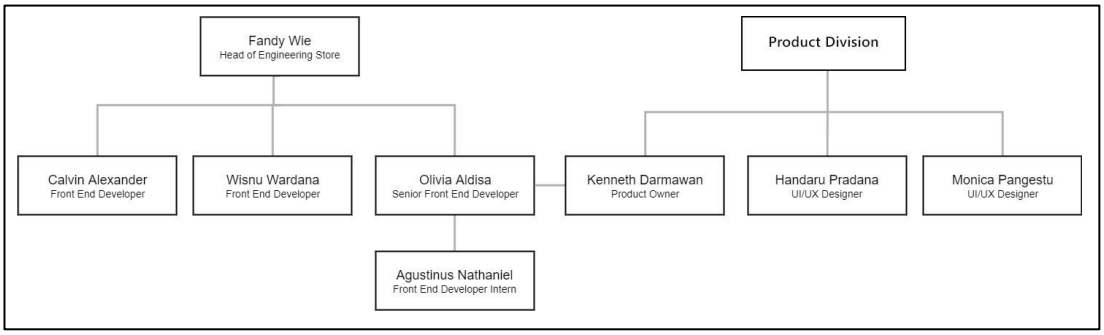
BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi



Gambar 3.1 Kedudukan pelaksana magang

Pelaksanaan kerja magang sebagai Front End Developer Intern dilakukan di bawah bimbingan Ibu Olivia Aldisa selaku Senior Front End Developer. Selain itu, dalam pengembangan *library* komponen SIRCLO-TDK dan template STARTER-TDK, pelaksana magang berkoordinasi dengan Bapak Kenneth Darmawan selaku Product Owner, Bapak Handaru Pradana dan Ibu Monica Pangestu selaku UI/UX Designer, serta Bapak Wisnu Wardana dan Bapak Calvin Alexander selaku Front End Developer.

3.2 Tugas yang Dilakukan

Selama pelaksanaan kerja magang, tugas dan tanggung jawab yang diberikan adalah sebagai berikut:

a. Terlibat dalam Project SIRCLO-TDK (Template Development Kit) Next Generation yang digunakan sebagai library komponen untuk template engineer dalam membuat template.

b. Membuat komponen yang dapat melakukan input dan edit data menggunakan ReactJS.

c. Menampilkan data dari database menggunakan ReactJS.

d. Menggunakan GraphQL untuk mendapatkan dan memanipulasi data.

e. Melakukan unit test dan dokumentasi terhadap setiap komponen yang dibuat.

f. Membuat template STARTER-TDK yang merupakan base template untuk me-representasikan penggunaan SIRCLO-TDK.

3.3 Uraian Pelaksanaan

3.3.1 Proses Pelaksanaan

Pelaksanaan kerja magang selama tiga bulan diuraikan dalam *timeline* kerja sebagai berikut:

Tabel 3.1 *Timeline* Kerja Magang

Kegiatan	Minggu ke-												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Onboarding, setup environment													
Pengerjaan Pre-Prius													
Pengerjaan SIRCLO-TDK													
Pengerjaan STARTER-TDK													

Pada minggu pertama, pembimbing lapangan memberikan panduan *setup environment* dan tugas untuk mengerjakan Pre-Prius. Prius adalah *user admin panel store* dengan tampilan baru yang sedang dibuat. Pre-Prius dibuat sebagai tampilan baru sementara sebagai transisi dari tampilan lama sebelum berubah menuju Prius. Pre-Prius dibuat dengan memodifikasi CSS tampilan lama *user admin panel store.* Di akhir minggu pertama, diadakan plotting untuk pengambilan tugas pembuatan SIRCLO-TDK. Metode manajemen projek yang digunakan adalah Agile dengan Sprint yang berlangsung selama dua minggu.

Sprint pertama dimulai pada minggu kedua hingga minggu ketiga, dilakukan dengan pembuatan komponen Social Media, Warning Message, dan Selected Payment serta pengerjaan Pre-Prius. Dilakukan pula unit testing untuk setiap komponen SIRCLO-TDK yang dibuat. Setelah sprint pertama selesai, dilakukan sebuah sprint review. Dari sprint review tersebut, diputuskan untuk dilakukan *research* terhadap flow penggunaan SIRCLO-TDK, bagaimana SIRCLO-TDK akan diimplementasi, dan bagaimana template writer dapat memanfaatkan SIRCLO-TDK dengan baik.

Setelah *research* dilakukan, diputuskan untuk dibuat sebuah template STARTER-TDK sebagai template awal implementasi SIRCLO-TDK. Dengan adanya STARTER-TDK, diharapkan template writer dapat lebih mengerti bagaimana cara menggunakan SIRCLO-TDK dalam membangun *template* generasi baru. Sambil mengerjakan SIRCLO-TDK dan STARTER-TDK, Pre-Prius sudah mulai di *deploy* serta ditemukan beberapa *bug* secara tampilan untuk diperbaiki.

Komponen SIRCLO-TDK yang selanjutnya dibuat adalah Shopping Cart, Order Summary, dan product list category. Pada tahap ini juga telah dicoba

penggunaan komponen SIRCLO-TDK yang telah dibuat sebelumnya pada *template* STARTER-TDK. Agar komponen Shopping Cart dan Order Summary dapat berfungsi dengan baik, dibuat *template function* yang berperan untuk *state management* pada level *template*.

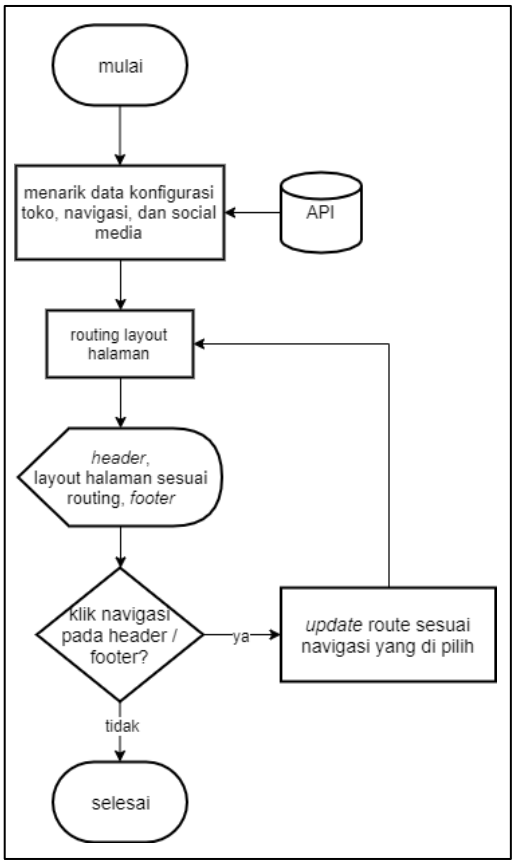
Tiga minggu terakhir digunakan untuk menyelesaikan template STARTER-TDK fase pertama, diantaranya menyelesaikan alur pemesanan barang sehingga fungsi memasukkan barang ke dalam Cart, input data pengiriman, metode pengiriman, dan metode pembayaran dapat dilakukan. Halaman yang terkait dengan fungsi tersebut diantaranya adalah Home, List Products, Product Details, Cart, Shipping Details, Shipping Method, dan Payment Method.

3.3.2 Tools dan Library yang Digunakan

Pada minggu pertama pelaksanaan magang, dilakukan setup environment untuk menyesuaikan dengan alur kerja yang berlaku pada lokasi magang. Untuk pembagian dan koordinasi tugas, *platform* yang digunakan adalah Phabricator dan Arcanist. Phabricator adalah sebuah *platform* pengembangan perangkat lunak sumber terbuka yang dapat digunakan untuk manajemen projek (MediaWiki, 2019). Arcanist merupakan *command line tool* yang digunakan untuk *submit, review* dan *land (commit)* pembaruan terhadap repositori *git* yang ada (MediaWiki, 2019). Setiap ada perubahan atau penambahan yang dilakukan terhadap suatu repositori harus di *review* terlebih dahulu oleh sesama tim pengembang terkait. Apabila dalam proses review ditemukan kekurangan atau membutuhkan revisi, maka harus diselesaikan terlebih dahulu sebelum akhirnya di *approve* agar dapat dilakukan *land* (*commit*) terhadap repositori projek.

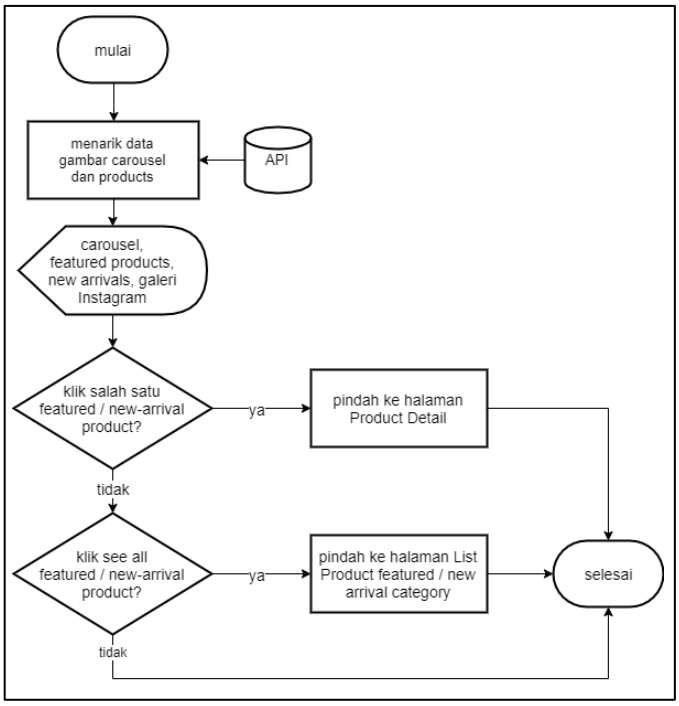
Pembuatan *library* komponen SIRCLO-TDK dan template STARTER-TDK menggunakan *library* ReactJS dan Redux. Pengambilan data dari API *backend* dilakukan melalui GraphQL. ReactJS merupakan library javascript berbasis komponen untuk membangun *user interface*. Redux merupakan library untuk mengelola *state* suatu aplikasi. GraphQL (*query language*) merupakan suatu konsep untuk membangun API yang diimplementasikan pada sisi *server*. GraphQL tidak berhubungan langsung dengan *database* tertentu. Posisi GraphQL terletak pada sisi *client* dan *server* yang berhubungan mengakses suatu API (HowToGraphQL, 2018). Tujuannya adalah untuk mempermudah komunikasi data antara backend dan front-end suatu aplikasi. Dengan GraphQL, dapat di desain suatu API yang dapat mengembalikan data sesuai yang dibutuhkan saja melalui sistem skema. Bahasa yang digunakan untuk pengembangan SIRCLO-TDK dan STARTER-TDK adalah Typescript. Typescript adalah bahasa pemrograman berbasis Javascript yang menambahkan fitur *strong-typing*.

3.3.3 Flowchart



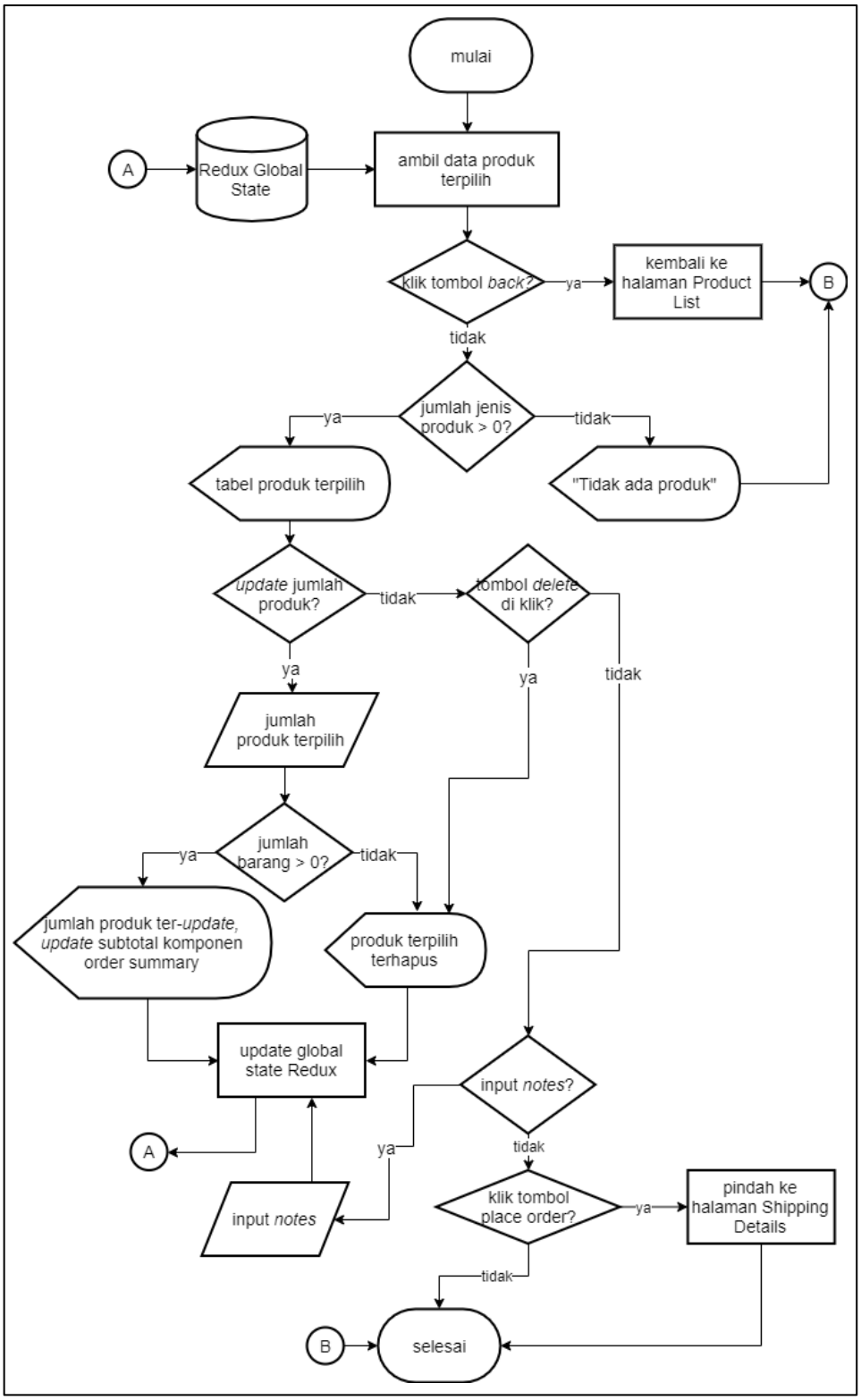
Gambar 3.2 *Flowchart* halaman index

Laporan ini fokus membahas pada pengembangan halaman index serta *layout* halaman Home, Cart, dan Shipping Details pada template STARTER-TDK. Halaman index berperan sebagai pembungkus *layout* halaman berbasis *routing*. Setiap halaman memiliki komponen *header* yang berisi navigasi dan logo toko serta komponen *footer* yang berisi komponen Social Media dan navigasi lainnya. Komponen template *header* dan *footer* tersebut di *render* melalui halaman index. Sehingga saat berpindah halaman, yang akan di update adalah *layout* halaman berdasarkan *routing*. Komponen SIRCLO-TDK yang dibuat oleh pelaksana magang dan dipakai diantaranya adalah Social Media, Shopping Cart dan Order Summary.



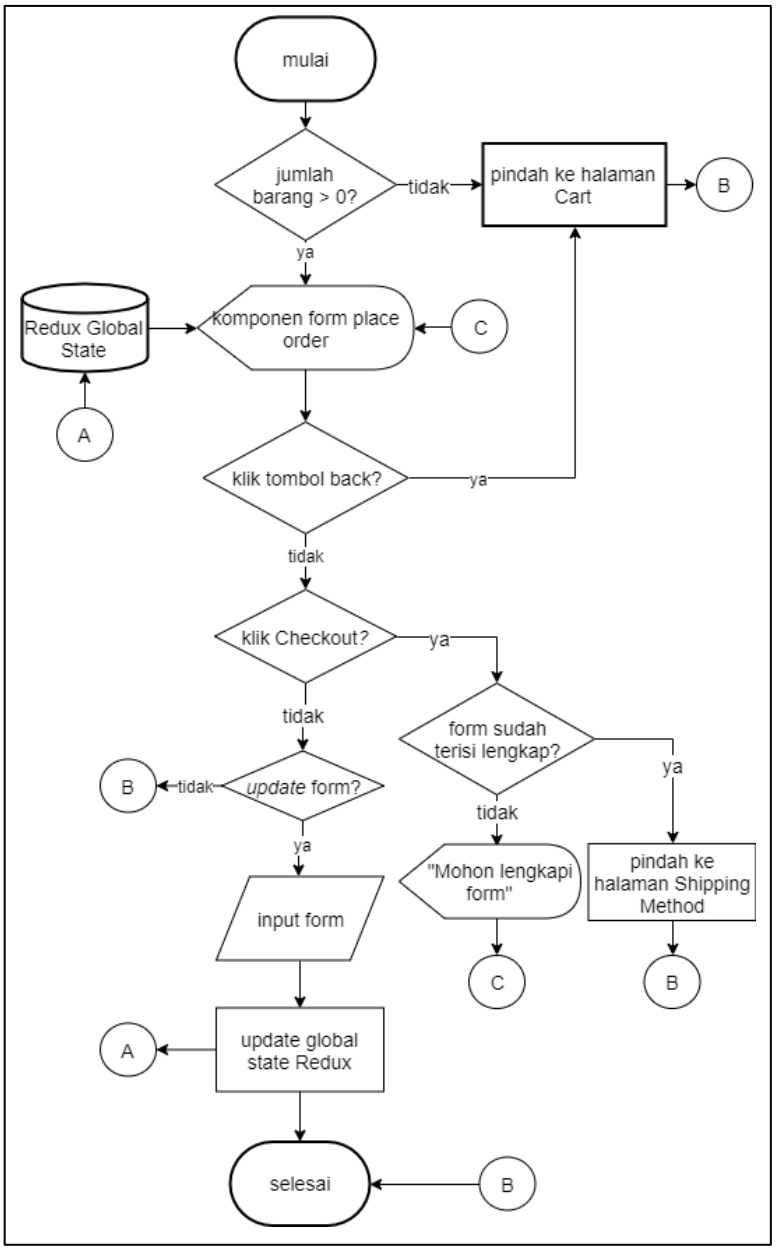
Gambar 3.3 *Flowchart layout* halaman Home

Halaman Home merupakan halaman pertama yang ditampilkan ketika membuka halaman toko dengan *template* STARTER-TDK. Layout halaman Home berisi carousel, featured products, new products, dan galeri Instagram toko.



Gambar 3.4 *Flowchart layout* halaman Cart

Layout halaman Cart terdiri atas komponen Shopping Cart dan Order Summary. Saat menampilkan *layout* halaman Cart, jumlah barang terpilih di cek terlebih dahulu. Apabila belum ada barang yang dipilih ke dalam Cart atau semua produk dalam Cart terhapus, maka halaman Cart menampilkan teks "Tidak ada produk" atau "*You haven't choose any product*". Jika sudah ada produk yang dimasukkan ke dalam Cart, maka ditampilkan komponen Shopping Cart yang berisi tabel barang terpilih, komponen Order Summary yang berisi subtotal harga seluruh barang terpilih, dan komponen *notes for seller* yang berisi input form teks. Pada halaman ini user dapat melakukan *update* terhadap jumlah barang setiap produk. *Update* jumlah produk dapat dilakukan dengan input angka maupun melalui tombol tambah atau kurang. Untuk menghapus suatu produk dapat dilakukan dengan input angka 0 atau melalui tombol hapus. Setiap *update* atau hapus produk dilakukan, angka subtotal pada komponen Order Summary akan ter-*update*. Jika user ingin melanjutkan pemesanan, dapat dilakukan dengan klik tombol "*Place Order*" dan akan berpindah menuju halaman Shipping Details.

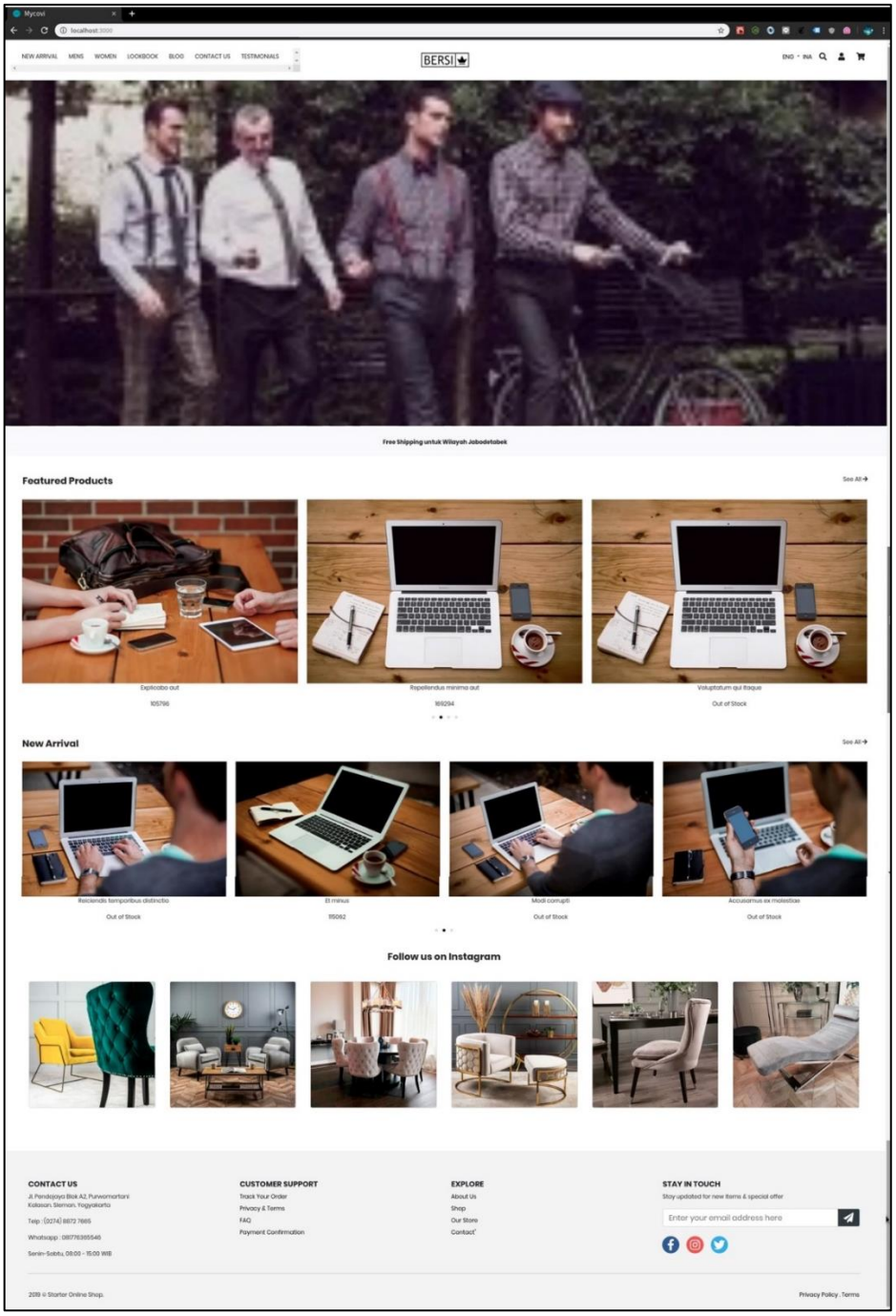


Gambar 3.5 *Flowchart layout* halaman Shipping Details

Halaman Shipping Details terdiri atas komponen *form place order* dan menampilkan Order Summary yang berisi angka subtotal. Jika belum ada produk yang masuk dalam Cart, maka akan langsung diarahkan ke halaman Cart. Pada halaman ini user dapat memasukkan data kontak dan alamat tujuan pengiriman. Setelah mengisi semua data yang diperlukan, user dapat melakukan pindah ke halaman Shipping Method dengan melakukan klik pada tombol "*Checkout*".

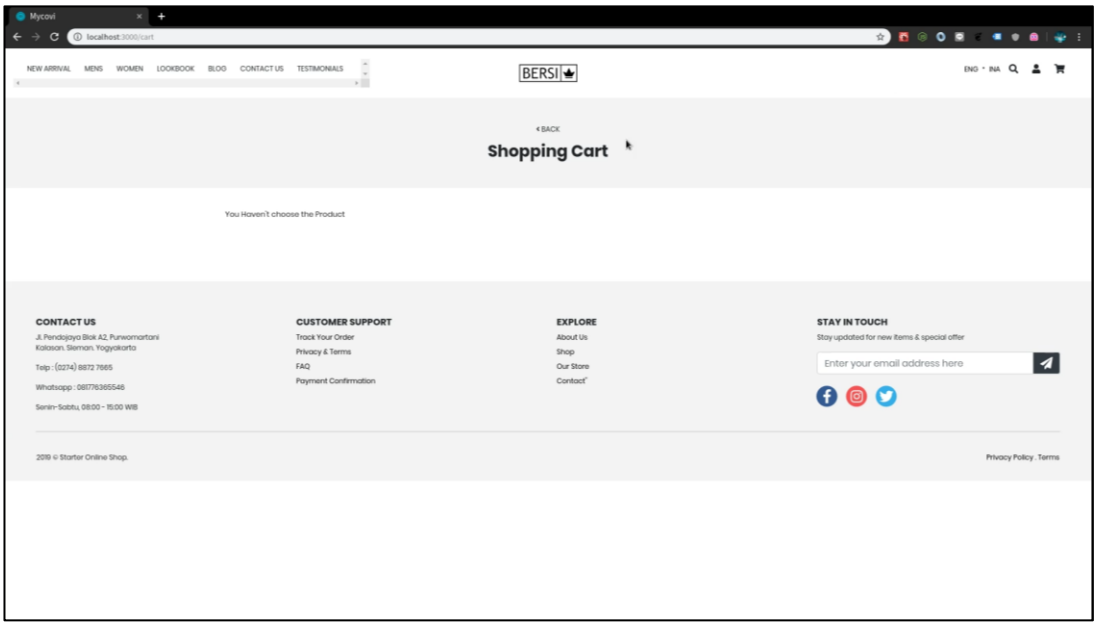
Namun, apabila data yang diperlukan belum lengkap atau ada field yang masih kosong, maka akan muncul teks “Mohon lengkapi form” atau “*Please complete the form*” dan tidak dapat pindah ke halaman Shipping Method.

3.3.4 Implementasi



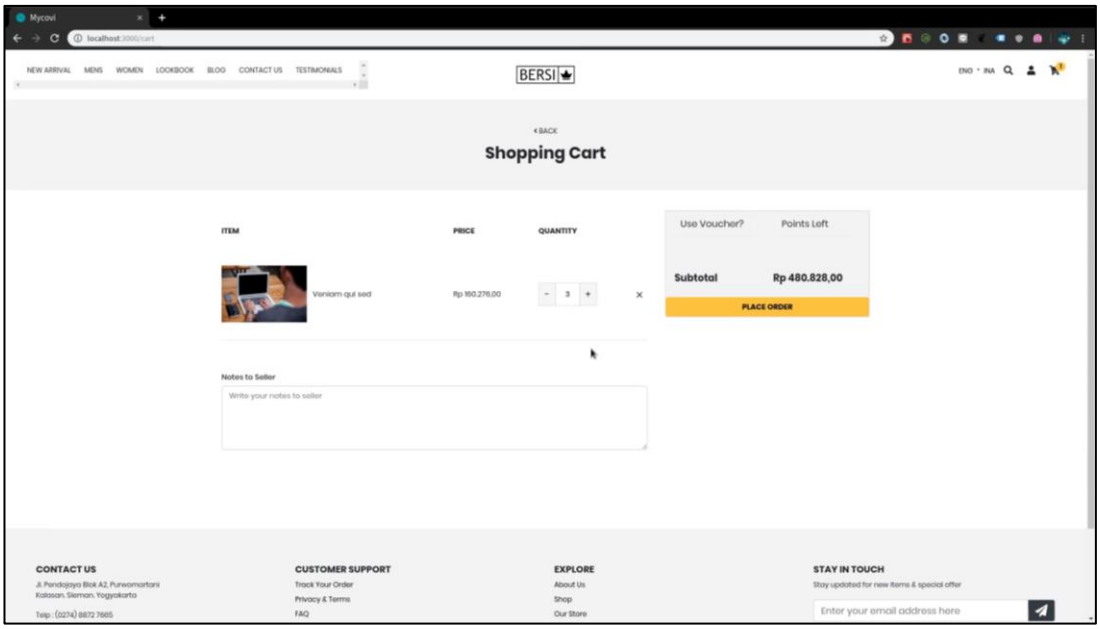
Gambar 3.6 Halaman Home

Halaman Home sebagai halaman pertama yang ditampilkan ketika halaman toko dikunjungi. Data yang digunakan masih merupakan data dummy yang didapatkan dari API *backend*. Pada bagian footer terdapat komponen Social Media dari library SIRCLO-TDK dengan data yang diambil dari konfigurasi toko.

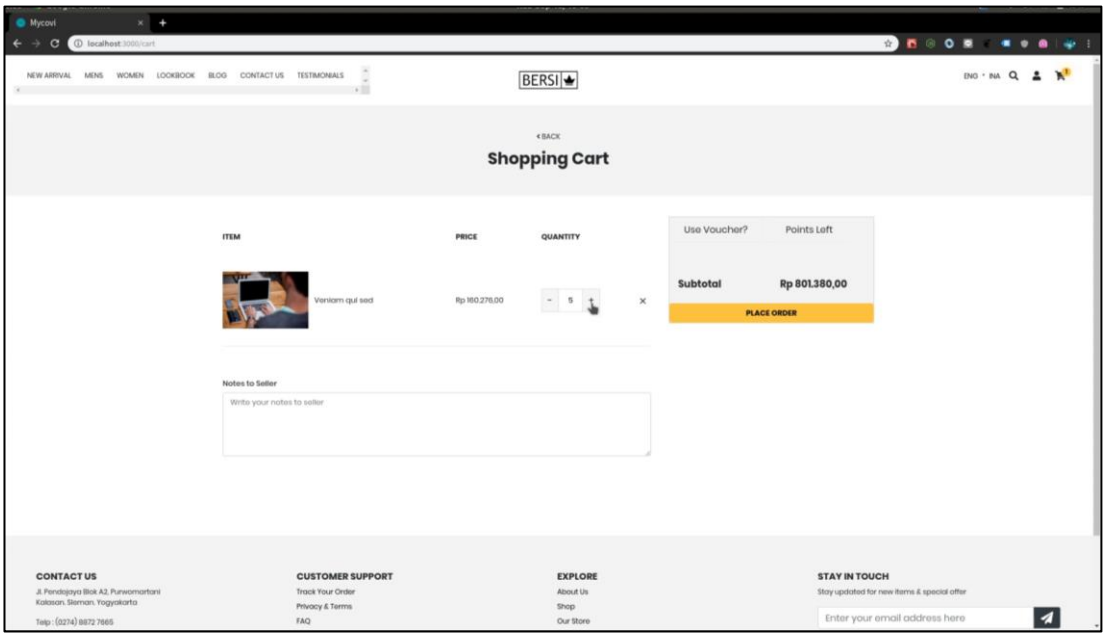


Gambar 3.7 Halaman Cart menampilkan teks *"You haven't choose the product"* saat belum ada barang yang dimasukkan ke dalam Shopping Cart.

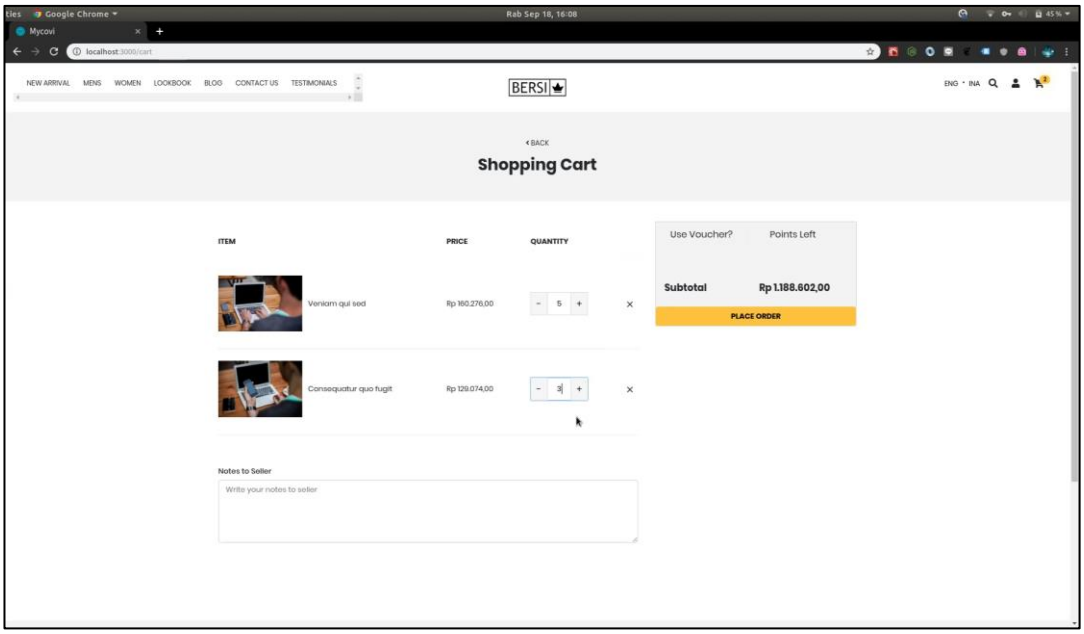
Halaman Cart dapat dikunjungi melalui klik tombol *icon* Cart pada header atau setelah memasukkan barang ke dalam Shopping Cart melalui halaman detail salah satu produk. *Layout* halaman Cart terdiri atas komponen Shopping Cart berupa tabel dan Order Summary berupa *box* yang menampilkan angka subtotal.



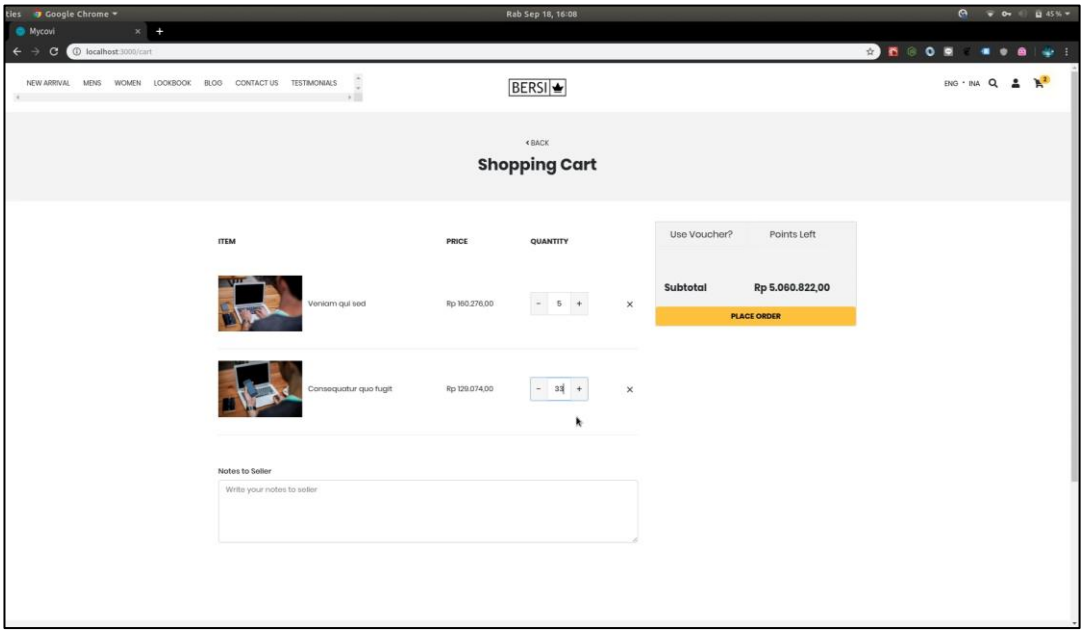
Gambar 3.8 Halaman Cart menampilkan barang yang sudah dimasukkan dalam Shopping Cart



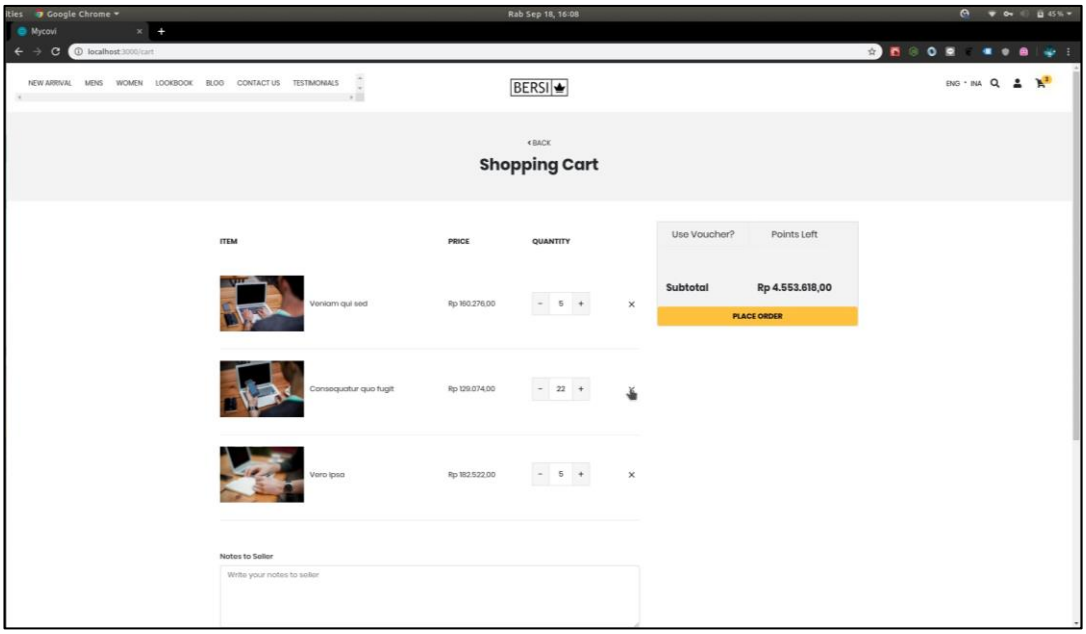
Gambar 3.9 Halaman Cart saat update jumlah barang suatu produk melalui tombol tambah



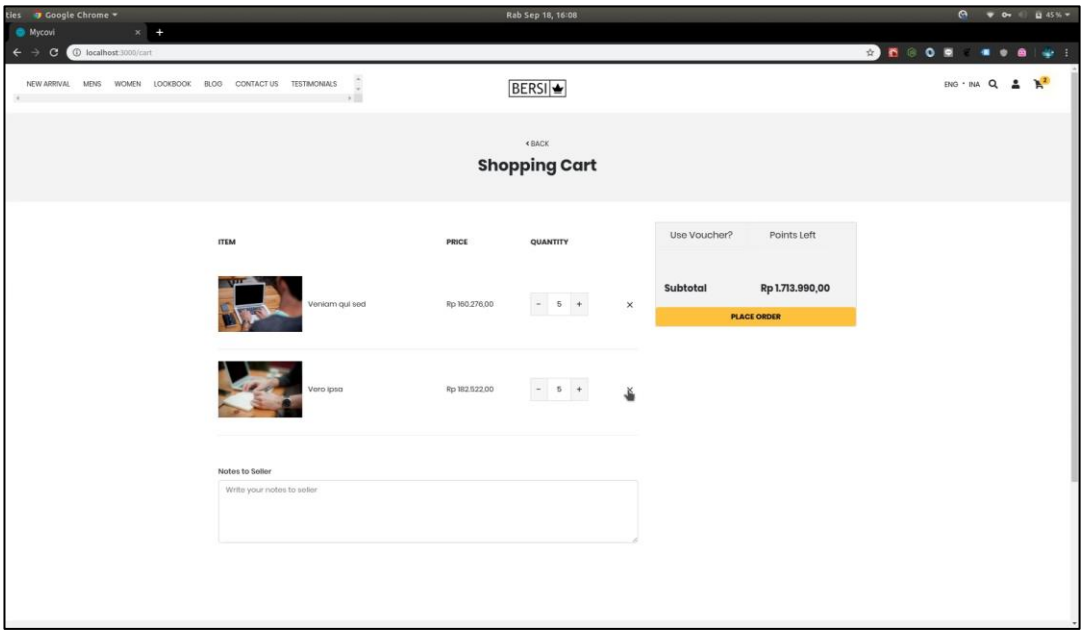
Gambar 3.10 Halaman Cart saat menambahkan barang lain



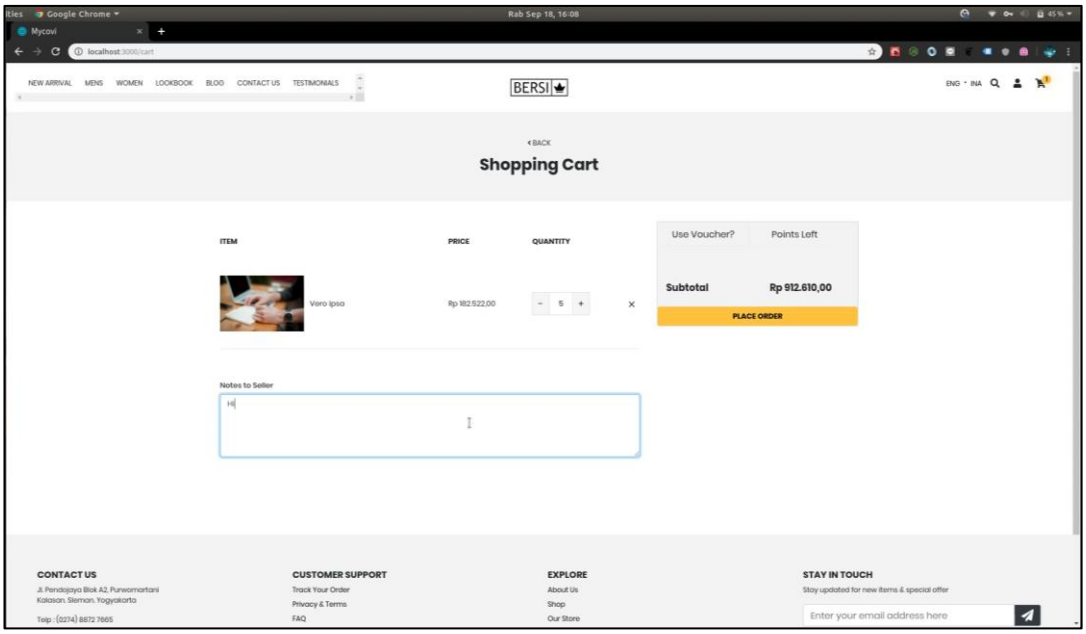
Gambar 3.11 Halaman Cart saat update jumlah suatu barang melalui input angka



Gambar 3.12 Halaman Cart saat menambahkan barang lainnya dan mau menghapus barang kedua

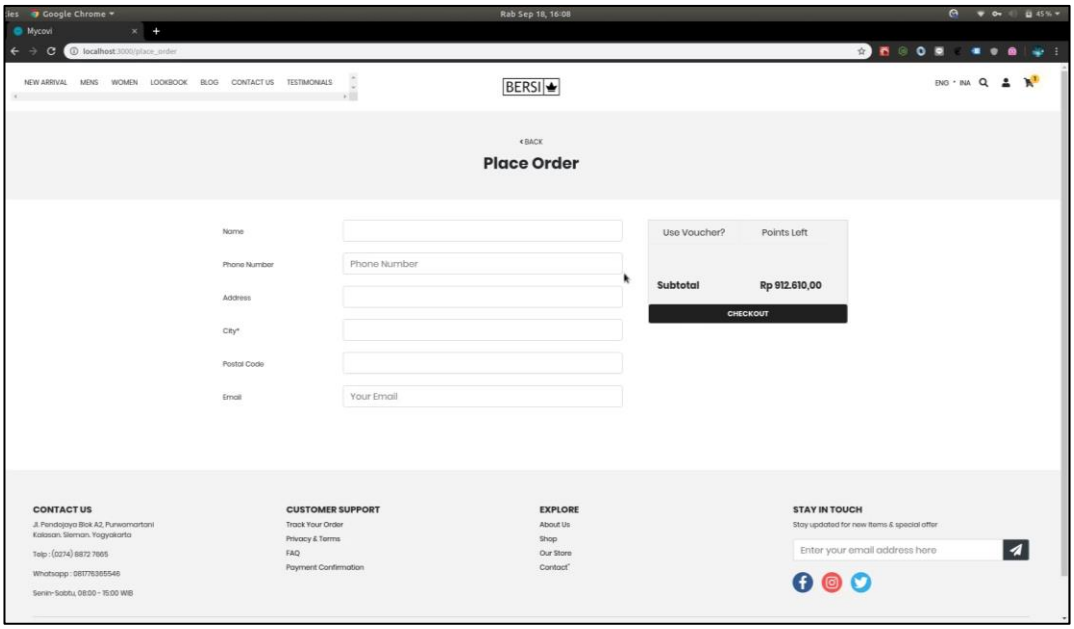


Gambar 3.13 Halaman Cart saat barang kedua telah dihapus melalui tombol hapus

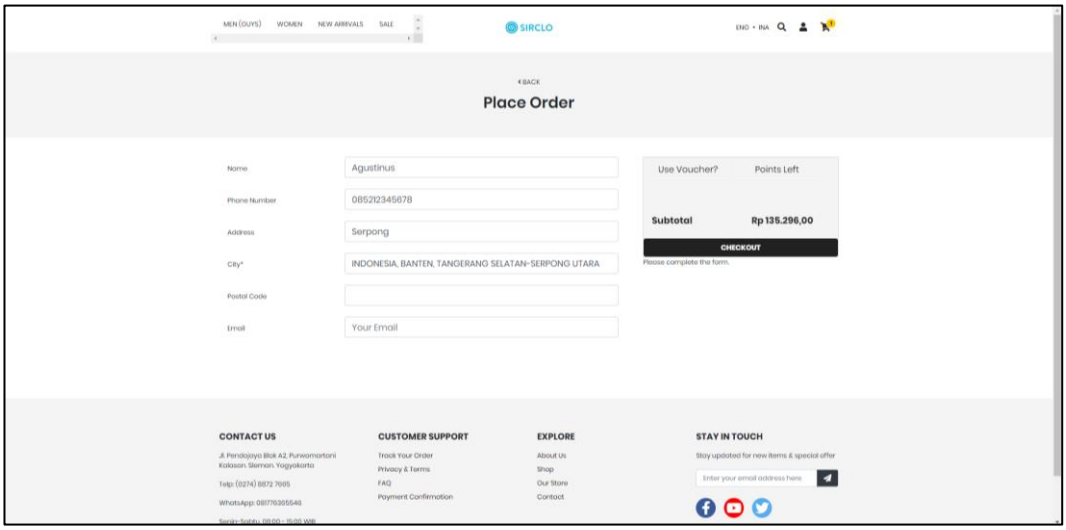


Gambar 3.14 Halaman Cart saat *update notes to seller* melalui form teks

Jika sudah ada barang yang dimasukkan ke dalam Cart dan user melakukan klik Place Order pada komponen Order Summary, maka akan terjadi perpindah halaman menuju halaman Shipping Details untuk mengisi data pengiriman. Halaman Shipping Details terdiri atas komponen form Place Order dan Order Summary.



Gambar 3.15 Halaman Shipping Details



Gambar 3.16 Halaman Shipping Details saat diisi dan belum lengkap, lalu klik tombol *checkout*

Layout halaman Shipping Details terdiri atas form Place Order dan Order Summary. Jika form Place Order pada halaman Shipping Details telah diisi dengan lengkap, maka user dapat melanjutkan ke halaman Shipping Method dengan melakukan klik Checkout pada komponen Order Summary.

3.3.5 Kendala yang Ditemukan

Kendala teknis yang dihadapi diantaranya adalah kurangnya pengetahuan pelaksana magang dalam mengembangkan aplikasi menggunakan ReactJS, Redux, unit testing, penggunaan Arcanist, serta ada beberapa tugas yang harus menunggu karena *blocking* dari backend. *Blocking* yang terjadi diantaranya API yang belum siap sehingga harus menunggu terlebih dahulu agar komponen terkait dapat dibuat. Hal ini terkadang membuat pelaksana magang menghabiskan waktu yang cukup lama dalam masa awal pengembangan.

Kendala non-teknis yang dihadapi diantaranya sering ditemukan requirement yang belum jelas. Sehingga seringkali suatu komponen yang sudah jadi dikerjakan kembali untuk disesuaikan kembali dengan requirement yang dibutuhkan.

3.3.6 Solusi atas Kendala yang Ditemukan

Solusi yang ditemukan diantaranya adalah dengan bertanya kepada rekan kerja dan pembimbing lapangan serta melakukan *browsing* konten yang berkaitan dengan kesulitan yang dihadapi. Selain itu, secara rutin mengikuti Daily Stand-Up mendukung untuk menyampaikan kesulitan yang dihadapi dan *progress* yang telah dilakukan sehingga sesama rekan kerja dapat memberikan masukan dan bantuan. Backlog Refinement dan Sprint Planning yang dilakukan seiring waktu membantu menyesuaikan kemampuan setiap individu yang terlibat terhadap tugas-tugas yang akan diambil serta memperjelas requirement setiap tugas.

Jika terdapat waktu tunggu dari *blocking* atau ketika pelaksana magang sedang merasa buntu terhadap tugas yang sedang dikerjakan, pelaksana magang menghampiri sesama rekan kerja untuk membantu tugas yang sedang dikerjakan

sesama rekan kerja. Dengan begitu, pelaksana magang dapat belajar hal baru dari tugas sesama rekan kerja serta membantu mempercepat selesainya tugas-tugas lain dalam satu tim.