

## BAB 3 PELAKSANAAN KERJA MAGANG

### 3.1 Kedudukan dan Organisasi

Pelaksanaan kerja magang di *Focus On the Family* Indonesia berada di dalam *department Digital Media* sebagai *full-stack developer*. *Full-stack developer* memiliki tugas untuk merancang sebuah aplikasi dimulai dari *Frontend*, hingga *Backend* untuk aplikasi penjualan *merch* yang berbasis *website* dengan menggunakan *framework* Laravel 10.

### 3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama proses *internship* di *Focus On the Family* Indonesia antara lain adalah

1. Membangun *workflow* dan *environment* kerja dengan menggunakan GitHub.
2. Memperbaiki *bug*.
3. Melakukan riset pengadaan fitur.
4. Mengembangkan fitur yang diperlukan oleh *staff*.
5. Membuat panduan penggunaan CMS.
6. Membuat buku panduan penggunaan *website* penjualan *merch*.
7. Merancang serta membangun sebuah aplikasi berbasis *website* untuk penjualan *merch*.

### 3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang di *Focus On the Family* Indonesia diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama magang

Minggu Ke-	Pekerjaan yang dilakukan
1	Melakukan <i>Onboarding</i> , Perkenalan dengan seluruh <i>staff</i> dan peserta magang lainnya, memahami <i>tech-stack</i> yang digunakan, melakukan <i>transfer knowledge</i> , melakukan diskusi dan riset.
2	Melakukan <i>meeting coordination</i> , bergabung ke dalam <i>repository</i> GitHub CMS, melakukan analisis terhadap <i>website</i> yang sudah ada, melakukan <i>technical meeting</i> dan melaksanakan <i>event</i> di SMAK 2 Penabur Jakarta.
3	Melakukan <i>meeting coordination</i> , menambahkan status done pada <i>progress form</i> , menambahkan <i>attribute</i> agama pada <i>database</i> dan <i>form</i> konseling, melakukan penyesuaian terhadap <i>detail</i> informasi didalam <i>form</i> , melakukan riset mengenai <i>server hosting</i> .
4	Melakukan <i>meeting coordination</i> , memaparkan hasil riset mengenai <i>server hosting</i> , melakukan riset mengenai pengadaan <i>merch store website</i> .
5	Melakukan <i>meeting coordination</i> , melaksanakan <i>event character building</i> di SMAK Penabur Summarecon Bekasi, membuat akun hostinger, Github, dan Cloudinary untuk keperluan <i>server hosting</i> , melakukan pemindahan <i>repository</i> ke organisasi yang sudah dibuat di GitHub, melakukan <i>server deployment</i> , <i>setting server database</i> , <i>setting company domain email</i> .
6	Melakukan riset mengenai <i>Continuous Integration and Continuous Delivery</i> atau (CI/CD), melaksanakan <i>event character building</i> di SMAK 6 Penabur Jakarta, mengimplementasikan CI/CD yang sudah di riset, melaksanakan <i>No Apologies</i> seminar di SMAK 2 Penabur Jakarta dan Penabur Harapan Indah Bekasi.
7	Melakukan <i>meeting coordination</i> , melakukan <i>update</i> pada <i>form</i> konseling, membuat <i>follow up question logic</i> pada pertanyaan status pernikahan.
8	Melaksanakan seminar mengenai <i>parenting: Rising Future Ready Kids</i> , melakukan <i>update</i> pada pertanyaan didalam <i>form</i> konseling, melakukan <i>debugging</i> mengenai logika yang digunakan pada pertanyaan di <i>form</i> konseling.
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu selama magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
9	Melakukan <i>meeting coordination</i> , melaksanakan seminar <i>Rising Future Ready Kids</i> , merilis <i>form</i> konseling yang sudah diperbaharui ke <i>development server</i> , melakukan <i>testing</i> pada <i>development server</i> . merilis <i>form</i> konseling ke <i>production server</i> , melakukan <i>testing</i> pada <i>production server</i> .
10	Melaksanakan <i>Marriage Seminar</i> di GPDI Hebron Gading Serpong, melaksanakan seminar <i>No Apologies</i> di SMAK Penabur Gading Serpong dan SMAK 2 Penabur Jakarta.
11	Melakukan riset mengenai <i>design detail product</i> , membuat repository untuk <i>merch store website</i> pada organisasi yang sudah dibuat di Github, melakukan <i>setup</i> dan <i>setting</i> untuk <i>tech-stack</i> yang akan digunakan, melakukan riset dan menginstall <i>framework</i> CSS yang akan digunakan.
12	Melakukan <i>meeting coordination</i> , merealisasikan <i>design website merch store</i> pada Figma ke dalam <i>codingan</i> , membuat halaman <i>HomePage</i> , <i>NavigationBar component</i> , dan <i>Footer component</i> , melakukan riset terhadap <i>design product card</i> , <i>login system</i> , membuat <i>ProductCard component</i> , membuat halaman <i>DetailProduct</i> .
13	Melakukan <i>meeting coordination</i> , melakukan revisi pada <i>merch store website</i> yang diberikan pada saat <i>meeting coordination</i> , membuat <i>database</i> untuk <i>users</i> , melakukan <i>migration</i> dan melakukan <i>testing</i> memasukan data ke dalam <i>database</i> , membuat <i>login controller</i> , merancang <i>login system logic</i> , membuat halaman <i>Login</i> , membuat halaman registrasi <i>user</i> , membuat <i>registration controller</i> , melakukan <i>testing</i> pada halaman <i>registration</i> .
14	Melakukan <i>meeting coordination</i> , melakukan <i>testing</i> pada <i>registration system</i> dan <i>login system</i> , mencari <i>bug</i> atau <i>error</i> , melakukan pembaharuan harga pada <i>form</i> konseling.
Lanjut pada halaman berikutnya	

Tabel 3.1 Pekerjaan yang dilakukan tiap minggu selama magang (lanjutan)

Minggu Ke-	Pekerjaan yang dilakukan
15	Melaksanakan seminar <i>Parenting</i> di GBI SLCC Gading Serpong, melaksanakan seminar <i>No Apologies</i> di Bandung selama dua hari, melakukan research dan membangun admin panel untuk <i>merch store project</i>

### 3.4 Perangkat Penunjang

Selama pelaksanaan kerja magang yang dilaksanakan di *Focus On the Family* Indonesia terdapat perangkat lunak dan perangkat keras yang digunakan untuk sebagai penunjang dalam penyelesaian pekerjaan magang. Berikut adalah daftar perangkat lunak yang digunakan.

1. Visual Studio Code v1.89.1
2. PHP v8.1.6
3. Laravel 10
4. Composer v2.4.2
5. Node v18.17.0
6. NPM v9.6.7
7. XAMPP Control Panel v3.3.0
8. Windows 10 Home Single Language v22H2
9. GitHub Desktop v3.3.18
10. Google Chrome v125.0.6422.78

Selain itu, berikut ini adalah daftar perangkat keras yang digunakan selama pelaksanaan kerja magang.

1. ASUS TUF GAMING A15 FX506II
2. AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz

3. Nvidia GeForce GTX 1650 Ti 4 GB
4. RAM 16 GB DDR4-3200
5. 512 GB PCIe SSD

### 3.5 Proses Pelaksanaan Kerja Magang

*Focus On the Family* Indonesia memproduksi *merch* untuk salah satu program yang bernama *No Apologies*. *Merch* tersebut dijual untuk umum dan FOFI ingin semua lini masyarakat dapat membelinya. Sampai saat ini FOFI hanya menjual *merch* dari *platform* media sosial, dan dari mulut ke mulut saja. Untuk meningkatkan persentasi pemasaran dan penjualan, FOFI membutuhkan suatu *platform* untuk menjual *merch* yang mereka produksi. *Platform* tersebut akan memiliki dua sisi yaitu sisi *End-User* atau calon pembeli, dan juga sisi Admin.

Berikut ini adalah fitur-fitur yang dapat diakses oleh *End-User* atau calon pembeli.

1. Mengakses halaman utama (HomePage)
2. Mengakses halaman product (DetailProductPage)
3. Melihat keranjang belanja (CartPage)
4. *Add to Cart*
5. *Buy Product*
6. *Login*
7. *Register*
8. *Payment*
9. *Logout*

Untuk Admin, berikut adalah fitur-fitur yang dapat diakses oleh Admin melalui panel admin.

1. *Login*
2. *Add Product*

3. *Product Control*

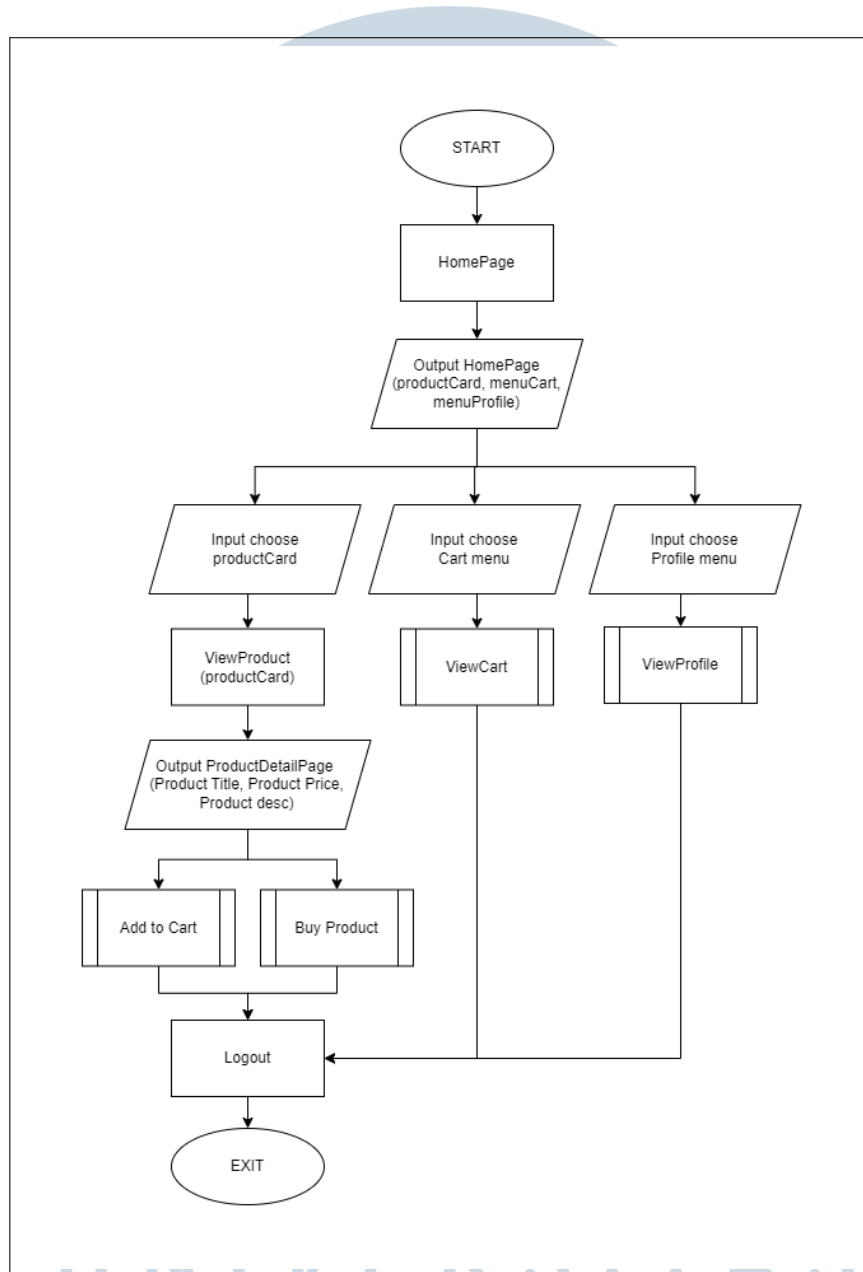
4. *Manage Order*

5. *Logout*

Proses pengerjaan *project merch store* mulai dikerjakan semenjak *design User Interface/User Experience (UI/UX)* pada Figma selesai dikerjakan. Proses pertama yang dilakukan adalah pembentukan arsitektur atau kanvas pengerjaan. Proyek *merch store* ini dibangun dengan menggunakan Laravel 10 dan Inertia.js sebagai *library* pendukungnya. Laravel adalah sebuah *framework* PHP yang banyak digunakan untuk membangun sebuah aplikasi berbasis *website* dan Inertia.js adalah sebuah *library* yang memungkinkan *developer* untuk membangun aplikasi *Single Page Application (SPA)* dengan menggunakan *framework server-side* seperti Laravel. Inertia.js berperan sebagai jembatan yang menghubungkan *server-side framework* seperti Laravel dengan *client-side framework* seperti React, Vue.js, atau Svelte. Pada proyek *merch store*, arsitektur yang akan digunakan adalah Laravel dan React yang dihubungkan dengan menggunakan Inertia.js.



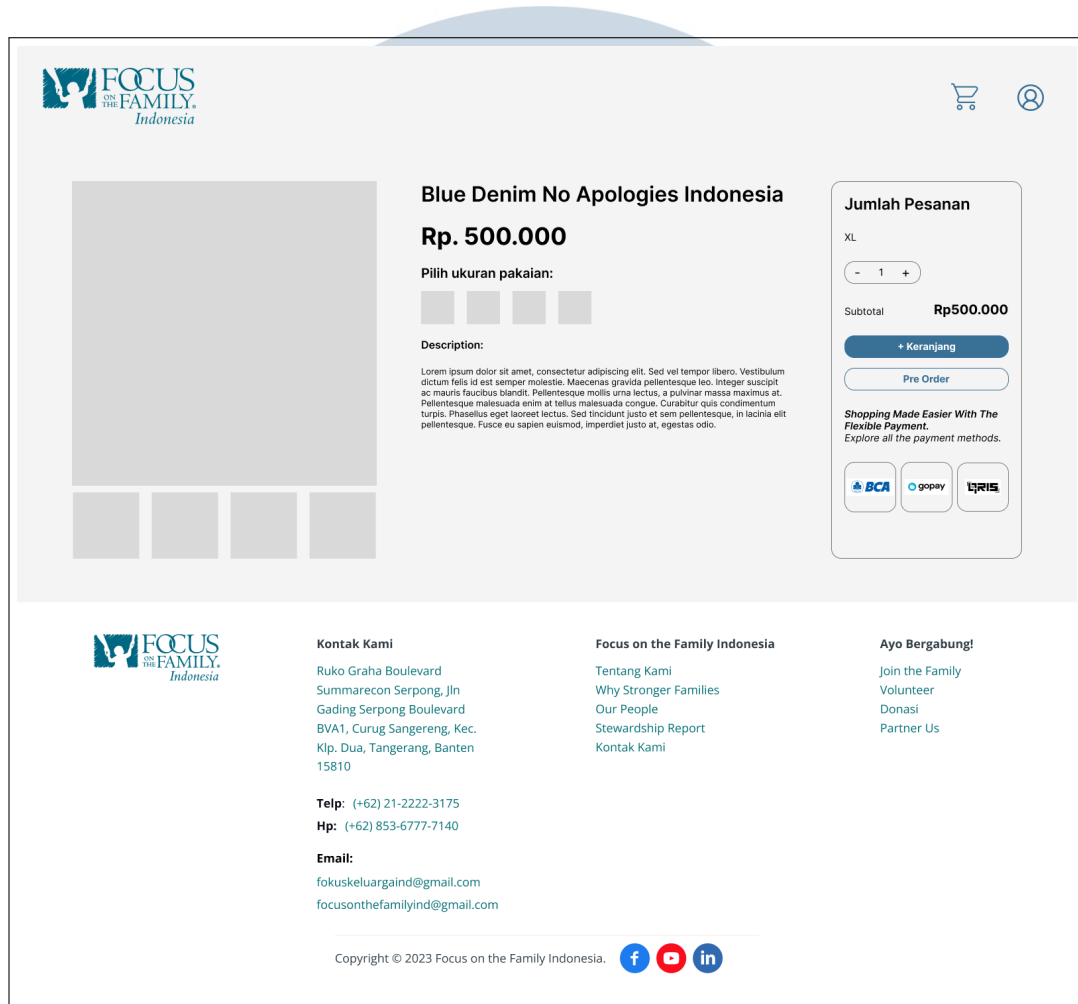
### 3.5.1 Proyek Pembangunan Aplikasi *Merch Store* Pada Sisi Pengguna



Gambar 3.1. Flowchart Merch Store Sisi Pengguna

Gambar 3.1 adalah sebuah gambaran terhadap *flow* yang dapat dilakukan oleh *End-User*. Pada tampilan *HomePage*, *End-User* dapat melakukan tiga aksi yaitu *ViewProduct*, *ViewCart*, dan *ViewProfile*.

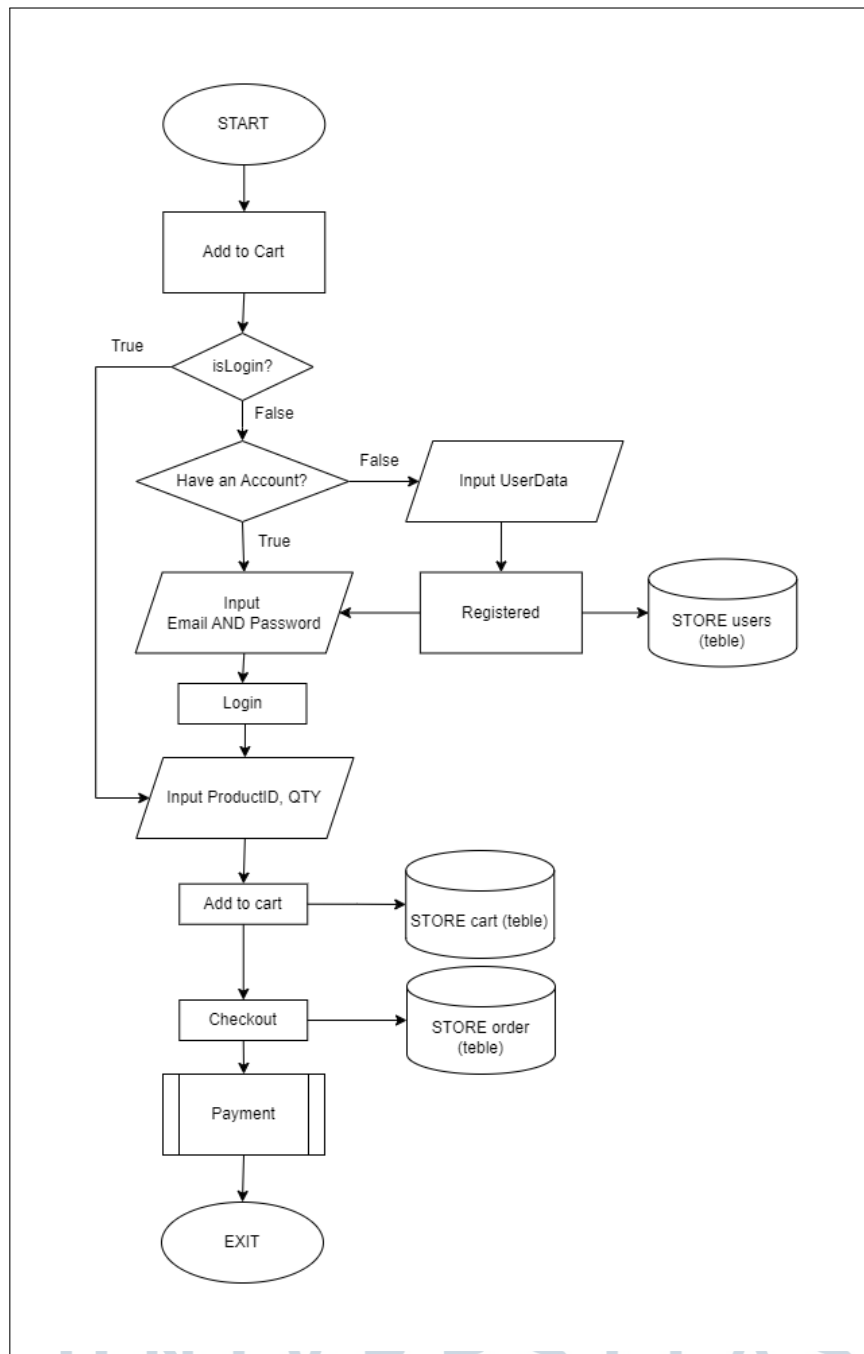
## A. Aksi ViewProduct



Gambar 3.2. *Prototype* Halaman *Detail Product*

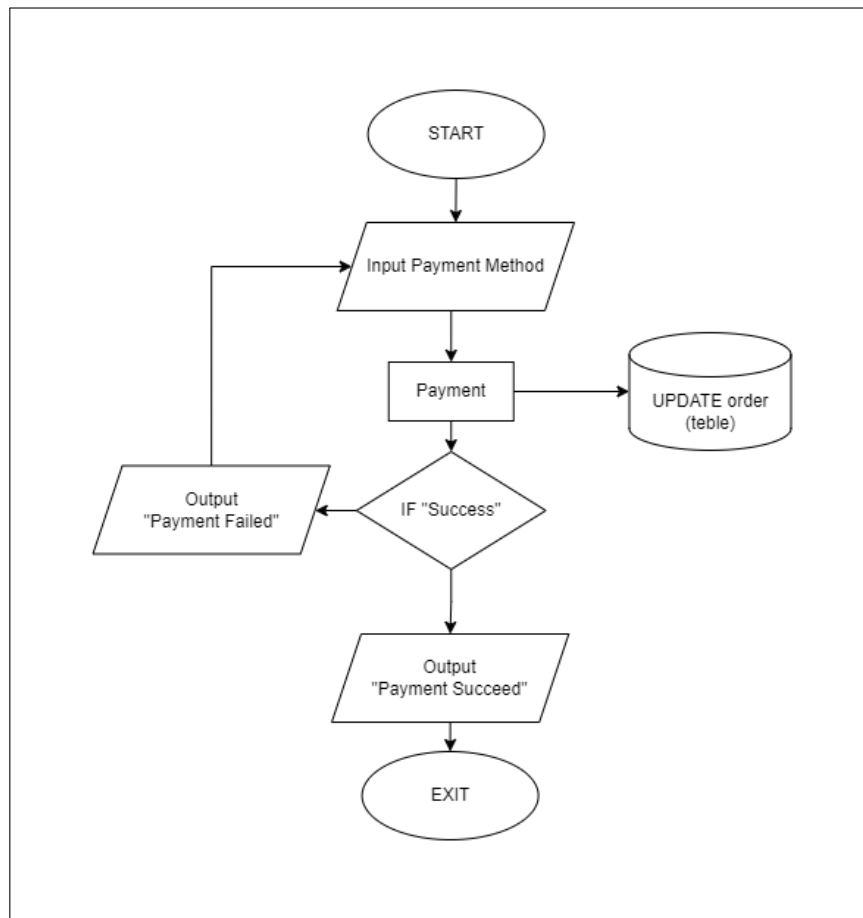
Pada aksi ViewProduct, *End-User* akan dibawa ke halaman *detail product* seperti pada Gambar 3.2. Pada halaman tersebut *End-User* dapat melihat *detail* foto produk, nama produk, harga produk, ukuran produk, dan deskripsi produk. *End-User* juga dapat melakukan beberapa aksi yaitu *Pre-Order* atau *Buy Product* dan *Add to Cart*.





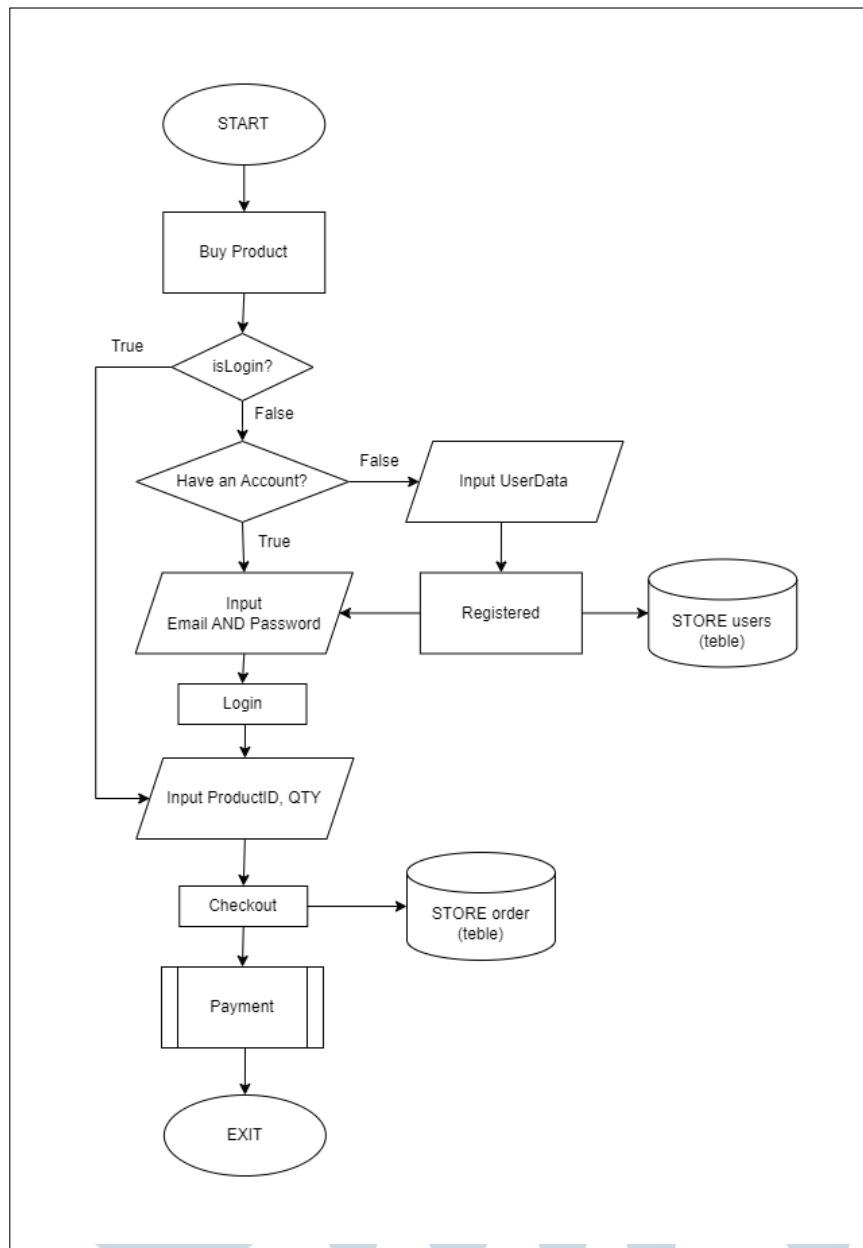
Gambar 3.3. *Prototype Add to Cart Flowchart*

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.4. *Prototype Payment Flowchart*

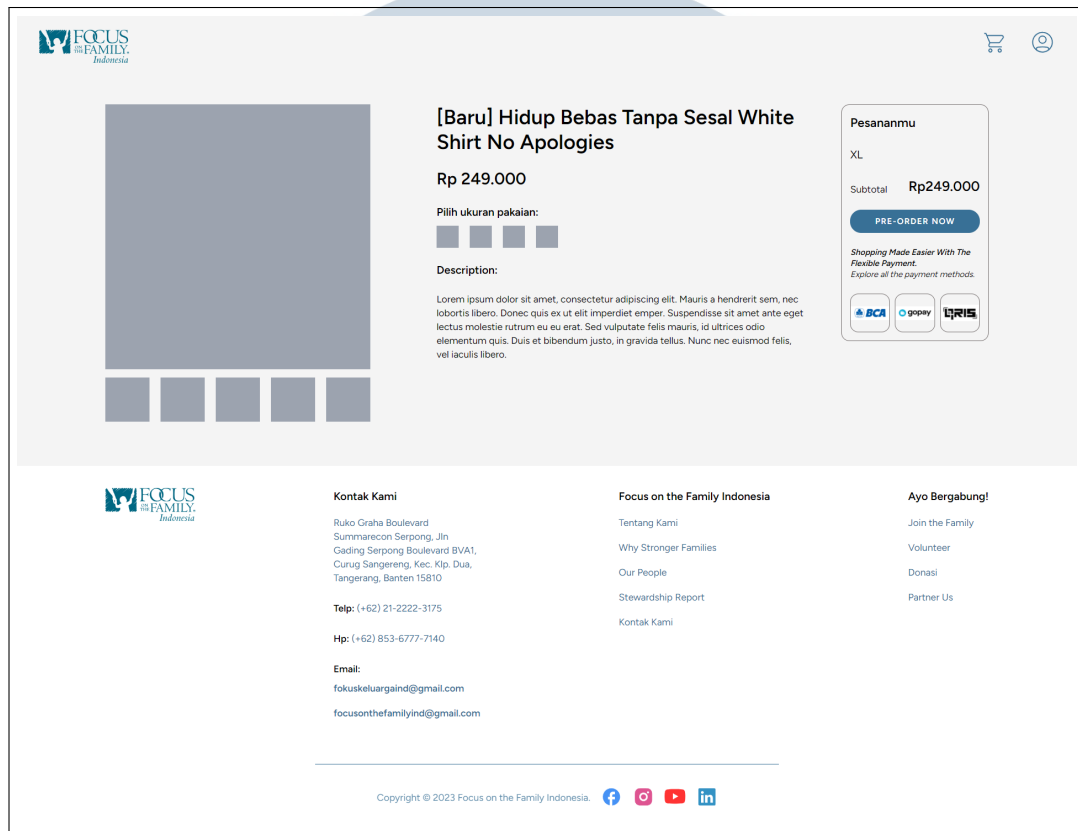
Gambar 3.4 merupakan alur yang akan dijalankan oleh pengguna jika ingin melakukan aksi *Add to Cart*. Untuk melakukan aksi tersebut pengguna harus melakukan *login*, dan jika pengguna belum memiliki akun, maka pengguna akan diarahkan ke halaman *register* untuk melakukan pendaftaran akun. Setelah berhasil mendaftar, pengguna akan diarahkan lagi untuk melakukan *login*. Ketika produk sudah masuk ke dalam keranjang belanja, maka sistem akan melakukan aksi *STORE* data ke dalam *database*. Setelah berhasil melakukan aksi *Add to Cart*, pengguna dapat keluar atau melakukan aksi *Checkout*. Pada saat pengguna melakukan aksi *Checkout*, sistem akan melakukan aksi *STORE* data ke dalam *database* dengan status pembayaran *Not Process*. Apabila pengguna sudah melakukan pembayaran maka sistem akan melakukan aksi *UPDATE* data pada *database* dengan mengubah status pembayaran menjadi *Done* dan pengguna akan kembali ke halaman utama.



Gambar 3.5. *Prototype Buy Product Flowchart*

Gambar 3.5 adalah *flow* ketika pengguna ingin melakukan aksi *Buy Product*. Sistem akan memeriksa apakah pengguna sudah melakukan *login* atau belum. Apabila belum dan pengguna belum mempunyai akun, maka pengguna akan diarahkan ke halaman *register* untuk melakukan pendaftaran akun. Ketika pendaftaran berhasil sistem akan memasukan data user ke dalam *database* dan mengarahkan pengguna ke halaman *login* untuk melakukan *login*. Ketika sudah melakukan *login*, pengguna dapat melakukan *checkout* dan sistem akan mencatat data produk ke dalam *database* dengan status pembayaran *Not Process*, apabila

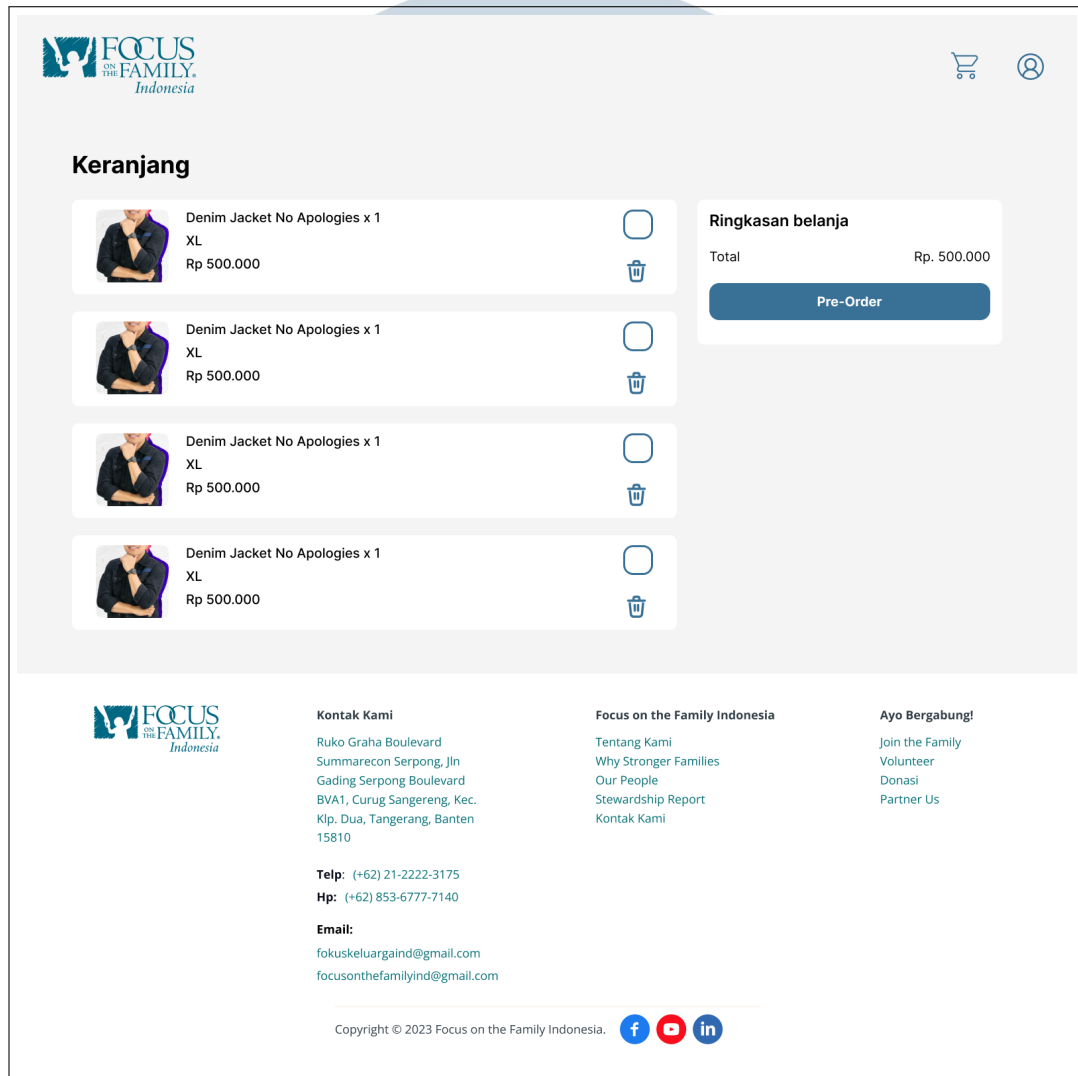
aksi *Payment* sudah berhasil maka sistem akan melakukan *UPDATE* pada *database* dengan merubah status pembayaran menjadi *Done*.



Gambar 3.6. Implementasi Halaman *Detail Product*

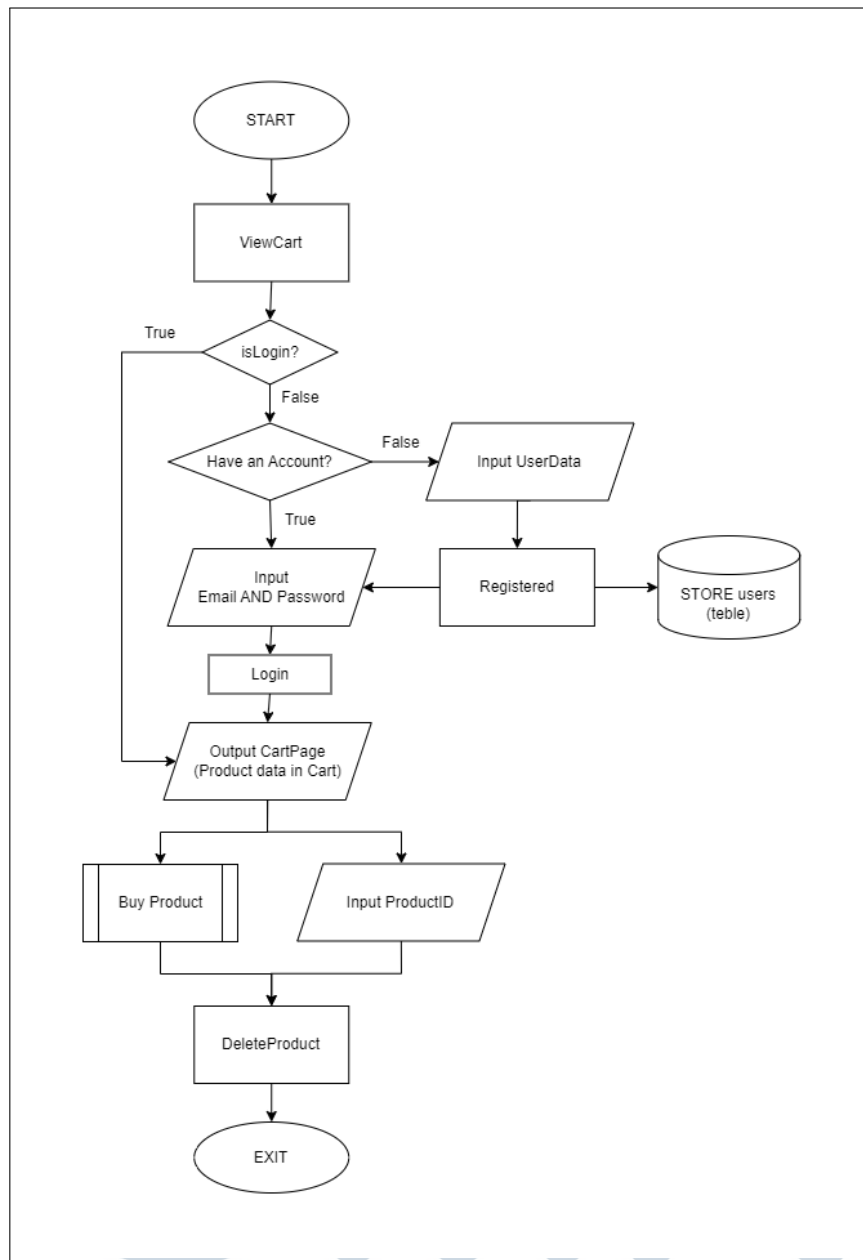
UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## B. *Prototype Aksi ViewCart*



Gambar 3.7. *Prototype Halaman Cart*

Gambar 3.7 merupakan tampilan dari halaman *Cart* yang dapat dilihat oleh pengguna. Pada halaman ini pengguna dapat melihat produk-produk yang sudah berhasil dimasukkan ke dalam keranjang belanja. Pengguna juga dapat melihat beberapa *detail* produk seperti nama produk, kuantitas produk, ukuran produk, dan harga produk.

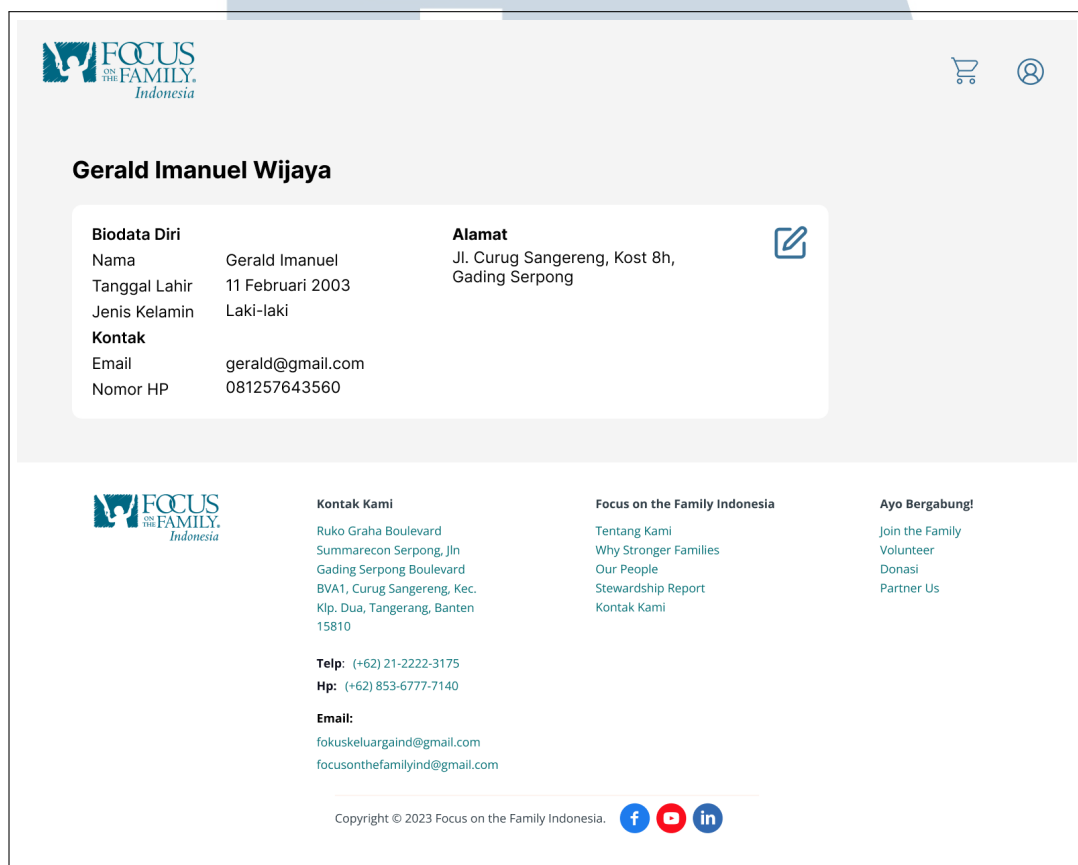


Gambar 3.8. *Prototype ViewCart Flowchart*

Pada Gambar 3.8 ditunjukkan *flow* ketika pengguna melakukan aksi ViewCart. Pertama pengguna akan dipastikan sudah melakukan *login*, apabila belum maka sistem akan mengarahkan pengguna untuk melakukan *login*. Jika pengguna belum memiliki akun maka sistem akan mengarahkan pengguna ke halaman *register* untuk melakukan registrasi akun dan ketika pengguna sudah berhasil membuat akun sistem akan mengarahkan pengguna untuk melakukan *login* lagi. Pengguna dapat melakukan beberapa aksi ketika berada pada halaman *cart* seperti menghapus produk, dan juga melakukan aksi *checkout* produk. Ketika

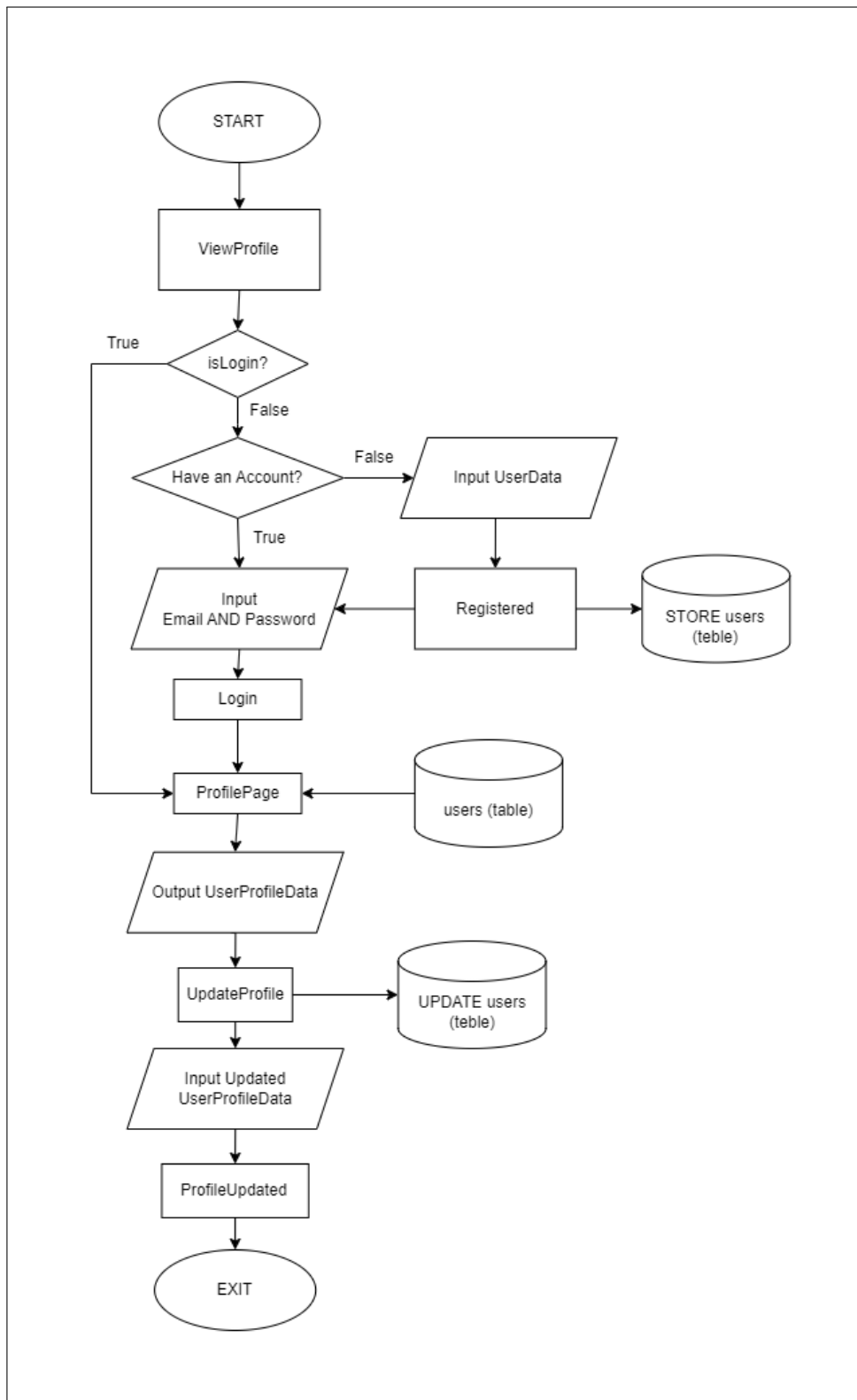
pengguna melakukan aksi menghapus produk, maka sistem akan menjalankan perintah *DELETE* data pada produk yang dipilih dan menghapus dari *database* pada *cart table*. Ketika pengguna melakukan aksi *Buy Product* maka proses yang akan berjalan sama seperti pada sub sub bab A di dalam sub bab 3.5.1.

### C. Aksi ViewProfile



Gambar 3.9. *Prototype* Halaman *Profile*

Gambar 3.9 merupakan tampilan yang dilihat pengguna jika melakukan aksi ViewProfile. Ketika pengguna menjalankan aksi ViewProfile, pengguna akan di arahkan sistem ke halaman profile dan pengguna dapat melihat data-data pengguna seperti nama, tanggal lahir, jenis kelamin, email, nomor telepon dan alamat. Pada aksi ViewProfile pengguna hanya dapat melakukan aksi mengedit biodata diri dan alamat. Pada proses tersebut dijelaskan dengan lebih detail menggunakan *flowchart*.

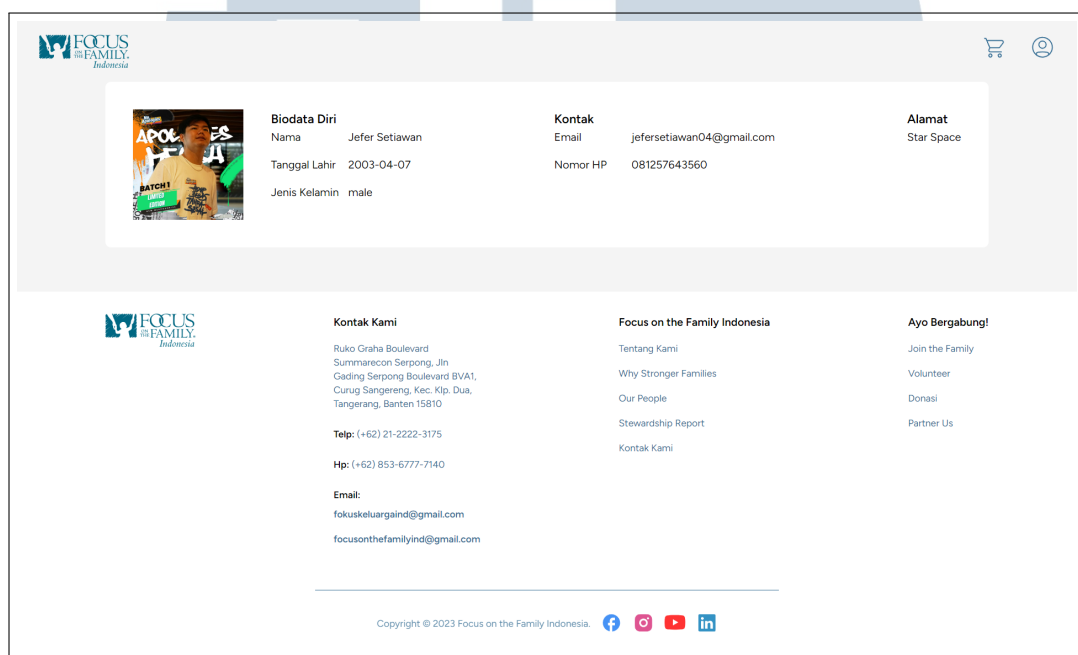


Gambar 3.10. Flowchart ViewProfile

Pada Gambar 3.10 dijelaskan ketika pengguna ingin melakukan aksi



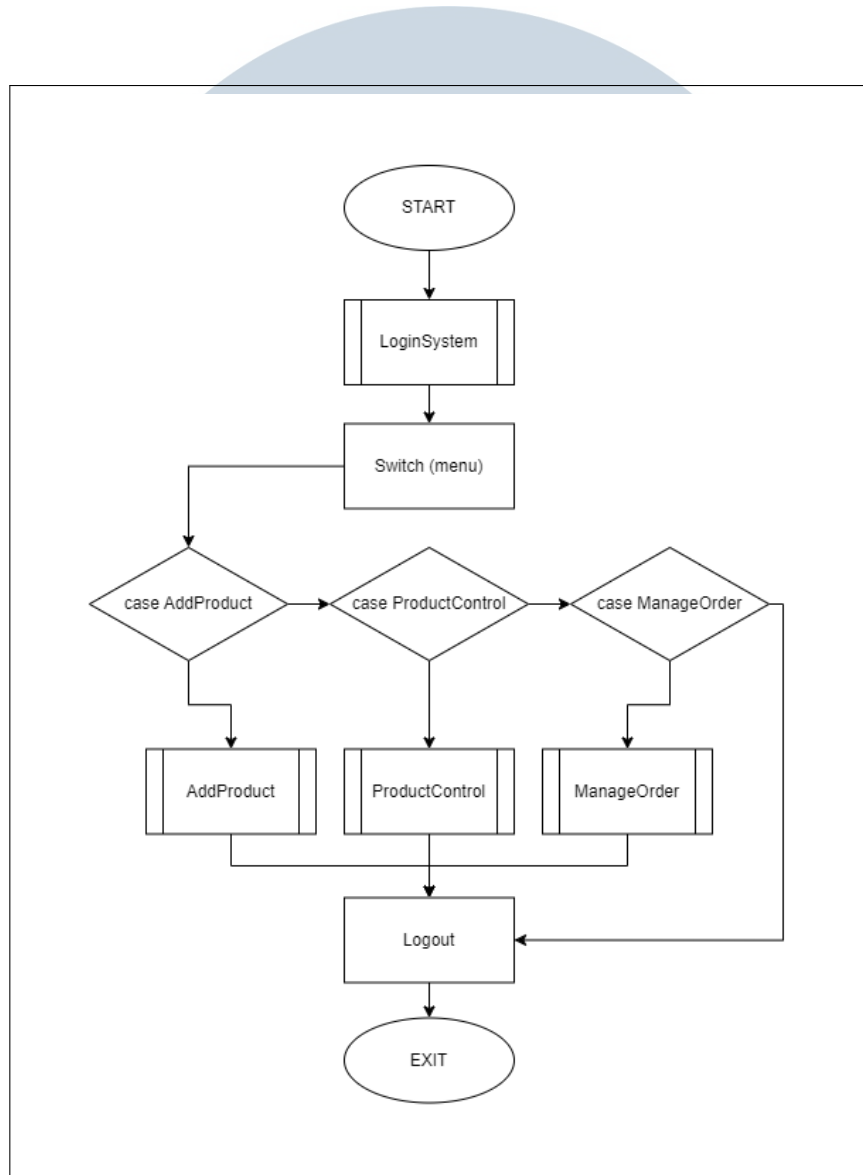
ViewProfile maka pengguna harus melakukan *login* terlebih dahulu. Apabila pengguna belum *login* sistem akan mengarahkan pengguna untuk melakukan *login* dan apabila pengguna belum memiliki akun maka akan diarahkan untuk melakukan register kemudian diarahkan lagi untuk melakukan *login*. Ketika pengguna sudah melakukan *login*, sistem akan memperoleh data dari *database* dan menampilkan data tersebut ke pengguna. Pengguna juga dapat melakukan aksi edit terhadap data tersebut. Ketika proses edit sudah selesai, sistem akan melakukan *UPDATE* pada data tersebut di *database*.



Gambar 3.11. Implementasi Halaman *Profile*

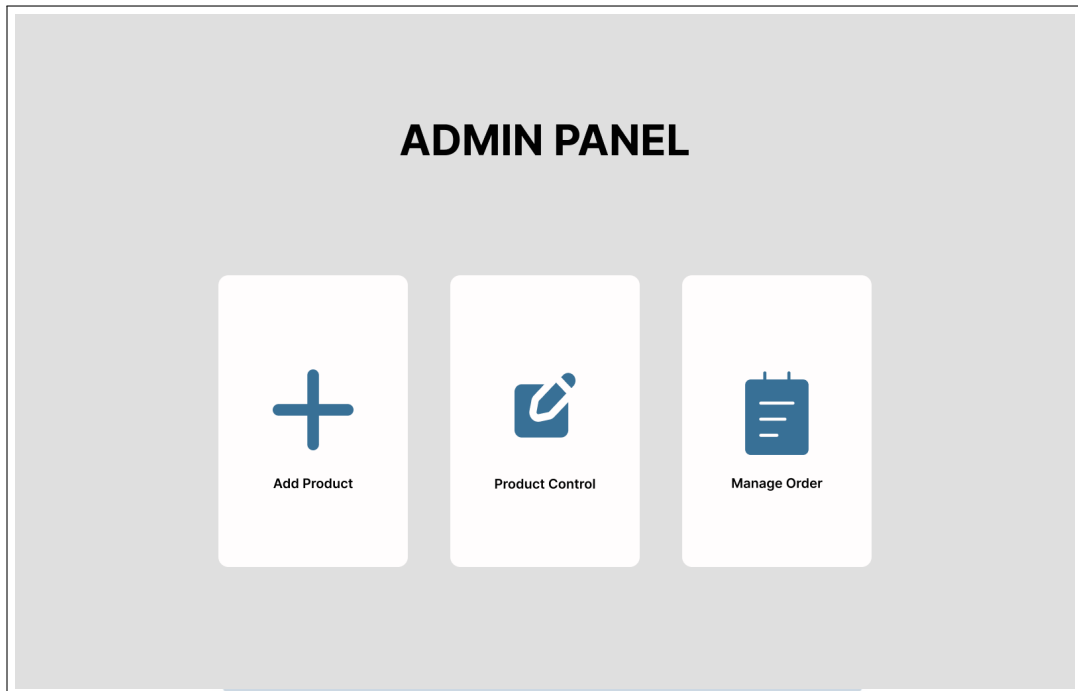
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

### 3.5.2 Rancangan Proyek Pembangunan Aplikasi Merch Store Pada Sisi Admin



Gambar 3.12. *Prototype Flowchart* Pada Sisi Admin

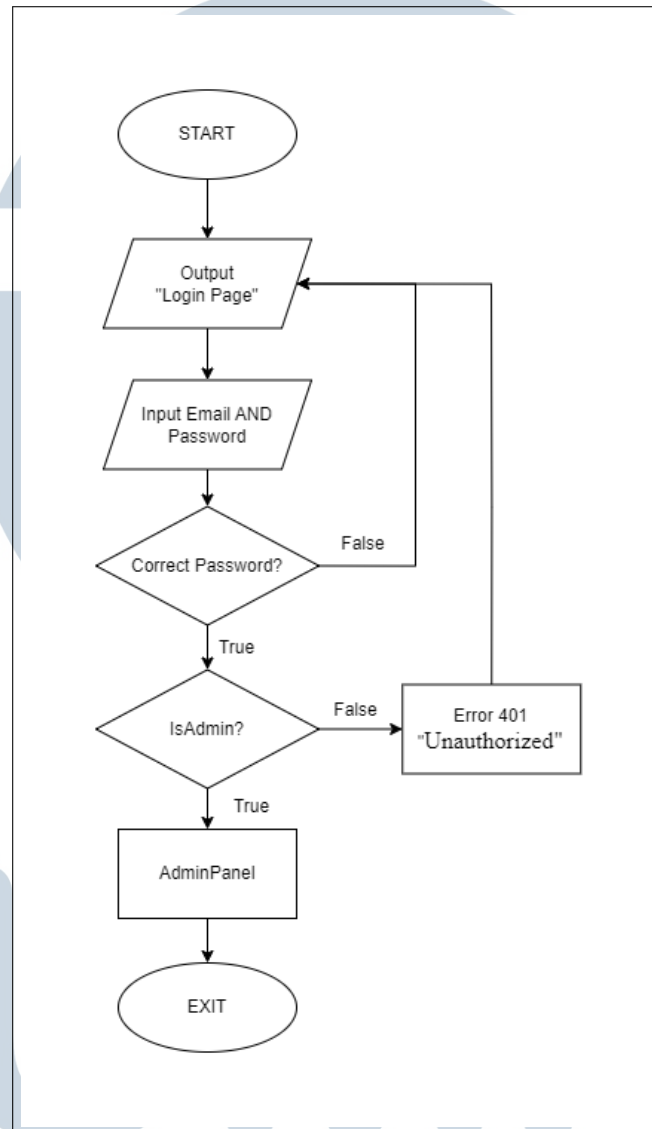
Sisi admin merupakan sisi yang digunakan oleh *staff* kantor untuk mengatur konten yang akan ditampilkan pada halaman utama *merch store*. Gambar 3.12 merupakan *flow* dari proses yang dapat dijalankan oleh admin. Pada saat pertama kali memasuki *admin panel*, admin akan diminta untuk melakukan *login*. Setelah melakukan *login* admin dapat melakukan tiga aksi, yaitu *AddProduct*, *ProductControl* dan *ManageOrder*. Gambar 3.13 adalah tampilan dari admin panel yang dilihat oleh admin ketika berhasil melakukan *login*.



Gambar 3.13. *Prototype Halaman Admin Panel*

UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

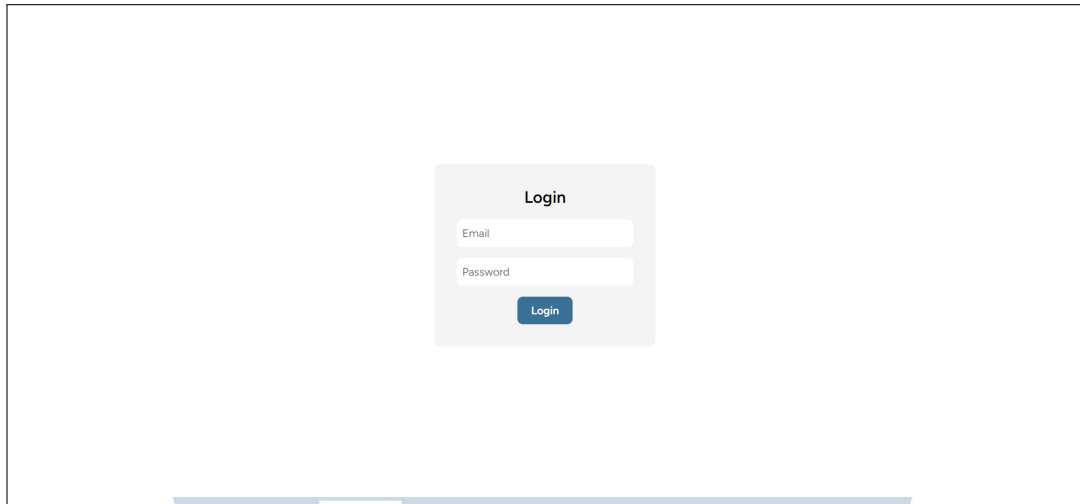
### A. Rancangan Admin Login System



Gambar 3.14. Prototype Flowchart Admin Login Sistem

Gambar 3.14 menunjukkan proses *login* pada admin panel ketika admin ingin melakukan *login*. Pertama admin akan diminta untuk memasukan *email* dan *password* dan sistem akan melakukan pemeriksaan pada *database users*. Apabila *password* yang dimasukkan salah maka admin akan diarahkan kembali ke halaman *login*, tapi jika admin memasukan akun dengan benar, sistem akan melakukan pemeriksaan *role* pada akun tersebut. Jika *role* pada akun tersebut adalah *user* maka sistem akan menunjukkan kode error 401:*Unauthorized* tapi jika *role* pada akun tersebut adalah *admin* maka pengguna akan diarahkan ke halaman AdminPanel

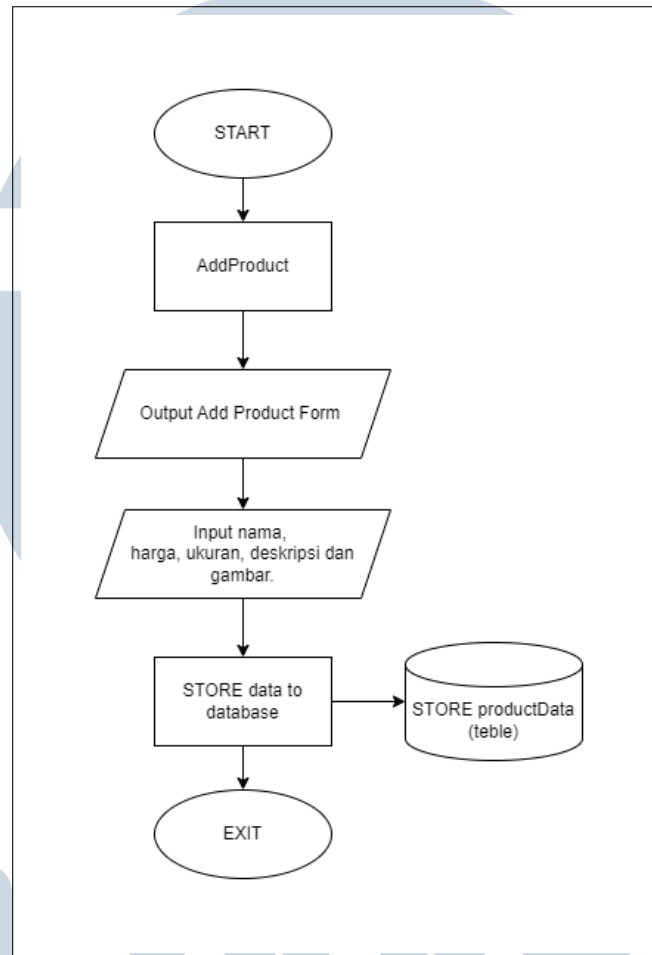
seperti pada Gambar 3.15.



Gambar 3.15. *Prototype Halaman Login Admin Panel*

UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

## B. Rancangan Fitur *Add Product*



Gambar 3.16. *Prototype Add Product Flowchart*

Fitur *add product* adalah fitur yang dapat digunakan oleh admin untuk menambah produk baru ke dalam *database*. Data yang masuk ke dalam *database* nantinya akan ditampilkan pada halaman utama agar pengguna dapat melihat produk terbaru tersebut. Dapat dilihat pada Gambar 3.16, admin akan diminta memasukkan sejumlah data seperti nama, harga, ukuran, deskripsi, dan juga gambar produk. Setelah admin melakukan *input* tersebut, sistem akan memasukkan data tersebut ke dalam *database*. Gambar 3.17 merupakan tampilan yang dilihat admin ketika melakukan aksi *add product*.

←

### Add Product

Nama Produk

Rp.

Deskripsi

Ukuran:

S

M

L

XL

Upload Photo

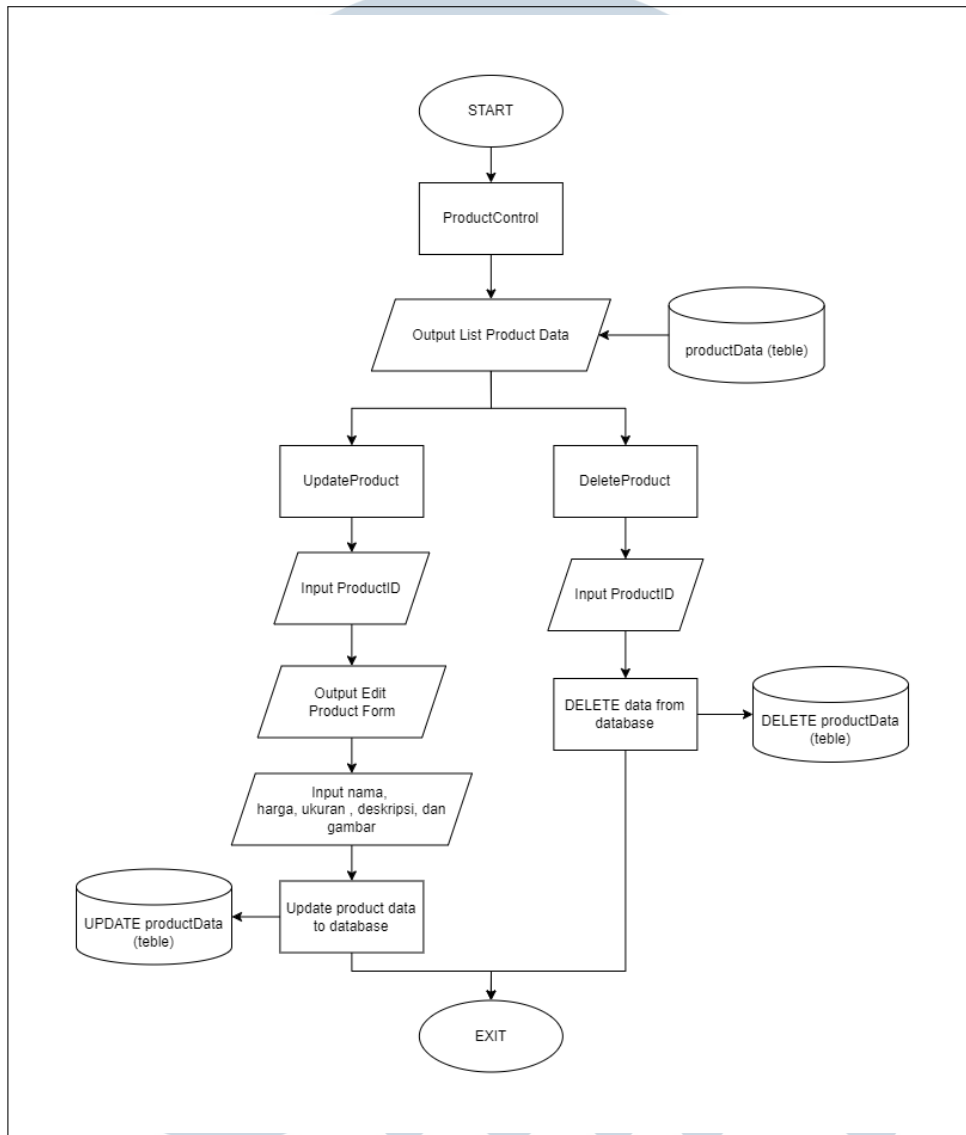
Choose File

Add Product

Gambar 3.17. *Prototype Halaman Add Product*

UIN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### C. Rancangan Fitur *Product Control*

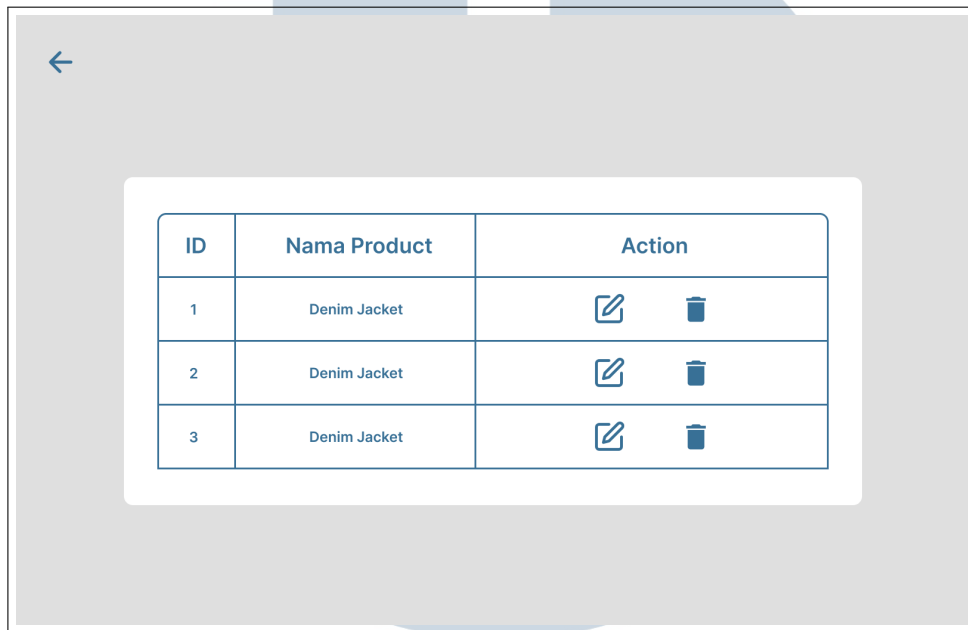


Gambar 3.18. *Prototype Product Control Flowchart*

Fitur *product control* adalah fitur yang dapat digunakan oleh admin untuk menghapus produk yang sudah tidak dijual lagi atau produk yang sudah tidak ada dan untuk melakukan update pada data produk. Pada Gambar 3.18 dijelaskan alur yang terjadi apabila admin melakukan aksi *delete product*. Pertama admin akan diberikan sejumlah *list* produk yang terdapat di dalam *database*. setiap *list* produk akan terdapat *button* atau tombol *delete*, admin hanya perlu menekan tombol *delete* pada *list* produk yang ingin dihapus. Ketika admin menekan tombol tersebut, sistem akan melakukan aksi *delete* pada produk di dalam *database*. Admin juga dapat

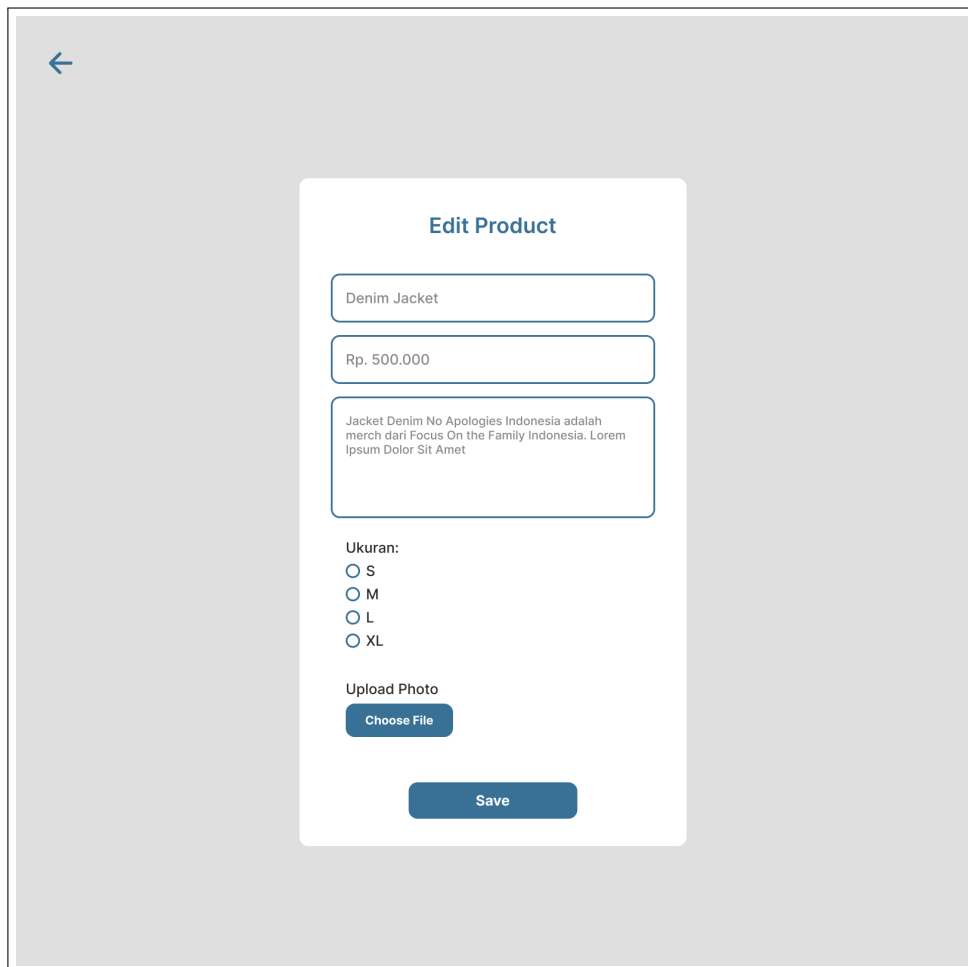


melakukan aksi edit pada produk yang dipilih. ketika admin memilih salah satu produk, maka admin dapat mengedit data pada produk tersebut. Setelah selesai mengedit, sistem akan melakukan *UPDATE* pada data produk tersebut di dalam *database*. Gambar 3.20 merupakan tampilan dari halaman *edit product*.

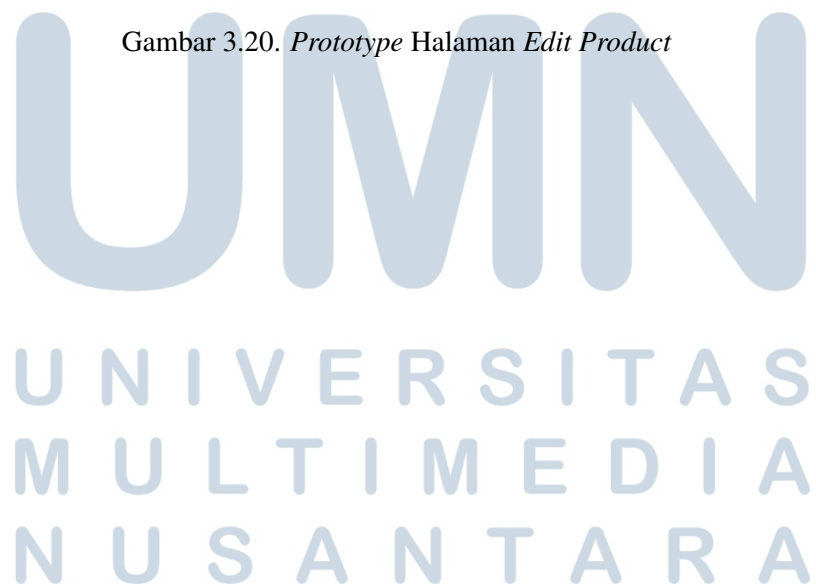


Gambar 3.19. *Prototype* Halaman *Product Control*

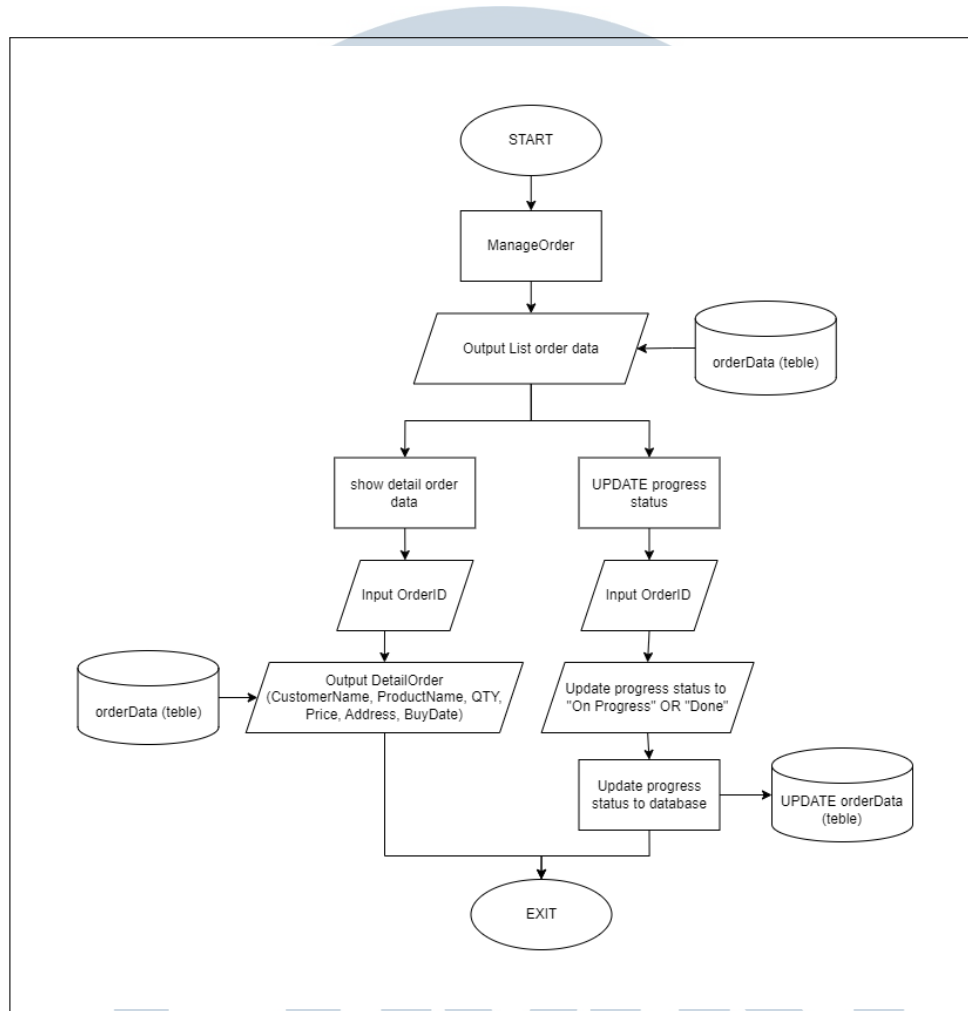
UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.20. *Prototype Halaman Edit Product*



#### D. Rancangan Fitur *Manage Order*



Gambar 3.21. *Prototype Manage Order Flowchart*

Fitur *manage order* adalah fitur yang dapat digunakan oleh admin untuk memantau pesanan yang masuk dan melakukan *update* pada progres status pesanan. Gambar 3.21 menjelaskan alur yang terjadi dan aksi yang dapat dilakukan oleh admin. Ketika memilih menu *manage order* admin akan disajikan dengan *list* pesanan yang sudah tercatat di dalam *database*. Admin juga dapat melakukan beberapa aksi terhadap *list* yang ditampilkan, seperti *show detail order* dan *update progress status*.

ID ORDER	CUSTOMER NAME	PRODUCT NAME	PAYMENT STATUS	PROGRESS STATUS	ACT
1	Gerald Imanuel	Denim Jacket	DONE	Not Process	⋮
2	Jefer Setiawan	Denim Jacket	DONE	Not Process	⋮
1	Gerald Imanuel	Denim Jacket	DONE	Done	⋮

Gambar 3.22. *Prototype Halaman Manage Order*

Pada aksi *show detail order* sistem akan menampilkan semua data mengenai pesanan seperti nama pembeli, tanggal pembelian, produk yang dibeli, jumlah pembelian, ukuran yang dibeli, harga, alamat pembeli, status pembayaran dan *progress status*. Pada aksi *update progress status* admin dapat mengubah *progress status* suatu pesanan dari *not process* menjadi *process* sampai *done* untuk menandakan bahwa pesanan belum diproses, sedang diproses atau sudah diproses.

ID ORDER: 1	STATUS	Alamat
Customer Name : Gerald Imanuel	Payment Status : Done	Jl. Curug Sangereng, Kost 8h Gading Serpong
Order Date : 20 April 2023	Progress Status : Not Process	
Product Name : Denim Jacket		
Quantity : 1		
Size : XL		
Total Price : Rp. 500.000		

Gambar 3.23. *Prototype Halaman Detail Order*

### 3.6 Kendala dan Solusi yang Ditemukan

Pada saat pelaksanaan kerja magang berlangsung, ada kendala yang terjadi. Berikut merupakan kendala yang dialami selama pengerjaan proyek atau pelaksanaan kerja magang berlangsung.

1. Tidak memiliki *environment* kerja yang baik, sehingga proses pengenalan dan *transfer knowledge* berlangsung lama.
2. Kekurangan tim dalam membangun *website* penjualan *merch*.
3. Pemilihan *framework* untuk membangun sebuah aplikasi *merch store* berbasis *website* yang bisa memenuhi semua kebutuhan dalam pembangunan sebuah aplikasi berbasis *website*.

Berdasarkan kendala-kendala yang sudah disebutkan, berikut merupakan solusi dari kendala-kendala yang terjadi selama proses pelaksanaan kerja magang.

1. Melakukan pengenalan dan mempelajari *environment* kerja sendiri di luar jam kerja, dan mengusulkan untuk membangun sistem kerja yang baik untuk kedepannya.
2. Melakukan semua pekerjaan sendiri sampai lembur dan bekerja di luar jam kerja.
3. Melakukan *research* terhadap berbagai macam *framework* yang dapat digunakan dengan target mendapatkan *framework* yang bisa memenuhi kebutuhan yang diperlukan. Didapatkan *framework* Laravel yang kemudian ditambahkan dengan *library* Inertia.js sebagai penghubung antara *server-side framework* (Laravel) dan *client-side framework* (React).

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