin* perlu memasukkan data *product*, kemudian memilih kategori *product*. Kemudian apabila *trigger create* diterima sistem, maka sistem akan menyimpan data tersebut, dapat dilihat dalam gambar 3.14.



Gambar 3.14 *Flowchart create product*

11) *Flowchart edit product*

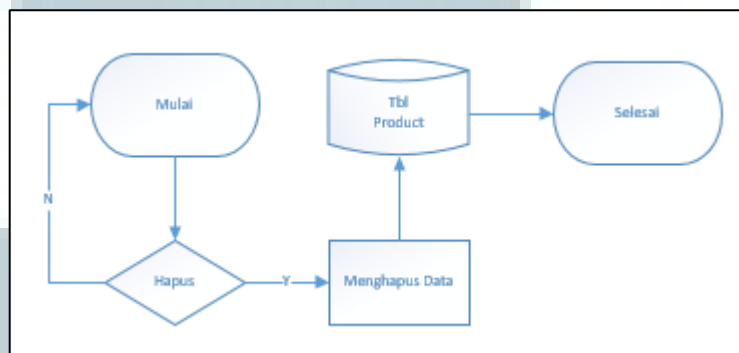
Fitur untuk melakukan perubahan pada data *product* yang telah dibuat oleh *admin*. *Admin* perlu kembali memilih kategori, dan jika sistem telah menerima *trigger edit*, maka sistem akan menyimpan data, dapat dilihat pada gambar 3.15.



Gambar 3.15 *Flowchart edit product*

12) *Flowchart delete product*

Fitur agar *admin* dapat menghapus *product* yang telah dibuat. Sistem harus menerima *trigger* hapus daripada data *product* yang telah dipilih. Jika sistem telah menerima maka data *product* tersebut akan dihapus, dapat dilihat pada gambar 3.16.

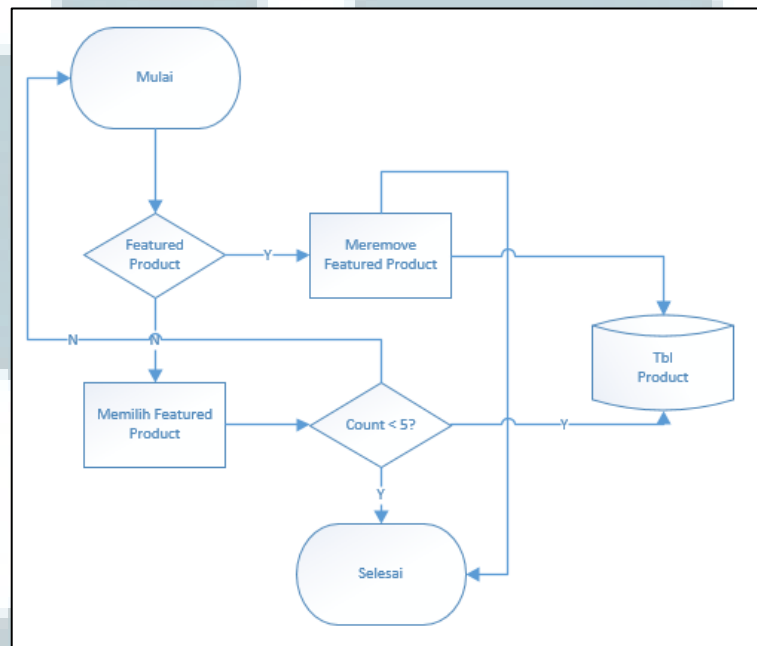


Gambar 3.16 *Flowchart delete product*

13) *Flowchart featured product*

Fitur untuk memilih *featured product*, *admin* pertama harus memilih data *product* mana yang akan dijadikan *featured product*. Kemudian apabila

featured product di *trigger* maka sistem akan melakukan pengecekan apakah data *product* tersebut sudah *featured* atau belum, jika data *product* sudah *featured* maka sistem akan menghapus *featured product* dari data *product* tersebut, dan berlaku sebaliknya, jika suatu data *product* bukan merupakan *featured product* maka sistem akan merubahnya menjadi *featured product*, apabila jumlah *featured product* yang ada dalam *database* sistem lebih kecil dari 5. Jika lebih besar dari 5 maka *admin* perlu menghapus terlebih dahulu *featured product* yang sudah ada, dapat dilihat pada gambar 3.17.

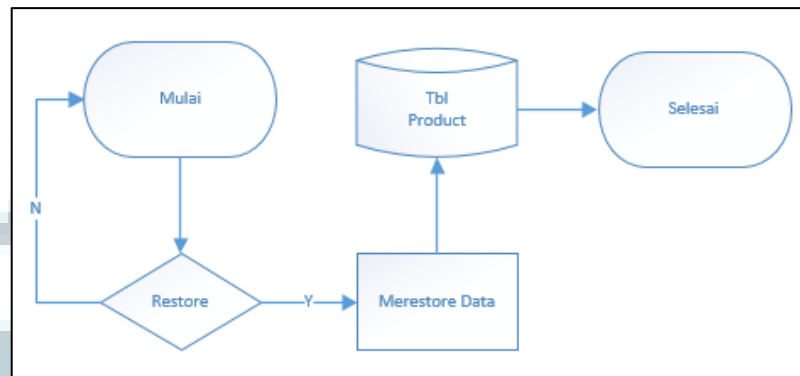


Gambar 3.17 *Flowchart featured product*

14) *Flowchart restore product*

Fitur untuk mengembalikan data *product* yang telah dihapus dan masih tersimpan dalam *trash*. Sistem perlu menerima *trigger restore* daripada

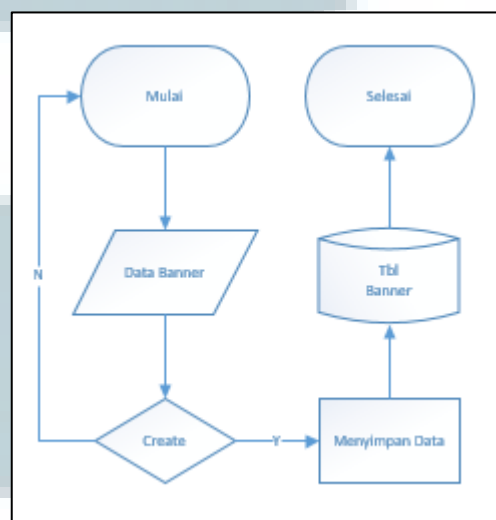
admin, jika sudah diterima maka data *product* yang bersangkutan akan di *restore* oleh sistem, dapat dilihat pada gambar 3.18.



Gambar 3.18 *Flowchart restore product*

15) *Flowchart banner create*

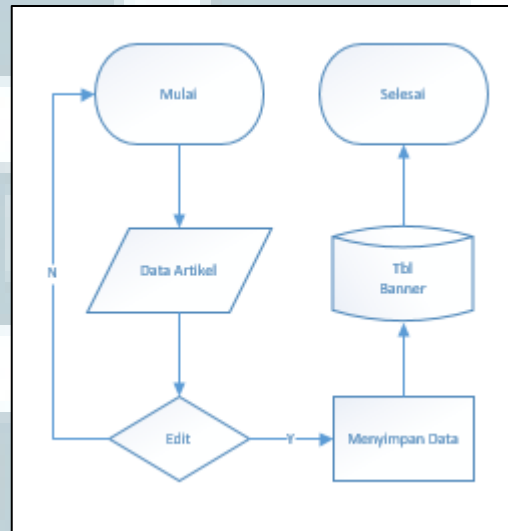
Fitur untuk membuat suatu *banner* baru. *Admin* perlu melakukan *input* data *banner*, dan melakukan *trigger create*. Apabila *trigger create* telah diterima oleh sistem maka sistem akan menyimpan data baru tersebut, dapat dilihat pada gambar 3.19.



Gambar 3.19 *Flowchart create banner*

16) *Flowchart banner edit*

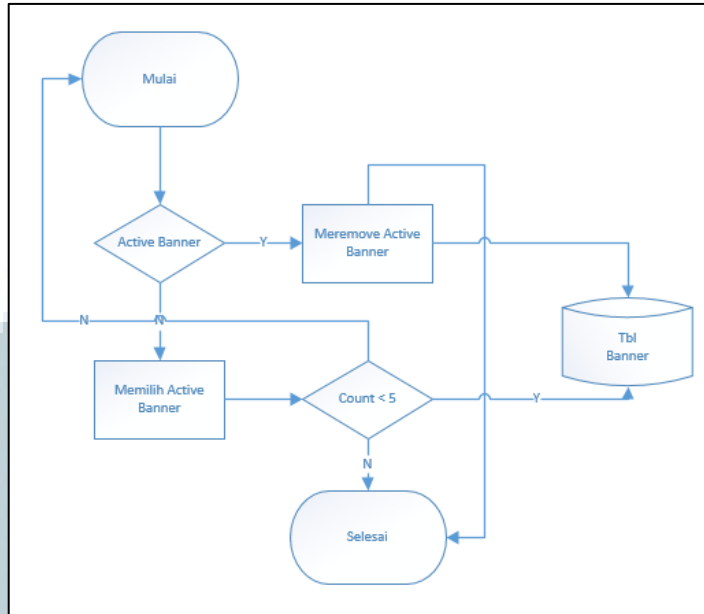
Fitur untuk melakukan *edit* daripada *banner* yang telah dibuat. Sistem perlu menerima *trigger edit*, dan masukkan data *banner* yang baru. Apabila sistem telah menerima, maka perubahan akan disimpan, dapat dilihat pada gambar 3.20.



Gambar 3.20 *Flowchart edit banner*

17) *Flowchart banner active*

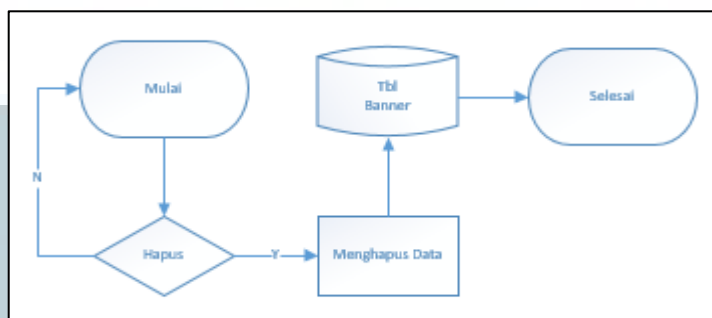
Fitur untuk memilih *banner* yang *aktif*. Alur logika daripada fitur ini mirip dengan fitur *featured product*, dapat dilihat pada gambar 3.21.



Gambar 3.21 *Flowchart active banner*

18) *Flowchart delete banner*

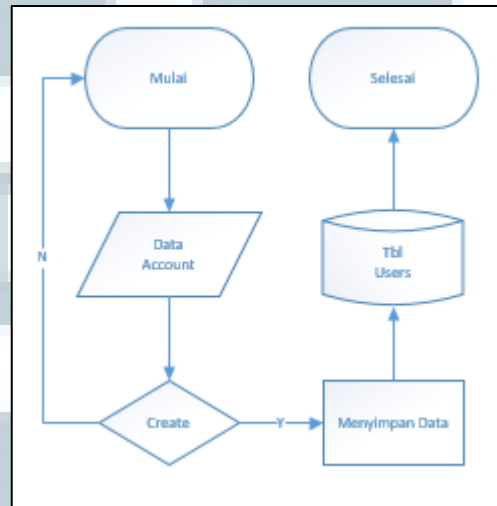
Fitur untuk melakukan proses penghapusan daripada *banner* yang telah dibuat. Sistem perlu menerima *trigger* hapus daripada data *banner* yang bersangkutan. Apabila sistem telah menerima *trigger* maka data *banner* tersebut akan dihapus dari sistem, dapat dilihat pada gambar 3.22.



Gambar 3.22 *Flowchart delete banner*

19) Flowchart create account

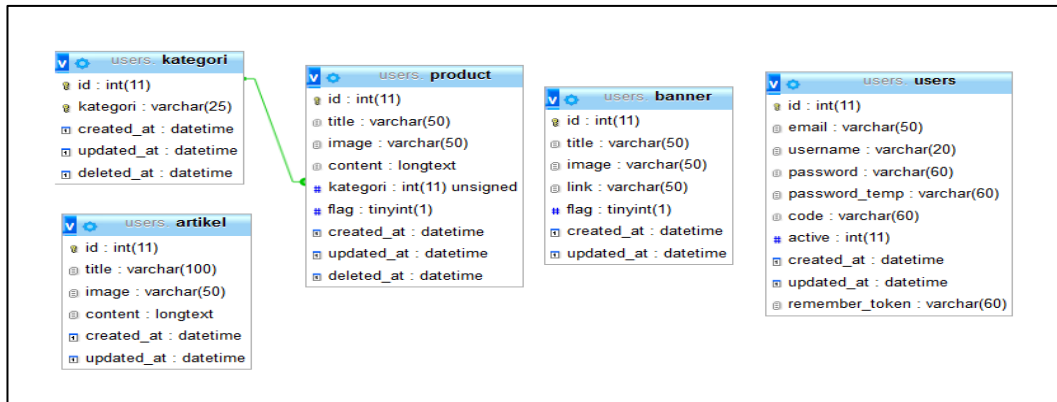
Fitur untuk melakukan pembuatan akun *admin* baru. *Admin* perlu memasukkan data *account*. Kemudian jika sistem menerima *trigger create*, maka data baru tersebut akan masuk ke dalam sistem, dan akun baru akan dibuat, dapat dilihat pada gambar 3.23.



Gambar 3.23 Flowchart create account

A.4 Entity Relationship Diagram

Dalam pembuatan sistem terdapat 5 buah tabel. Diantara tabel-tabel tersebut terdapat tabel yang memiliki relasi dengan tabel lainnya, seperti kategori dan *product*. Sedangkan sisanya seperti *users*, kategori, dan artikel tidak memiliki relasi dengan yang lainnya, dapat dilihat pada gambar 3.24.



Gambar 3.24 Entity Relationship Diagram GM Web Application

A.5 Struktur Tabel

1) Tabel Users

Tabel *users* digunakan untuk menyimpan data *account* daripada *admin*, tabel *users* sendiri memiliki 10 kolom pada tabel yaitu *users*, yaitu *id*, *email*, *username*, *password*, *password_temp*, *code*, *active*, *created_at*, *updated_at*, *remember_token*, seperti pada tabel 3.2.

Tabel 3.2 Struktur Tabel *Users*

No	Nama Kolom	Tipe Data	Keterangan
1	id	Int(11)	Primary key
2	email	Varchar(50)	
3	username	Varchar(20)	
4	password	Varchar(60)	
5	password_temp	Varchar(60)	
6	code	Varchar(60)	
7	active	Int(11)	
8	created_at	Datetime	
9	updated_at	Datetime	
10	remember_token	Varchar(60)	

2) Tabel Artikel

Tabel artikel digunakan untuk menyimpan data artikel, tabel artikel sendiri memiliki 6 kolom pada tabel, yaitu *id*, *title*, *image*, *content*, *created_at*, *updated_at*, seperti pada tabel 3.3.

Tabel 3.3 Struktur Tabel Artikel

No	Nama Kolom	Type Data	Keterangan
1	id	Int(11)	Primary key
2	title	Varchar(100)	
3	image	Varchar(50)	
4	content	Longtext	
5	created_at	Datetime	
6	updated_at	Datetime	

3) Tabel kategori

Tabel kategori digunakan untuk menyimpan data kategori yang merupakan kategori daripada *product*, sehingga tabel ini memiliki relasi dengan tabel *product*, tabel kategori memiliki 5 kolom yaitu, *id*, kategori, *created_at*, *updated_at*, *deleted_at*, seperti pada tabel 3.4.

Tabel 3.4 Struktur Tabel Kategori

No	Nama Kolom	Type Data	Keterangan
1	id	Int(11)	Primary key
2	kategori	Varchar(25)	
3	created_at	Datetime	
4	updated_at	Datetime	
5	deleted_at	Datetime	

4) Tabel *product*

Tabel *product* digunakan untuk menyimpan data *product*, tabel ini memiliki relasi dengan tabel kategori. Tabel *product* memiliki 9 kolom, yaitu *id*, *title*, *image*, *content*, kategori, *flag*, *created_at*, *updated_at*, *deleted_at*, seperti dalam tabel 3.5.

Tabel 3.5 Struktur Tabel *Product*

No	Nama Kolom	Tipe Data	Keterangan
1	id	Int(11)	Primary key
2	title	Varchar(50)	
3	image	Varchar(50)	
4	content	longtext	
5	kategori	Int(11)	Foreign key
6	flag	Tinyint(1)	
7	created_at	Datetime	
8	updated_at	Datetime	
9	deleted_at	Datetime	

5) Tabel *banner*

Tabel *banner* digunakan untuk menyimpan data *banner*, tabel ini memiliki 7 kolom, yaitu *id*, *title*, *image*, *link*, *flag*, *created_at*, *updated_at*, seperti dalam tabel 3.6.

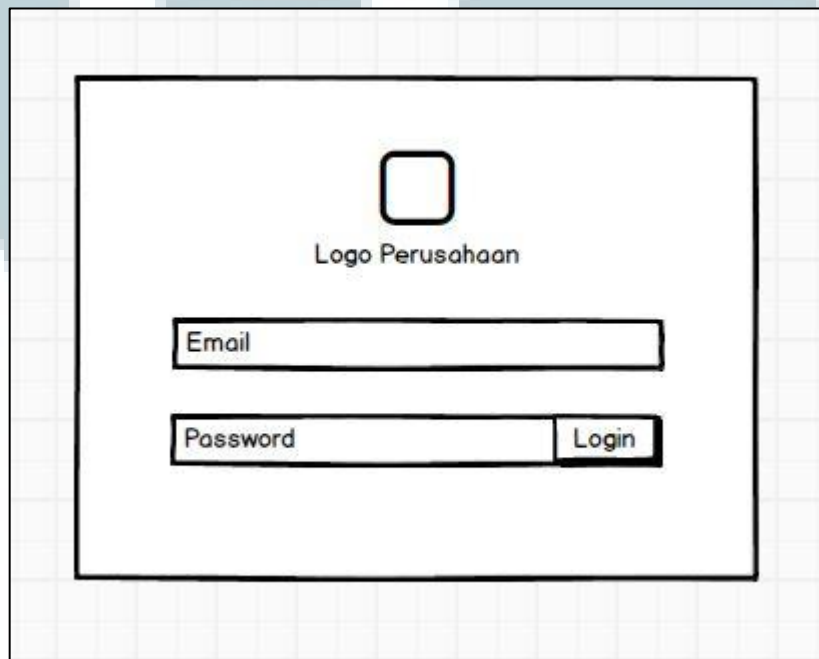
Tabel 3.6 Struktur Tabel *Banner*

No	Nama Kolom	Tipe Data	Keterangan
1	id	Int(11)	Primary key
2	title	Varchar(50)	
3	image	Varchar(50)	
4	link	Varchar(50)	
5	flag	Tinyint(1)	
6	created_at	Datetime	
7	updated_at	Datetime	

A.6 Rancangan Antarmuka

1) Laman Login

Laman *login* merupakan halaman yang digunakan oleh *admin* untuk masuk ke dalam sistem untuk mengelola laman *website*, dalam hal ini *admin* menggunakan CMS (*Content Management System*). Rancangan *interface* dibuat sesederhana mungkin agar mudah digunakan. Halaman *login* sendiri terdiri dari 2 buah teks boks yang digunakan untuk mengisi *email* dan *password*, dan terdapat logo daripada perusahaan dibagian atas serta ada 1 *button* untuk *login* ke dalam sistem, seperti pada gambar 3.25.

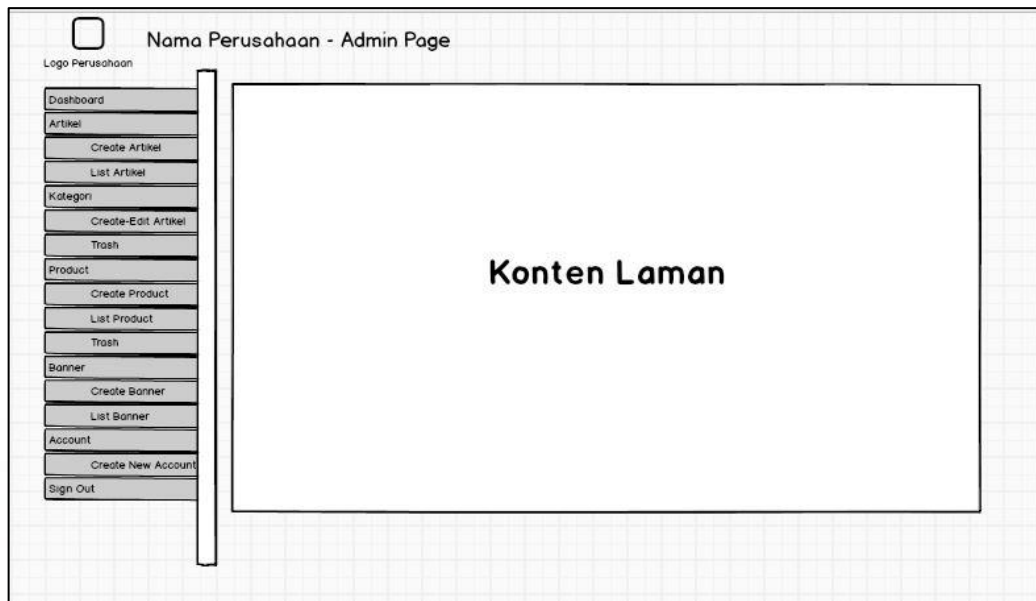


Gambar 3.25 Tampilan *Mock-up login* sistem

2) Laman Admin

Laman *admin* merupakan laman utama daripada antar muka *admin* yang pada sisi kiri terdapat *navigation bar*, pada sisi kanan terdapat isi konten daripada

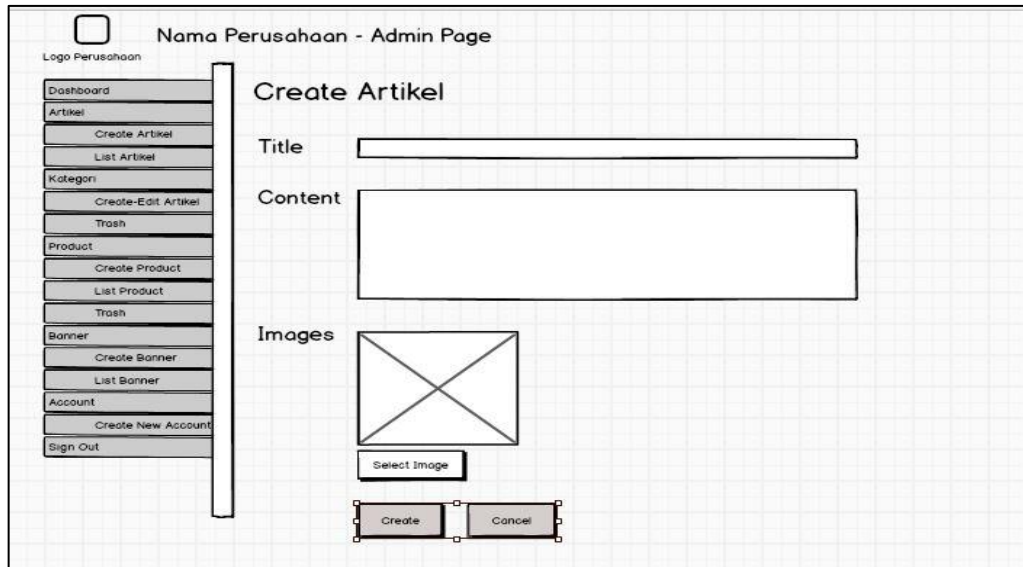
laman *admin*, serta dibagian atas terdapat logo perusahaan, serta terdapat nama perusahaan, seperti pada gambar 3.26.



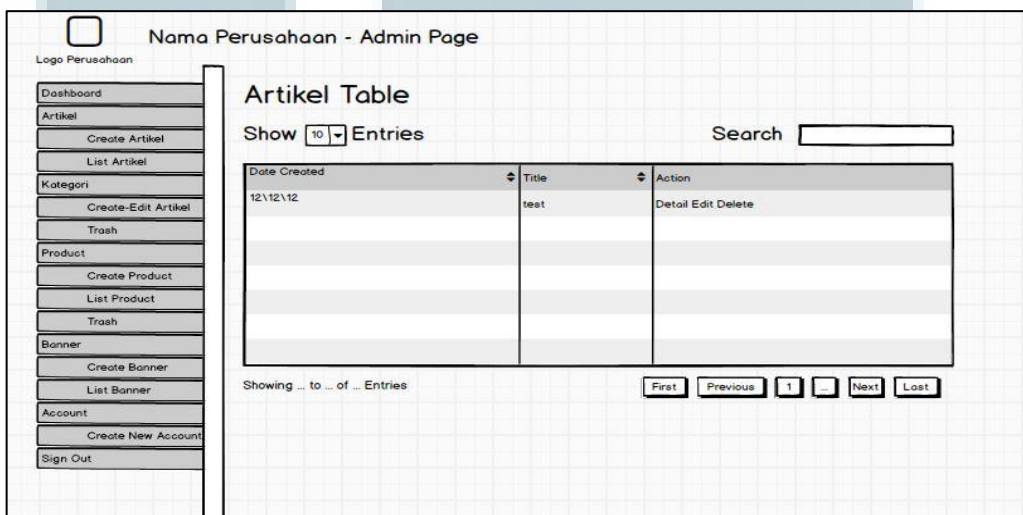
Gambar 3.26 Tampilan *Mock-up* laman utama *admin*

3) Laman Artikel

Dalam laman artikel, *admin* dapat melakukan penambahan konten artikel baru, melihat daftar artikel yang sudah ada, melakukan perubahan terhadap artikel, dan menghapus artikel yang sudah dibuat. Tampilan daripada laman artikel sendiri, mengikuti dengan dasar utama tampilan laman *admin*, di sisi kiri terdapat *navigation bar*, pada *header* terdapat logo dan nama perusahaan, serta dibagian sisi kanan merupakan konten daripada laman artikel itu sendiri, rancangan antarmuka dapat dilihat lebih jelas dalam gambar di bawah 3.27 dan 3.28.



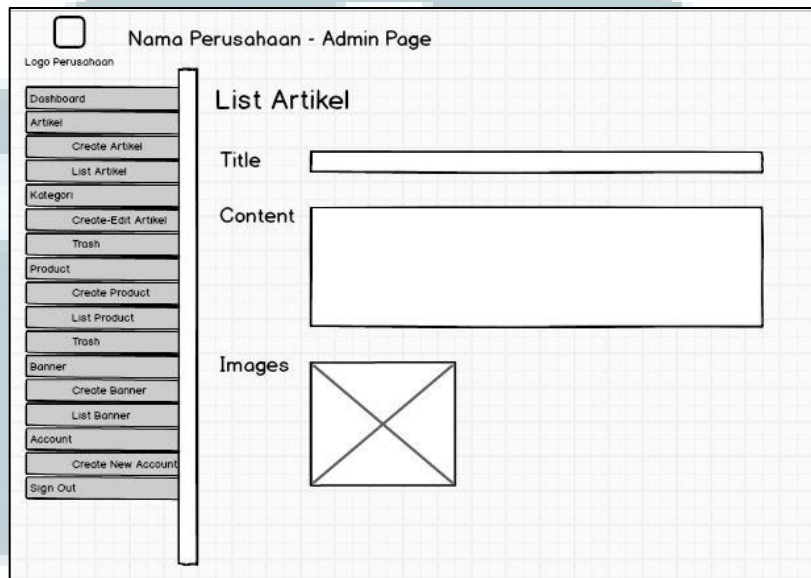
Gambar 3.27 Tampilan *Mock-up* laman *create* artikel



Gambar 3.28 Tampilan *Mock-up* laman *list* artikel

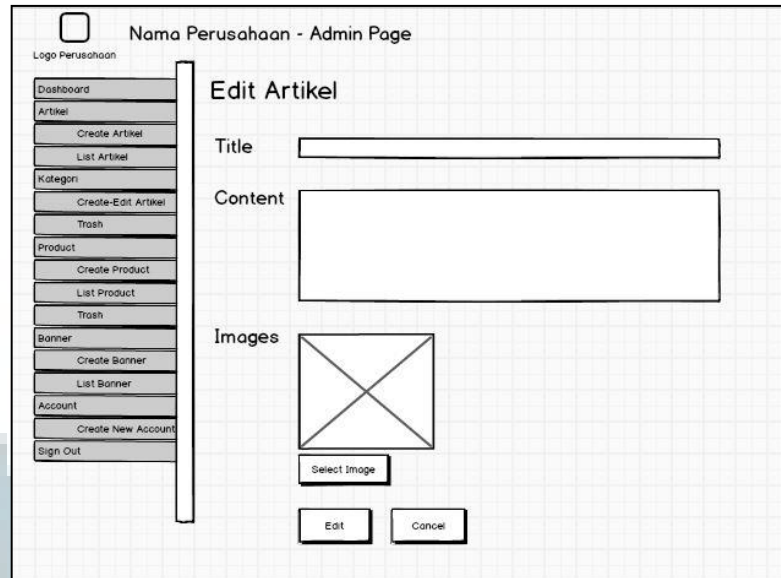
Pada tampilan *create* artikel terdapat *text box* untuk mengisi judul dari artikel, *text area* untuk mengisi konten daripada artikel, serta *thumbnail* daripada gambar yang akan dipilih untuk dimasukkan ke dalam artikel tersebut, serta terdapat 3 *button*, pertama untuk memilih gambar, lalu *button* untuk meng-*create* artikel, dan *button cancel*.

Pada tampilan *list* artikel akan menampilkan tabel artikel yang berisikan daftar daripada artikel yang terdapat beberapa *button* untuk memilih halaman, serta untuk menampilkan seberapa banyak data, dan terdapat *search box* untuk mencari data daripada artikel, serta pada bagian atas daripada tabel dapat di *click* untuk melakukan *sorting*. Dijelaskan pada gambar 3.29.



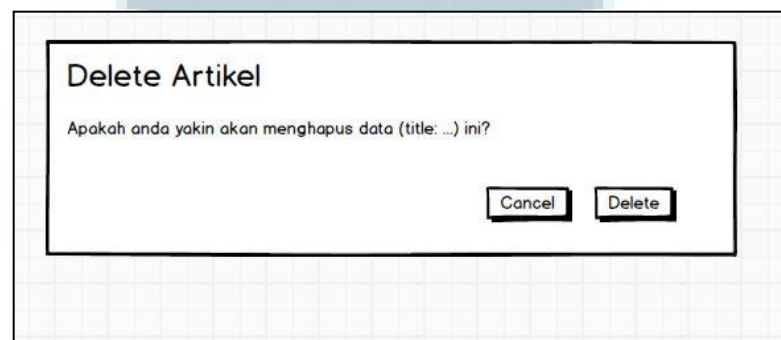
Gambar 3.29 Tampilan *Mock-up* laman utama *detail* artikel

Tampilan detail apabila suatu artikel dipilih untuk melihat detailnya dari laman *list* artikel, memiliki tampilan kurang lebih seperti laman *create* artikel, hanya saja tidak terdapat *button*.



Gambar 3.30 Tampilan *Mock-up* laman *edit* artikel

Laman *edit* artikel akan muncul apabila pengguna memilih untuk melakukan *edit* daripada suatu artikel, tampilan daripada laman ini sendiri sama persis seperti laman *create* artikel, hanya terdapat perbedaan di penamaan *button*. Dijelaskan pada gambar 3.30.

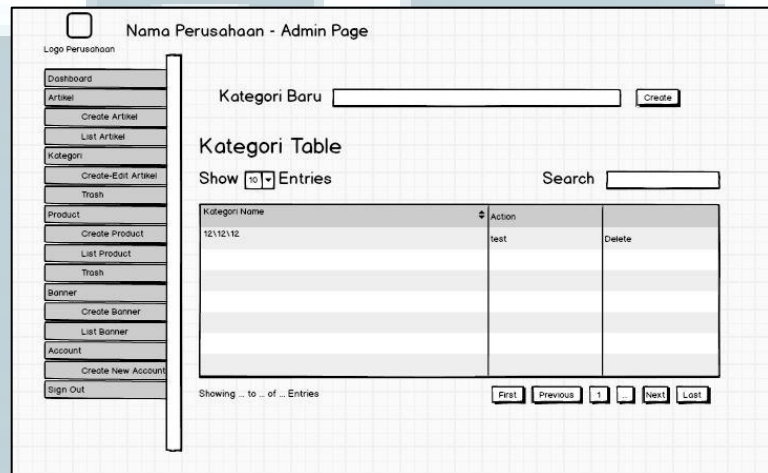


Gambar 3.31 Tampilan *Mock-up* laman *pop-up delete* artikel

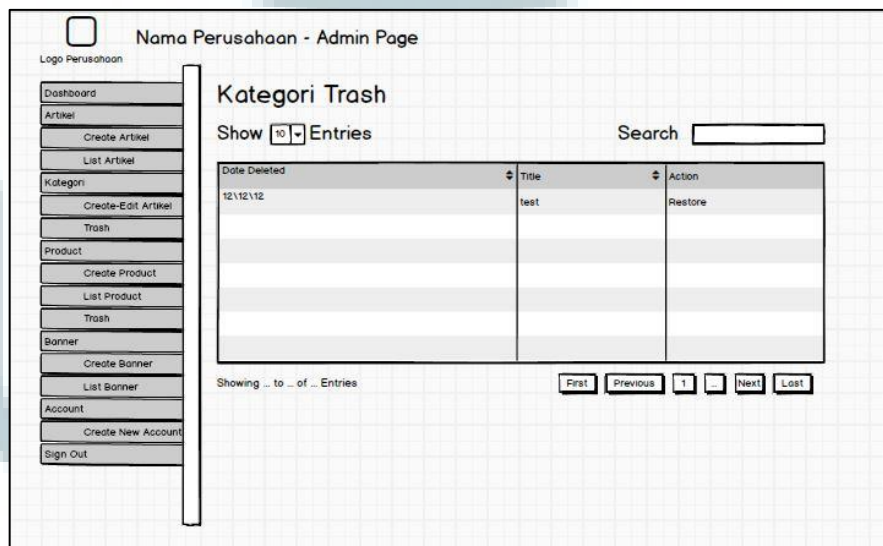
Laman *delete* artikel akan muncul apabila pengguna memilih untuk *delete* suatu artikel, yang akan menampilkan *pop-up* seperti gambar 3.31.

4) Laman Kategori

Laman *Create-Edit* kategori berfungsi untuk menambahkan kategori baru ke dalam sistem, yang dapat dimasukkan melalui *text box* yang terdapat di bagian atas daripada tampilan, sedangkan dibagian bawahnya memiliki tampilan tabel yang mirip dengan tampilan sebelumnya, dijelaskan dalam gambar 3.32 dan 3.33.



Gambar 3.32 Tampilan *Mock-up* laman *create-edit* kategori



Gambar 3.33 Tampilan *Mock-up* laman kategori *trash*

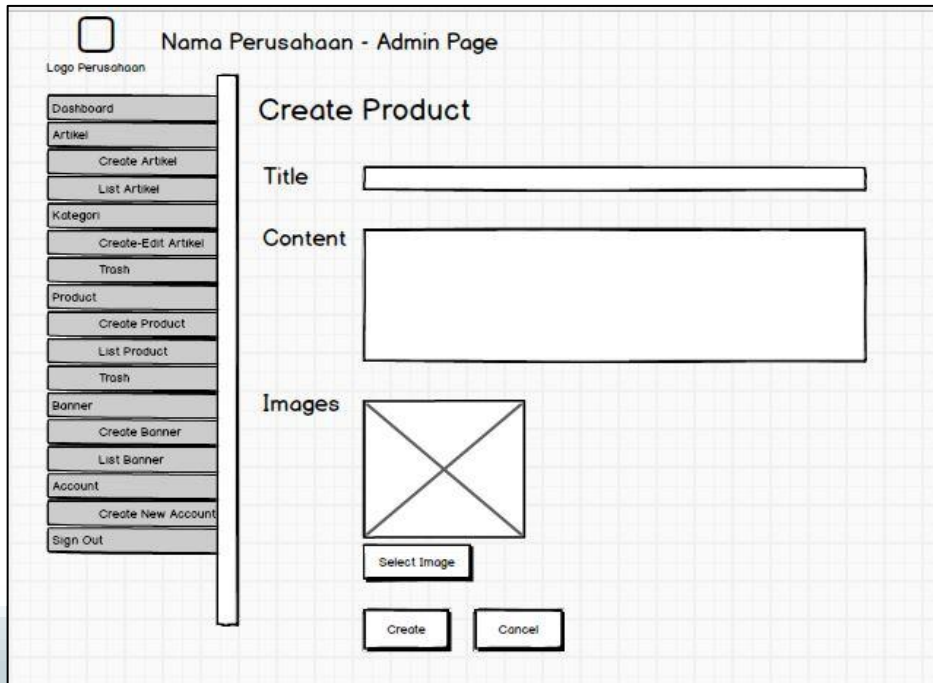
Laman kategori *trash* berfungsi untuk menampilkan tabel yang berisi kategori yang telah dihapus, terdapat opsi untuk melakukan *restore*, untuk mengembalikan kategori yang sudah dihapus kembali ke dalam sistem.

5) Laman *Product*

Laman *product* memiliki tampilan yang serupa dengan tampilan sebelumnya serta memiliki fungsi yang serupa pula dengan sebelumnya seperti melakukan *create product*, *list product*, *edit product*, *detail product*, *trash product*, dan *delete product*, serta dapat memilih *product* yang aktif digunakan untuk *featured product* untuk melihat tampilan *product* secara jelas dapat dilihat pada gambar 3.34, 3.35, 3.36, 3.37, 3.38, dan 3.39.

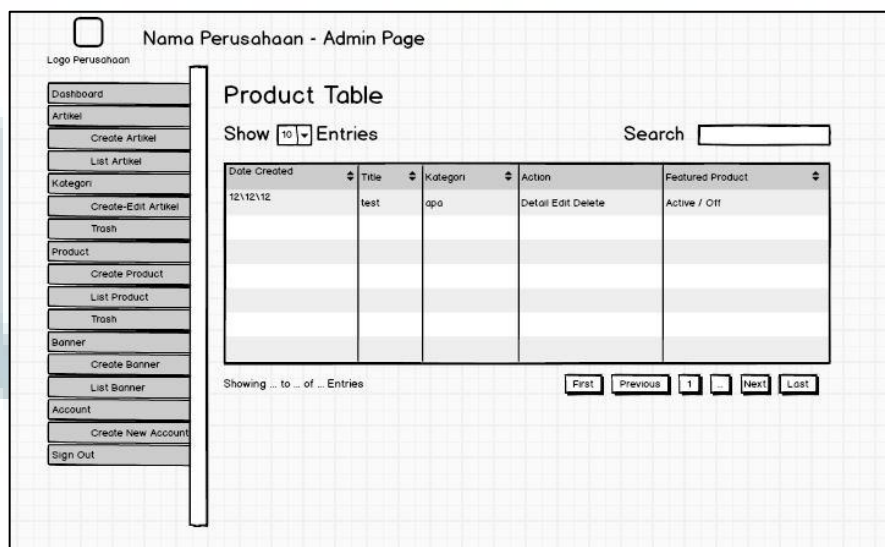
Pada gambar 3.34 terdapat 1 teks boks ditujukan untuk memasukkan judul daripada produk baru yang ingin diinput, kemudian terdapat teks *area* untuk memasukkan konten daripada produk tersebut berupa penjelasan dan sebagainya, dan terakhir terdapat 3 *button*, yang pertama *button select image* untuk memilih gambar yang ingin dimasukkan, lalu *button create* untuk memasukkan produk tersebut ke dalam sistem, dan *button cancel* untuk batal.

UMMN



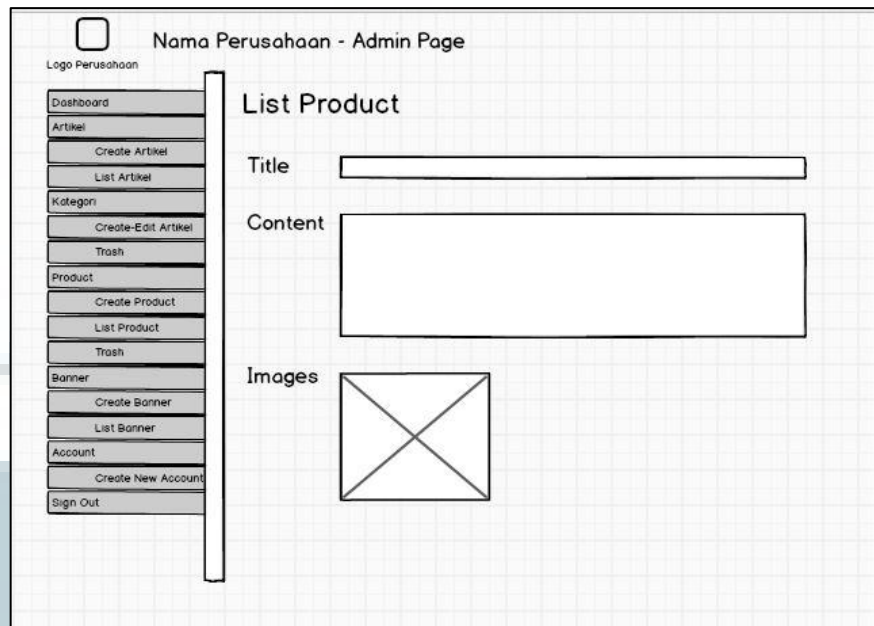
Gambar 3.34 Tampilan *Mock-up* laman *create product*

Pada gambar 3.35 menjelaskan fitur *list* produk yang menampilkan tabel yang berisikan data produk dari dalam *database*, serta terdapat opsi untuk melakukan *delete* dan *edit* produk tersebut, serta terdapat *button* untuk melakukan *toggle active* daripada fitur *featured* produk.



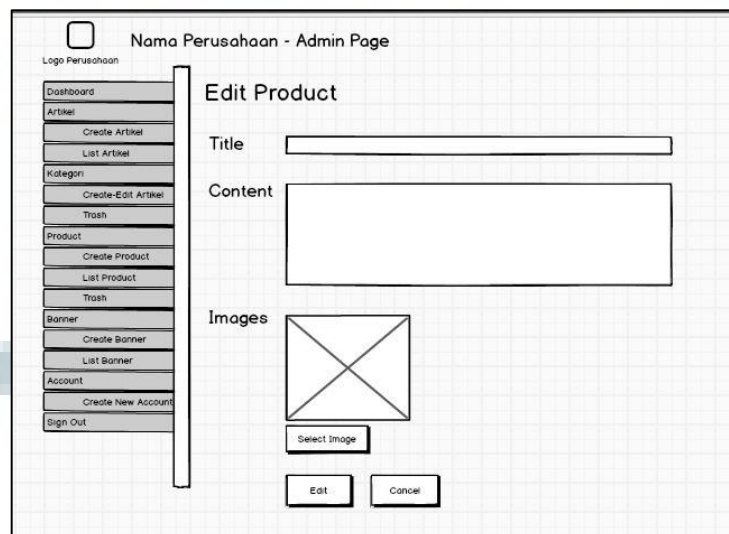
Gambar 3.35 Tampilan *Mock-up* laman *list product*

Kemudian pada gambar 3.36 dijelaskan fitur *detail* produk yang di dalamnya terdapat penjelasan tentang judul, konten, dan gambar tentang suatu produk.



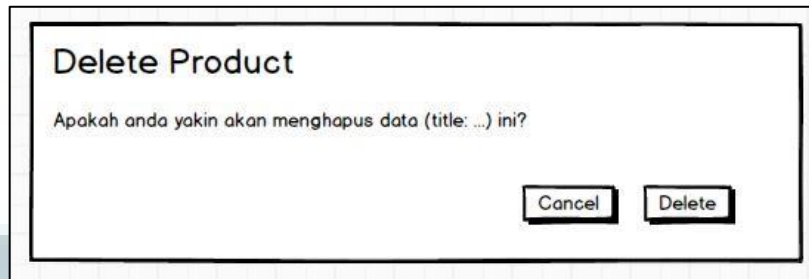
Gambar 3.36 Tampilan *Mock-up* laman *detail product*

Pada gambar 3.37 dijelaskan fitur *edit* produk yang di dalamnya terdapat teks boks untuk melakukan perubahan judul produk, perubahan isi konten, dan untuk merubah gambar dari suatu produk yang dipilih pada *list* produk.



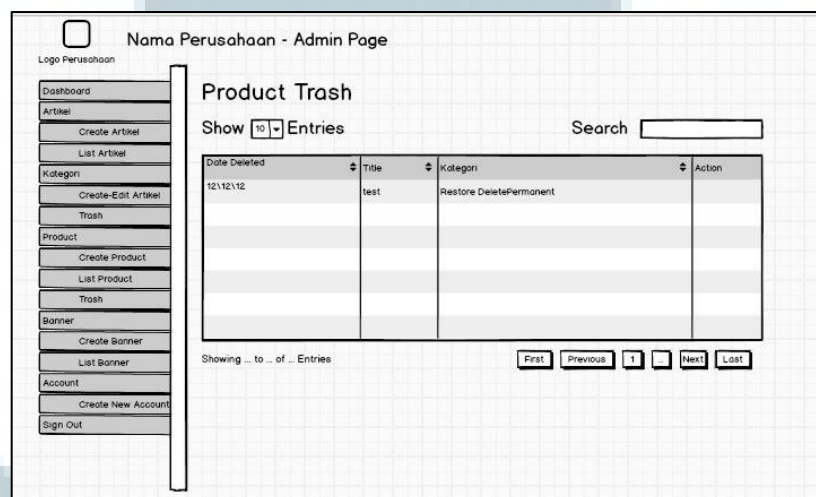
Gambar 3.37 Tampilan *Mock-up* laman *edit product*

Pada gambar 3.38 menjelaskan tentang *pop-up* apabila memilih untuk melakukan *delete product* dan terdapat menu *pop-up* konfirmasi untuk melakukan *delete*.



Gambar 3.38 Tampilan *Mock-up* laman *pop-up delete product*

Pada gambar 3.39 dijelaskan tentang tampilan daripada *product trash* yang menyimpan daftar produk-produk yang telah dihapus sebelumnya. Dalam fitur ini terdapat pilihan *action* untuk melakukan *restore* daripada produk yang telah dihapus tersebut, daftar produk ini sendiri ditampilkan dalam tabel seperti *list produk*.

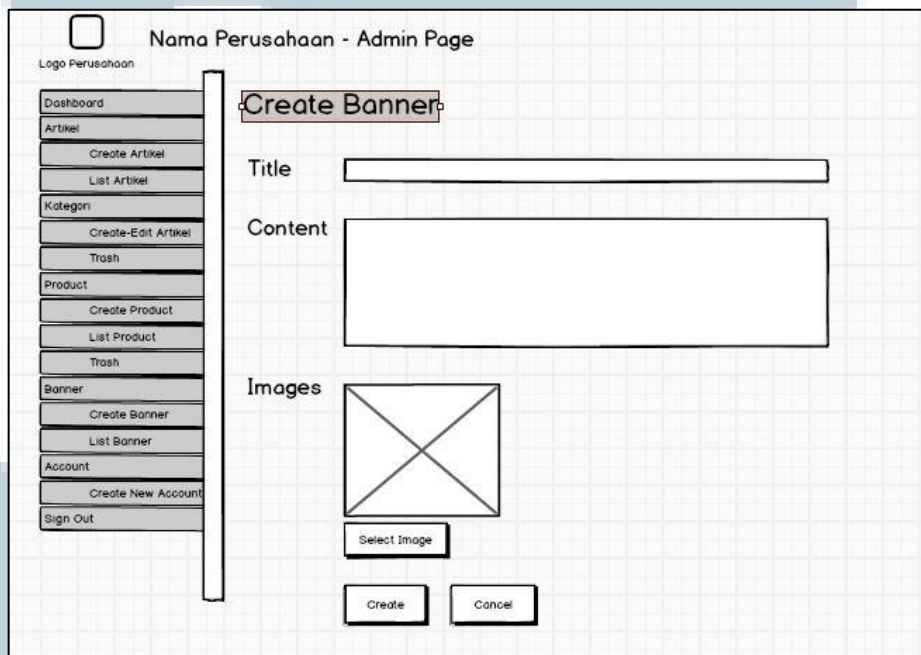


Gambar 3.39 Tampilan *Mock-up* laman *product trash*

6) Laman Banner

Tampilan daripada laman *banner* sendiri, serupa dengan tampilan-tampilan pada laman sebelumnya, terdapat laman *create*, *list*, *detail*, *edit*, dan *delete banner*, yang dapat dilihat pada gambar 3.40, 3.41, 3.42, 3.43, dan 3.44.

Pada gambar 3.40 dijelaskan tentang fitur untuk melakukan *create banner* yang didalamnya terdapat 1 teks boks untuk memasukkan judul daripada *banner*, kemudian terdapat 1 teks area untuk memasukkan konten daripada *banner* yaitu berupa link daripada gambar, kemudian pada bagian terakhir terdapat *button* untuk memasukkan *image* ke dalam *banner* yang akan dibuat tersebut.

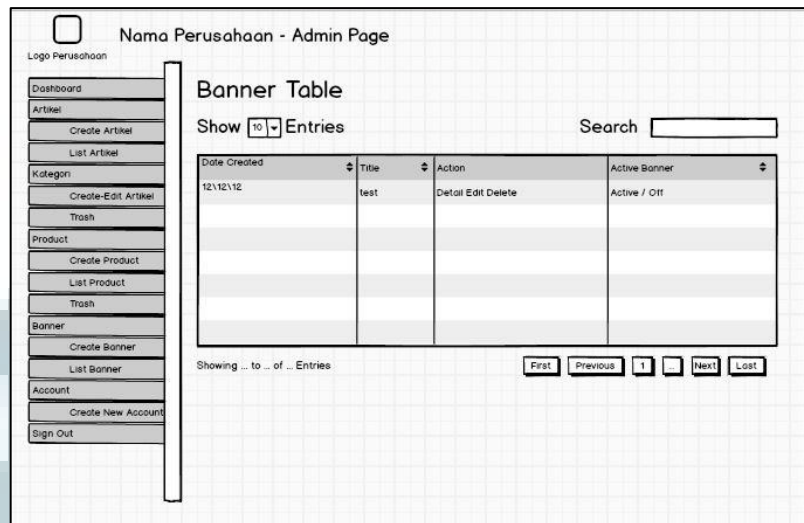


The image shows a web application interface for creating a banner. At the top, there is a header with a logo placeholder and the text 'Nama Perusahaan - Admin Page'. Below the header is a sidebar menu with the following items: Dashboard, Artikel (with sub-items 'Create Artikel' and 'List Artikel'), Kategori (with sub-items 'Create-Edit Artikel' and 'Trash'), Product (with sub-items 'Create Product', 'List Product', and 'Trash'), Banner (with sub-items 'Create Banner', 'List Banner'), Account (with sub-item 'Create New Account'), and Sign Out. The main content area is titled 'Create Banner' and contains three input fields: a text box for 'Title', a text area for 'Content', and an image selection area labeled 'Images'. The 'Images' area includes a square placeholder with an 'X' and a 'Select Image' button. At the bottom of the form are 'Create' and 'Cancel' buttons.

Gambar 3.40 Tampilan *Mock-up* laman *create banner*

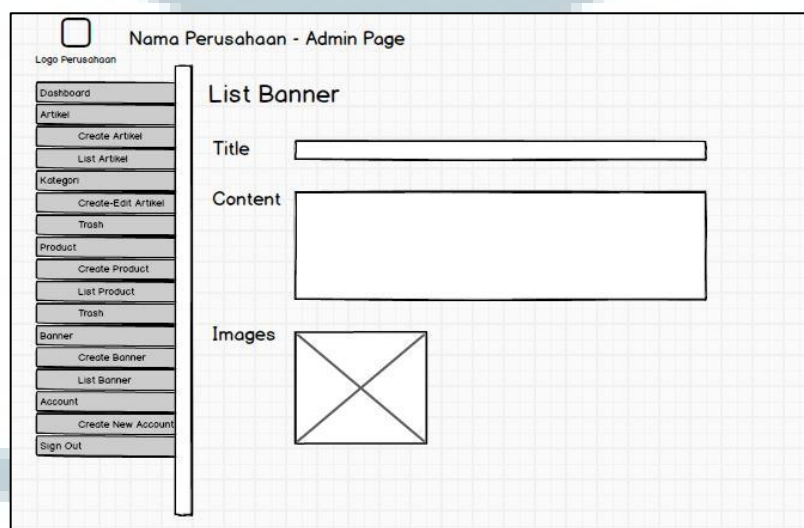
Pada gambar 3.41 merupakan tampilan tabel dalam menampilkan daftar dari *banner* yang telah dibuat dan disimpan dalam *database*. Di dalam tampilan

ini terdapat menu untuk melakukan *delete* dan *edit*, serta untuk mengaktifkan atau menonaktifkan *banner*.



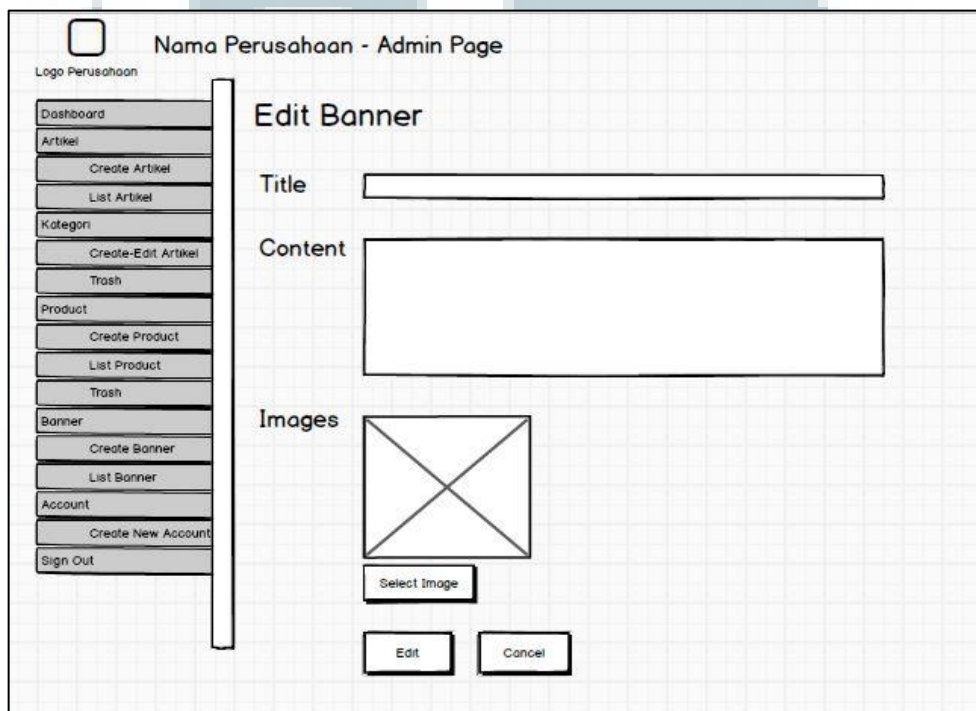
Gambar 3.41 Tampilan *Mock-up* laman *list banner*

Pada gambar 3.42 merupakan tampilan *detail banner* untuk melihat tampilan detail daripada *banner* yang telah dibuat. Di dalam tampilan ini terdapat judul, konten yang berisi *link*, serta gambar daripada *banner* tersebut.



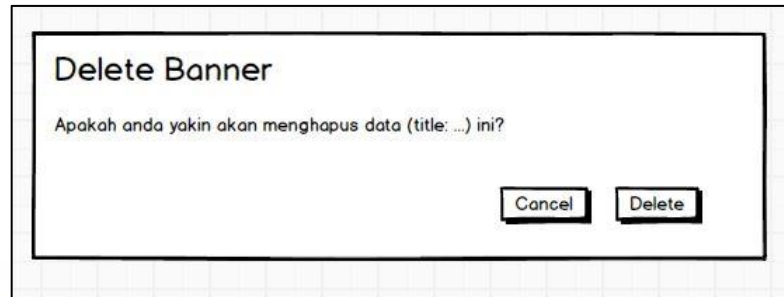
Gambar 3.42 Tampilan *Mock-up* laman *detail banner*

Pada gambar 3.43 digambarkan tampilan untuk melakukan *edit* daripada *banner* yang telah dibuat terdapat teks boks untuk melakukan perubahan terhadap *banner* yang telah dibuat, kemudian terdapat teks *area* untuk merubah isi konten daripada *banner* yang telah dibuat, serta dapat dilakukan *edit* daripada gambar pada *banner* tersebut untuk memilih gambar yang lainnya.



Gambar 3.43 Tampilan *Mock-up* laman *edit banner*

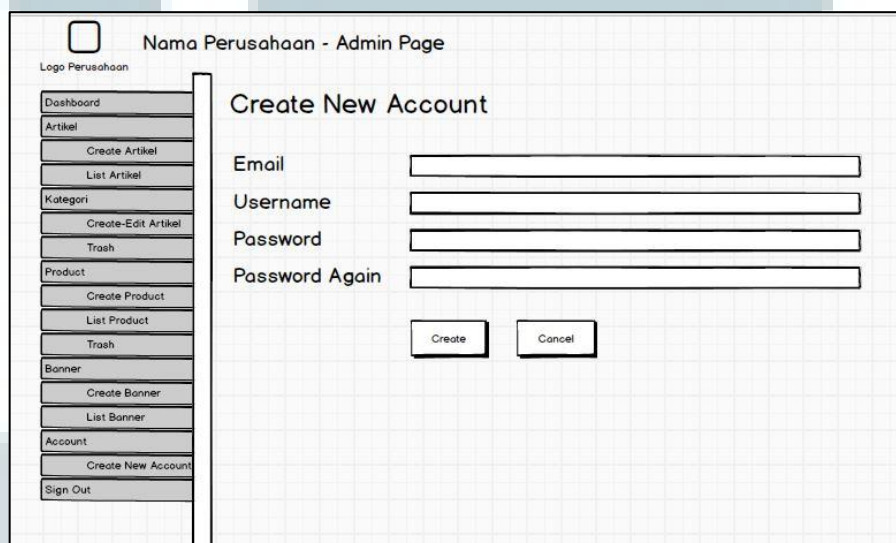
Sedangkan pada gambar 3.44 menampilkan *pop-up* untuk melakukan proses *delete banner*, dan ketika fitur untuk melakukan hapus daripada *banner* yang telah dibuat maka akan muncul dialog konfirmasi.



Gambar 3.44 Tampilan *Mock-up* laman *pop-up delete banner*

7) Laman Account

Laman *account* sendiri berfungsi untuk membuat *account admin* baru, yang memiliki 4 teks boks yang digunakan untuk meng-*input* data berupa *email*, *username*, *password*, dan pengulangan *password*, serta terdapat 2 *button*, yaitu *create*, dan *cancel*. Dijelaskan pada gambar 3.45.



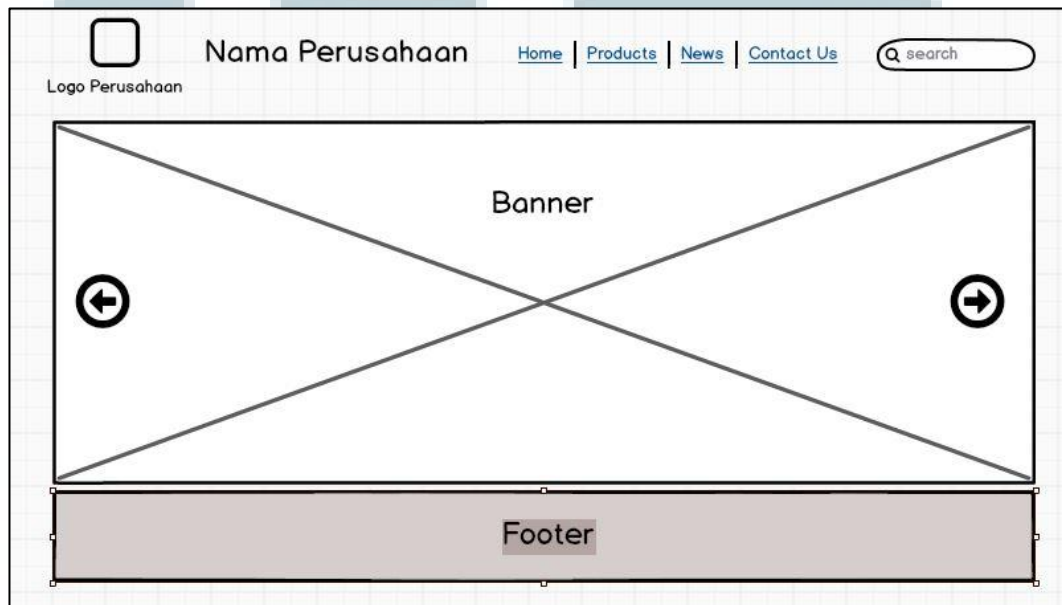
Gambar 3.45 Tampilan *Mock-up* laman *create new account*

8) Laman *Home* dan *Product* untuk User

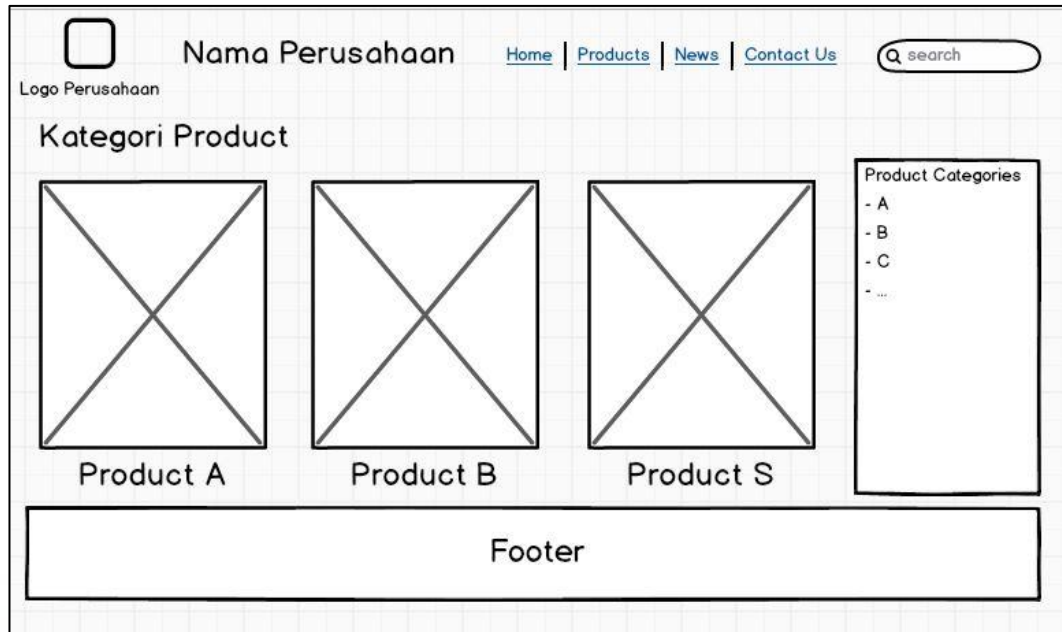
Laman *home* dan *product* yang merupakan tampilan utama daripada *website* yang ditujukan untuk *user* dari *web application* ini, untuk tampilan *home*

sendiri terdiri dari *header* yang berisikan logo perusahaan, nama perusahaan, serta *navigation bar*, dan adanya *search bar*, sedangkan untuk bagian konten berisikan *banner* yang merupakan *slideshow* gambar berukuran besar yang diperoleh dari *database banner*, lalu pada bagian bawah terdapat *footer*, digambarkan pada gambar 3.46.

Untuk *laman product* merupakan tampilan *product* bagi *user*. Secara garis besar tampilan seperti laman *home*, sedangkan untuk konten berisikan daftar daripada *product*, berupa gambar, dan nama *product*, dapat dilihat pada gambar 3.47.



Gambar 3.46 Tampilan *Mock-up* laman *home user*



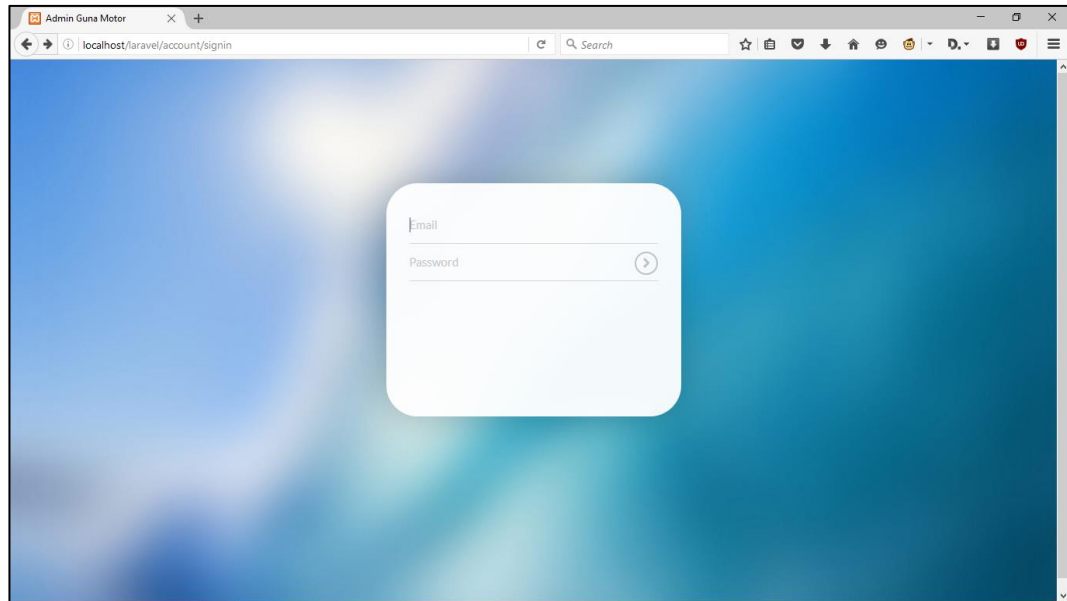
Gambar 3.47 Tampilan *Mock-up* laman *product user*

B. Implementasi Program

Implementasi program pada kerja magang dilakukan dengan menjalankan modul-modul yang ada pada sistem. Hal tersebut akan dijelaskan lebih lanjut pada poin-poin di bawah ini.

B.1 Modul Login

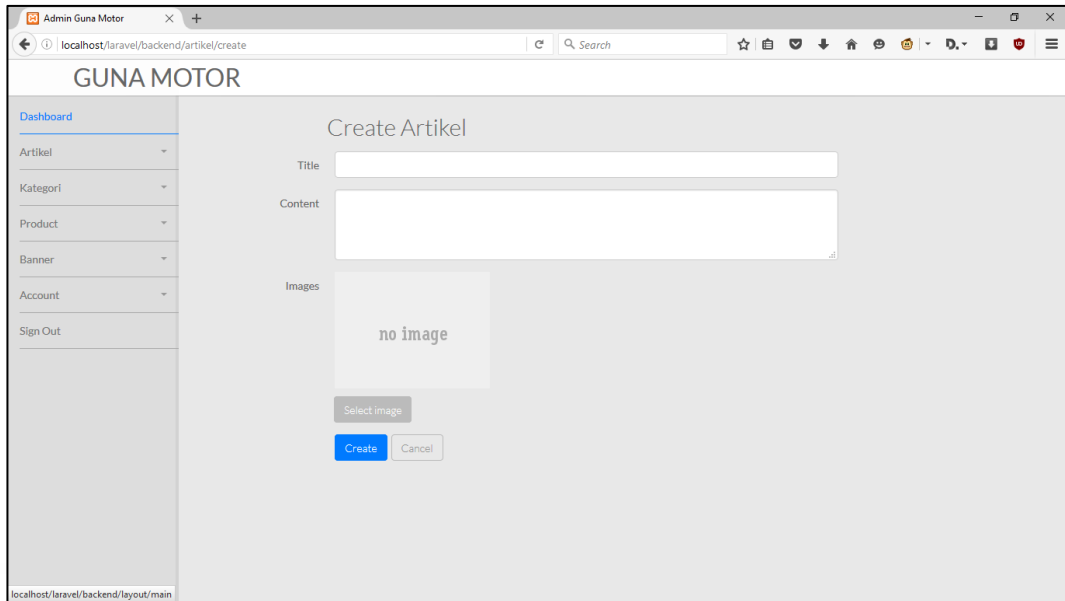
Modul *login* merupakan modul yang digunakan untuk masuk ke dalam sistem dengan memasukkan *email* dan *password* pada *text box*. Kemudian dengan menekan *enter* atau *button* yang tersedia. Jika data *login* yang dimasukkan benar maka akan masuk ke dalam tampilan *admin* untuk mengelola tampilan daripada laman *web*, tampilan seperti gambar 3.48.



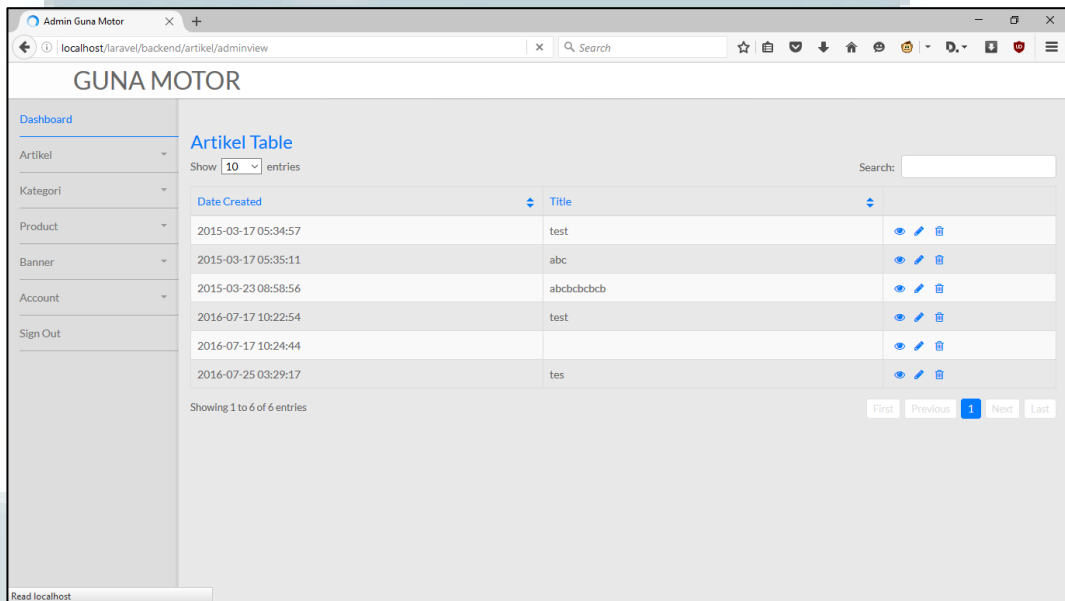
Gambar 3.48 Tampilan laman *login*

B.2 Modul Artikel

Modul artikel digunakan untuk membuat artikel baru, merubah artikel, dan menghapus artikel yang sudah ada, untuk mengakses ini dapat dengan memilih menu artikel pada *navigation bar*. Kemudian memilih submenu *create*, untuk membuat artikel, submenu *list* artikel, untuk melihat daftar artikel, dan memilih *action edit*, untuk melakukan perubahan, atau *delete* untuk menghapus, dijelaskan dalam gambar 3.49 dan 3.50.



Gambar 3.49 Tampilan laman *create* artikel

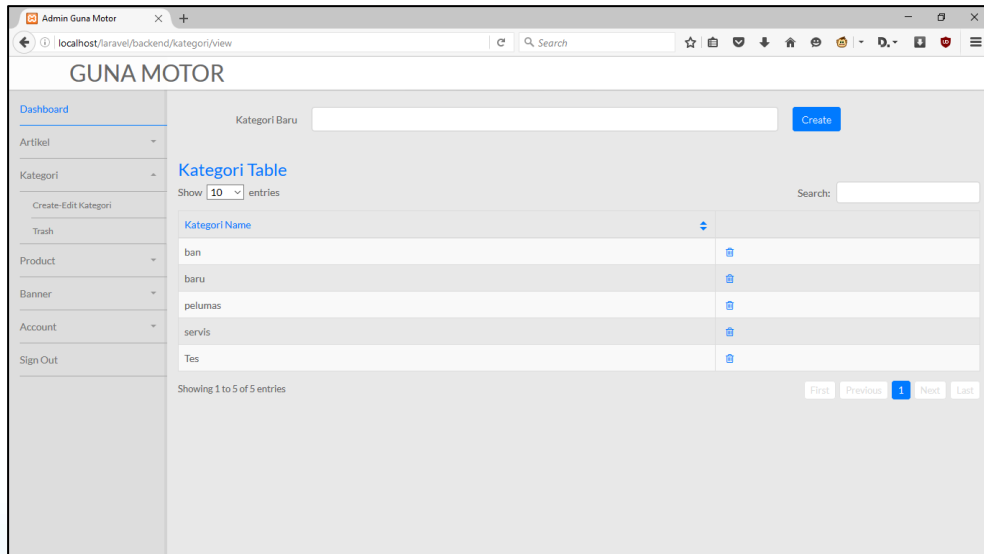


Gambar 3.50 Tampilan laman *list* artikel

B.3 Modul Kategori

Modul kategori digunakan untuk menciptakan kategori baru untuk *product*. Dalam modul ini admin dapat membuat, merubah, dan menghapus suatu kategori,

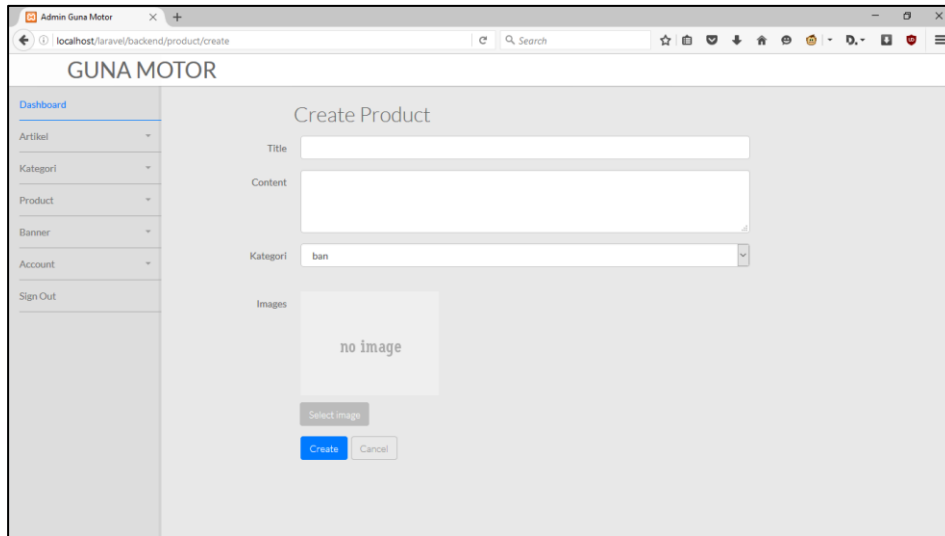
dengan langkah-langkah yang mirip dengan sebelumnya, dapat dilihat pada gambar 3.51.



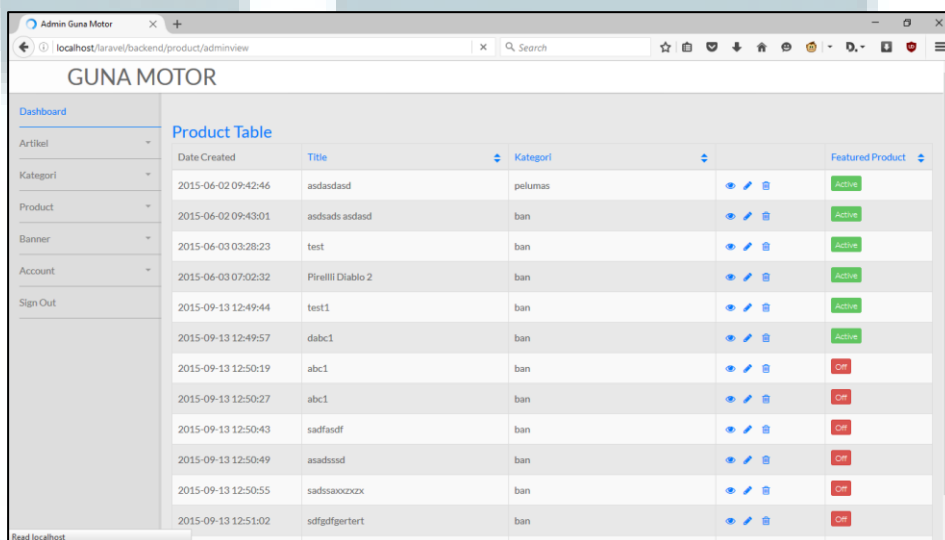
Gambar 3.51 Tampilan laman *create-edit* kategori

B.4 Modul Product

Modul *product* digunakan untuk menambahkan, merubah, atau menghapus suatu *product* dari/ke dalam sistem. Langkah-langkah untuk mengakses sama seperti pada modul sebelumnya, hanya saja di dalam modul *product* terdapat fitur *featured product* yang dapat diaktifkan dan dinonaktifkan pada suatu *product*, untuk lebih jelasnya dapat dilihat pada gambar 3.52 dan 3.53.



Gambar 3.52 Tampilan laman *create product*



Gambar 3.53 Tampilan laman *list product*

B.5 Modul Banner

Modul *banner* digunakan untuk menambahkan, merubah, atau menghapus suatu *banner* dari/ke dalam sistem. Langkah-langkah untuk mengakses sama seperti pada modul sebelumnya, hanya di dalam modul *banner* terdapat fitur *active banner*

yang dapat diaktifkan dan dinonaktifkan pada suatu *banner*, untuk lebih jelasnya dapat dilihat pada gambar 3.54.

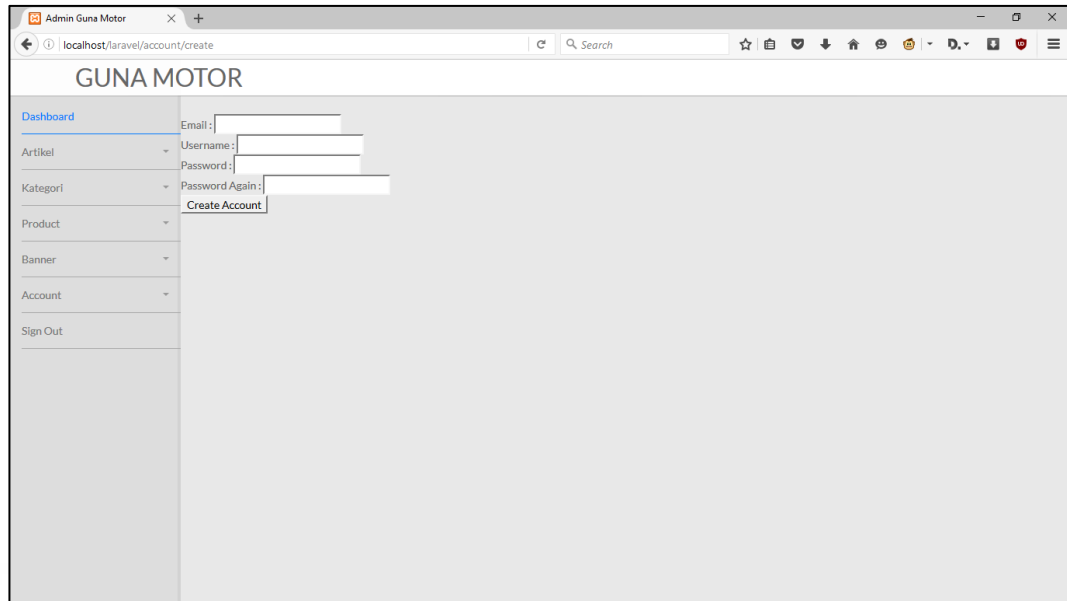
Date Created	Title	Active Banner
2015-06-03 11:14:57	123	Active
2015-08-05 20:53:54	aaaaacc	Active
2016-07-17 12:05:25	aaaa	Active
2016-07-17 12:05:37	dfdfd	Active
2016-07-17 12:05:47	ddd	Off

Gambar 3.54 Tampilan laman *list banner*

B.6 Modul Create Account

Modul *create account* digunakan untuk menambahkan *account admin* baru ke dalam sistem, langkah-langkah untuk mengakses sama seperti pada modul sebelumnya. Kemudian *admin* perlu untuk mengisi beberapa form berupa *text box*, untuk lebih jelasnya dapat dilihat pada gambar 3.55.

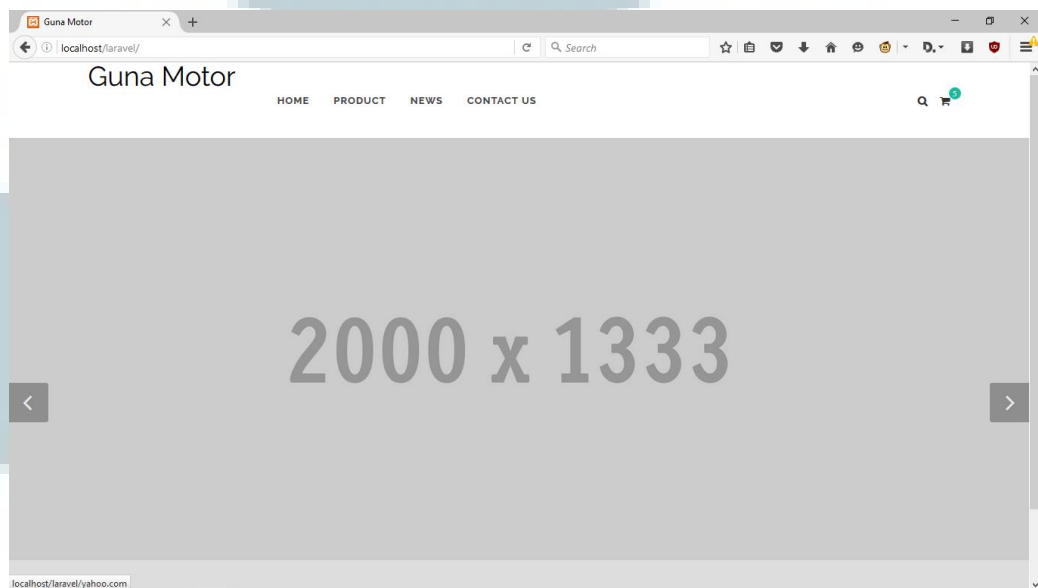
U
M
N



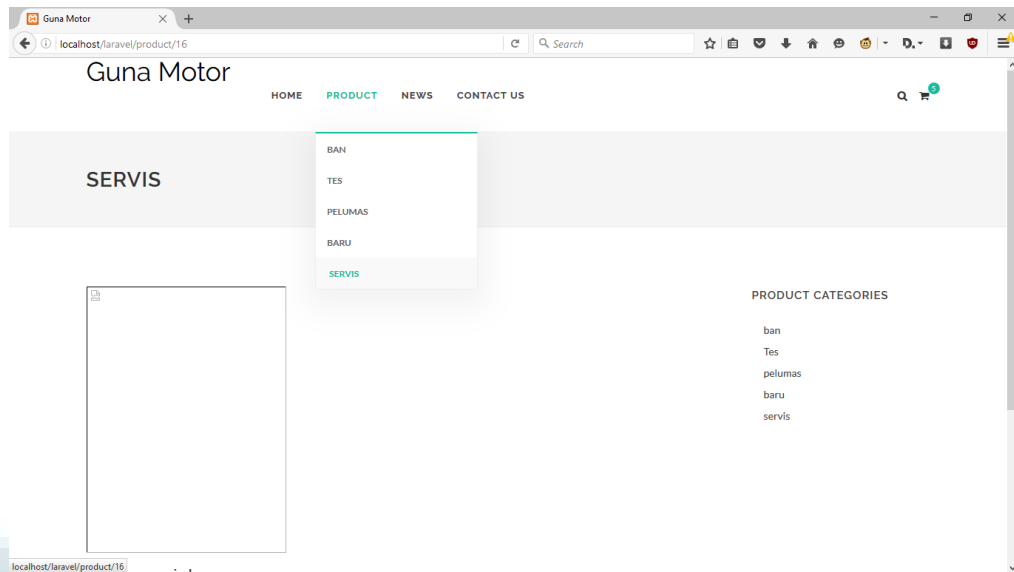
Gambar 3.55 Tampilan laman *create new account*

B.7 Tampilan Laman Home User

Tampilan yang dapat diakses *user* untuk menampilkan data-data yang telah dimasukkan sebelumnya dalam menu *admin*. Dapat dilihat pada gambar 3.56 dan 3.57.



Gambar 3.56 Tampilan laman *home user*



Gambar 3.57 Tampilan laman *product user*

C. Evaluasi Kerja Magang

Modul *login*, artikel, kategori, *product*, *banner*, dan *account* telah berhasil dibuat sesuai dengan kriteria yang telah diberikan. Modul yang sudah dibuat dapat berjalan dengan baik dan sesuai dengan fungsinya. Pembuatan laporan telah selesai dibuat dengan baik.

3.3.2 Kendala yang Ditemukan

Adapun beberapa kendala yang ditemukan ketika melakukan proses kerja magang di CV Guna Motor. Kendala tersebut diantaranya adalah:

1. Kurang nyamannya lingkungan kerja dikarenakan lokasi kantor sedang dalam renovasi besar,
2. Kesulitan dalam menyesuaikan diri dengan *framework* Laravel 4.2.

3.3.3 Solusi Atas Kendala yang Ditemukan

Solusi atas kendala yang ditemukan tersebut diantaranya adalah:

- 1) Berusaha untuk menyesuaikan diri dengan keadaan lingkungan kerja tersebut, misalnya dengan menggunakan masker ketika kondisi udara berdebu sekali,
- 2) Berusaha untuk lebih tekun dalam mempelajari hal-hal baru, dalam hal ini khususnya adalah *framework* Laravel, baik itu dengan referensi berupa bahan bacaan, maupun dari rekan kerja.

