BAB 3

PELAKSANAAN KERJA MAGANG

3. 1 Kedudukan dan Koordinasi

Dalam pelaksanaan kerja magang di PT. Indobest Artha Kreasi, penulis memegang jabatan sebagai *Software Engineer Intern* di Divisi IT Whitelabel. Pelaksanaan kerja magang disupervisi oleh Reza Satyawijaya selaku *Product Lead* Whitelabel. Reza Satyawijaya berperan dalam memberikan informasi, tugas, dan bimbingan terkait pengembangan produk Whitelabel PT. Indobest Artha Kreasi.

3. 2 Tugas yang Dilakukan

Selama pelaksanaan kerja magang, tugas dan tanggung jawab yang diberikan kepada penulis adalah sebagai berikut.

- Mengembangkan produk Whitelabel dengan melakukan penambahan fitur dan fungsionalitas.
- Mengembangkan produk Whitelabel dengan melakukan berbagai perbaikan pada sistem yang telah ada.
- Mendokumentasikan segala pengembangan sistem yang dilakukan.
- Berkolaborasi dengan sesama anggota tim Whitelabel dan seluruh anggota divisi IT untuk menyelesaikan masalah pengembangan produk Whitelabel.

3. 3 Uraian Pelaksanaan

3.3.1 Proses Pelaksanaan

Kerja magang dilaksanakan selama hampir 3 (tiga) bulan atau 11 (sebelas) minggu dengan *timeline* kerja sebagai berikut.

Tabel 3.1 Jadwal Kerja Magang

Nama Kegiatan	Minggu ke										
	1	2	3	4	5	6	7	8	9	10	11
Instalasi *tools* dan *setup environment*											
Pengenalan struktur proyek dan bisnis											
Pembaharuan *database*											
Melakukan perbaikan fitur											
Membuat fitur menu sidebar *custom*											
Membuat fitur *top up by excel*											
Membuat fitur perubahan otomatis *database tenant*											
Dokumentasi											

3.3.2 Framework yang Digunakan

Pada pengembangan produk Whitelabel di PT. Indobest Artha Kreasi, secara keseluruhan digunakan dua buah *framework*, yaitu Laravel dan Vue.js. *Framework* Laravel digunakan untuk melakukan pengembangan bagian *back-end*, sedangkan *framework* Vue.js digunakan untuk melakukan pengembangan bagian *front-end* pada Whitelabel.

Laravel adalah *framework* aplikasi berbasis web ternama dan telah digunakan oleh banyak pengembang aplikasi berbasis web. *Framework* Laravel merupakan *framework* berbasis PHP dan menggunakan arsitektur MVC (*Model – View –* Controller). Laravel merupakan *framework* yang terkenal akan sintaks yang elegan dan ekspresif. *Framework* ini juga merupakan *framework* yang memiliki kegunaan yang sangat luas dan relatif mudah untuk digunakan, karena menawarkan berbagai keringkasan dalam penanganan berbagai fungsionalitas aplikasi berbasis web seperti otentikasi, *routing*, *sessions*, dan *caching*. Dengan menggunakan *framework* Laravel, pengembang dapat membuat aplikasi berbasis web yang kompleks dengan mudah dan juga menghemat waktu (Taylor Otwell, 2011, laravel.com, 19 September 2020).

Vue.js adalah *framework* berbasis Javascript dan merupakan *framework* yang progresif. Desain *framework* Vue.js berfokus pada bagian *view* pada aplikasi berbasis web maupun seluler, sehingga dapat dengan mudah diintegrasikan dengan *framework* maupun *libraries* lainnya dalam sebuah proyek. Dengan menggunakan *framework* Vue.js, pengembang juga dapat mengembangkan *Single-Page Application* (SPA) yang canggih dan kompleks dengan mudah dan ringkas (Evan You, 2014, vuejs.org, 19 September 2020).

Selain menggunakan kedua *framework* tersebut, dalam pengembangan produk Whitelabel digunakan juga berbagai *libraries* antarmuka yang mendukung, seperti Vuetify, Element, tailwindcss, Lodash, dan berbagai *libraries* lainnya.

Penyimpanan *database* pada Whitelabel menggunakan MySQL dan MongoDB. MySQL digunakan secara khusus untuk menyimpan data – data utama sistem Whitelabel maupun data – data setiap situs web *tenant*, sedangkan MongoDB digunakan secara khusus menyimpan data – data yang berkaitan dengan transaksi yang dilakukan melalui Whitelabel.

3.3.3 Perancangan Sistem

Perancangan sistem dilakukan dengan pembuatan *flowchart* dan perancangan *user interface*. *Flowchart* digunakan sebagai alat bantu dalam menjelaskan alur suatu program dengan lebih mudah, dan sebagai dokumentasi pembuatan program. Terdapat 16 *flowchart* dan 8 rancangan *user interface* yang dibuat berdasarkan tiga buah fitur yang dikerjakan selama kerja magang.

Perancangan sistem *database* berupa *Entity Relationship Diagram* (ERD) ataupun deskripsi tabel – tabel *database* tidak dibuat pada laporan ini. Hal ini dikarenakan kebijakan PT. Indobest Artha Kreasi yang tidak memperbolehkan pencantuman salah satu aset perusahaan yang berupa sistem *database* ke dalam laporan.

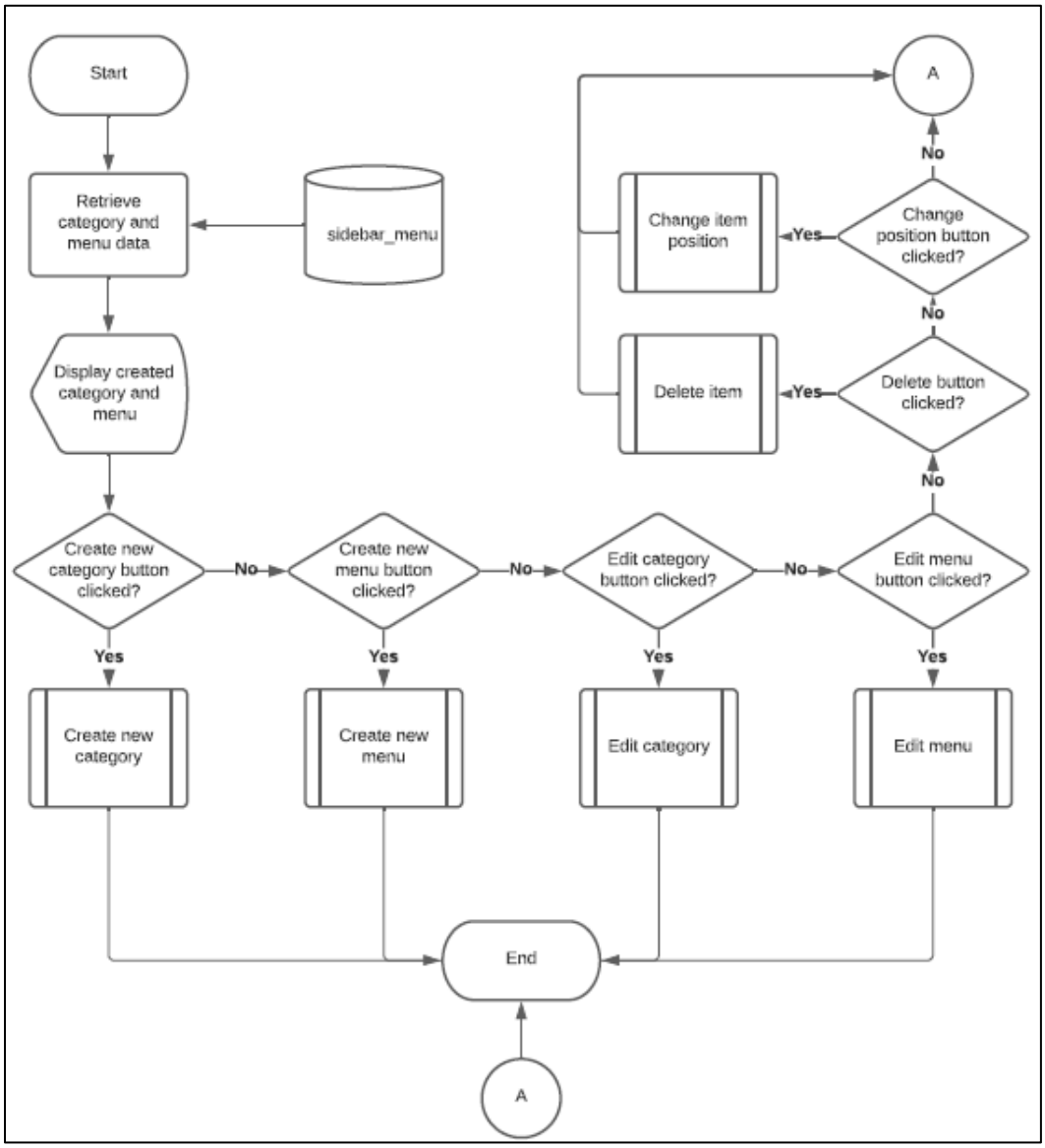
Rancangan *user interface* untuk fitur perubahan otomatis *database tenant* tidak dibuat pada laporan ini. Hal ini dikarenakan *user interface* fitur perubahan otomatis *database tenant* telah selesai dikerjakan oleh orang lain. Pengembangan fitur perubahan otomatis *database tenant* hanya berfokus pada bagian *back-end*.

A. Flowchart

A.1 Flowchart Fitur Menu Sidebar Custom

Fitur menu *sidebar custom* berfungsi untuk memungkinkan *tenant* Whitelabel melakukan penambahan menu *sidebar* baru yang dirancang sesuai dengan keinginan *tenant* sendiri pada situs Whitelabel miliknya. *Tenant* Whitelabel dapat menambahkan kategori *custom* khusus, dan juga menambahkan menu *custom* khusus pada kategori tersebut.

Gambar 3.1 menggambarkan alur kerja halaman utama dari fitur menu *sidebar custom*. Pada menu ini, pengguna dapat melihat seluruh data kategori dan menu yang telah ada. Selain itu, pengguna juga dapat melakukan beberapa hal lain, yaitu membuat kategori dan menu baru, memperbaharui data kategori dan menu, memperbaharui posisi tampil kategori dan menu, serta menghapus data kategori dan menu tersebut.

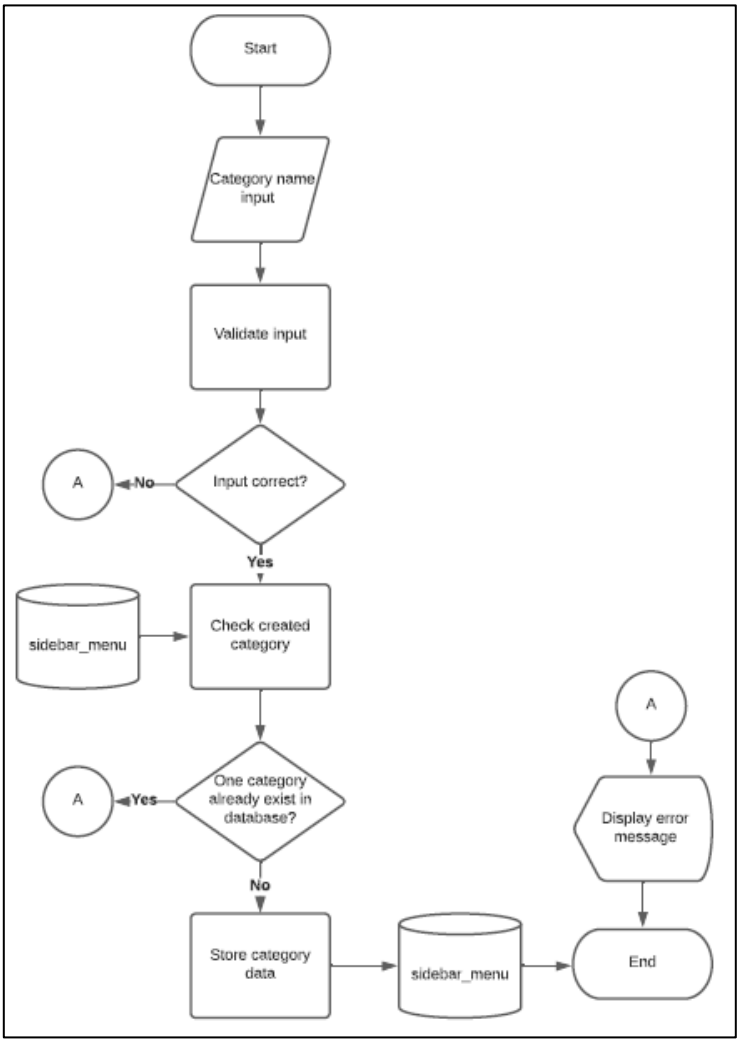


Gambar 3.1 Flowchart Halaman Utama Fitur Menu *Sidebar Custom*

Pada halaman ini, jika pengguna menekan suatu tombol, pengguna akan diarahkan ke halaman lain yang memiliki fungsi sesuai dengan masing – masing deskripsi tombol.

A.1.1 Flowchart Create New Category

Gambar 3.2 menggambarkan alur kerja halaman *create new category*.

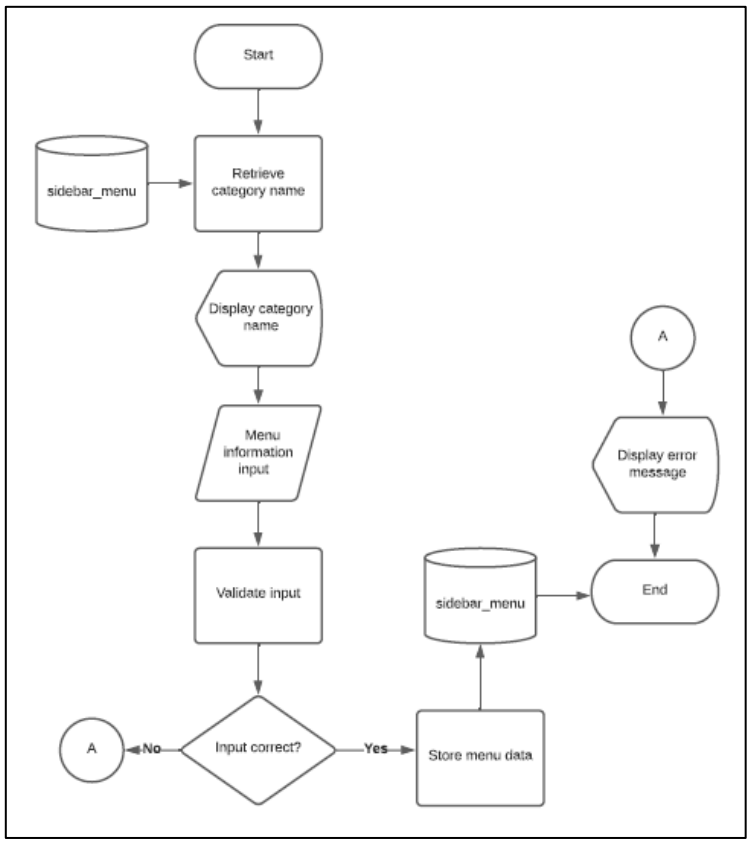


Gambar 3.2 Flowchart Halaman *Create New Category*

Pengguna akan diarahkan ke halaman ini jika menekan tombol *create new category* yang terdapat pada halaman utama fitur menu *sidebar custom*. Halaman ini memiliki fungsi agar pengguna dapat menambahkan kategori menu baru sesuai dengan keinginan pengguna. Setelah pengguna memasukkan nama kategori tersebut dan melakukan *submit*, program akan melakukan validasi dan mengecek apakah pengguna tersebut telah memiliki sebuah kategori di *database*. Jika pengguna telah memiliki sebuah kategori, halaman akan menampilkan pesan *error*. Hal ini dikarenakan, saat ini jumlah kategori yang diperbolehkan hanya satu agar tidak merusak tampilan yang telah ada. Jika pengguna belum memiliki sebuah kategori maka data kategori baru akan disimpan, dan halaman akan menampilkan pesan sukses.

A.1.2 Flowchart Create New Menu

Gambar 3.3 menggambarkan alur kerja halaman *create new menu*.

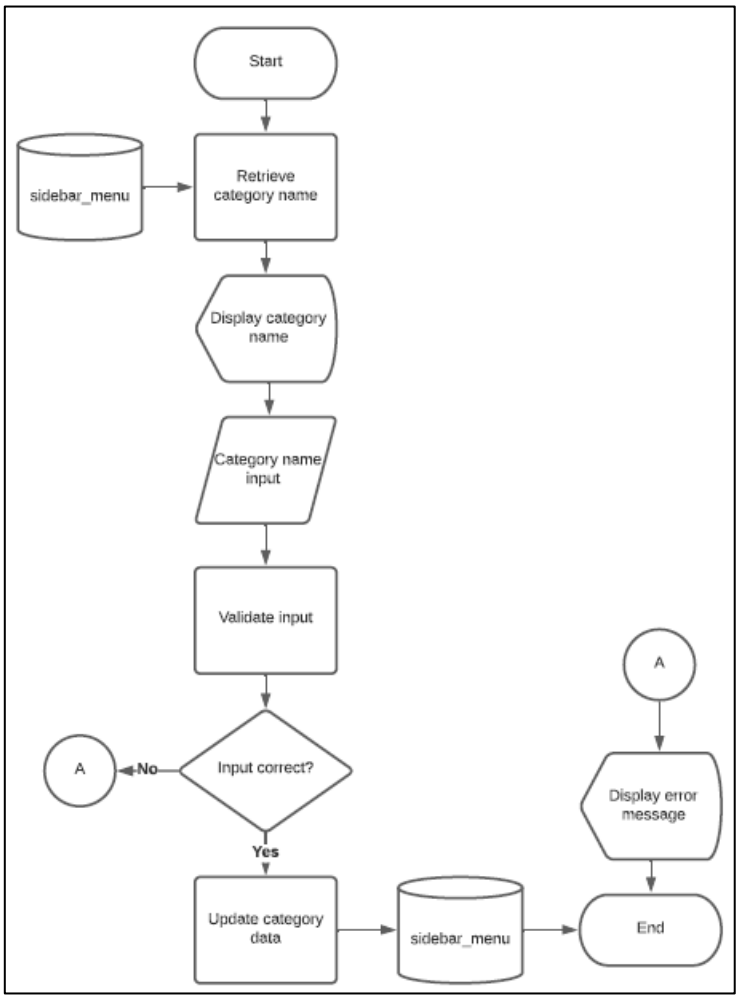


Gambar 3.3 Flowchart Halaman *Create New Menu*

Pengguna akan diarahkan ke halaman ini jika menekan tombol *create new menu* yang terdapat pada halaman utama fitur menu *sidebar custom*. Halaman ini memiliki fungsi agar pengguna dapat menambahkan menu baru pada sebuah kategori sesuai dengan keinginan pengguna. Setelah pengguna memasukkan data – data terkait menu yang ingin dibuat, program akan melakukan validasi dan menyimpan data – data tersebut ke *database*. Kemudian halaman akan menampilkan pesan sukses atau tidaknya proses tersebut.

A.1.3 Flowchart Edit Category

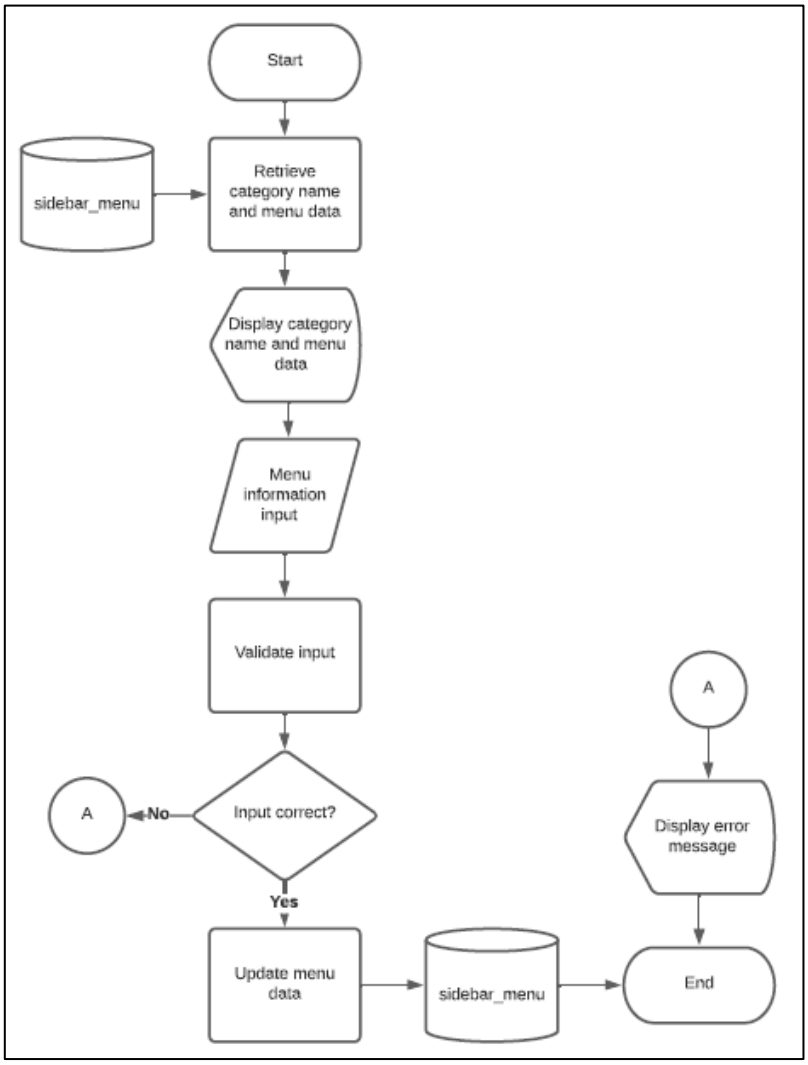
Gambar 3.4 menggambarkan alur kerja halaman *edit category*. Pengguna akan diarahkan ke halaman ini jika menekan tombol *edit category* yang terdapat pada halaman utama fitur menu *sidebar custom*. Halaman ini memiliki fungsi agar pengguna dapat mengubah nama kategori. Setelah pengguna memasukkan data nama kategori, program akan melakukan validasi dan menyimpan data tersebut ke *database*. Kemudian halaman akan menampilkan pesan sukses atau tidaknya proses tersebut.



Gambar 3.4 Flowchart Halaman *Edit Category*

A.1.4 Flowchart Edit Menu

Gambar 3.5 menggambarkan alur kerja halaman *edit menu.*



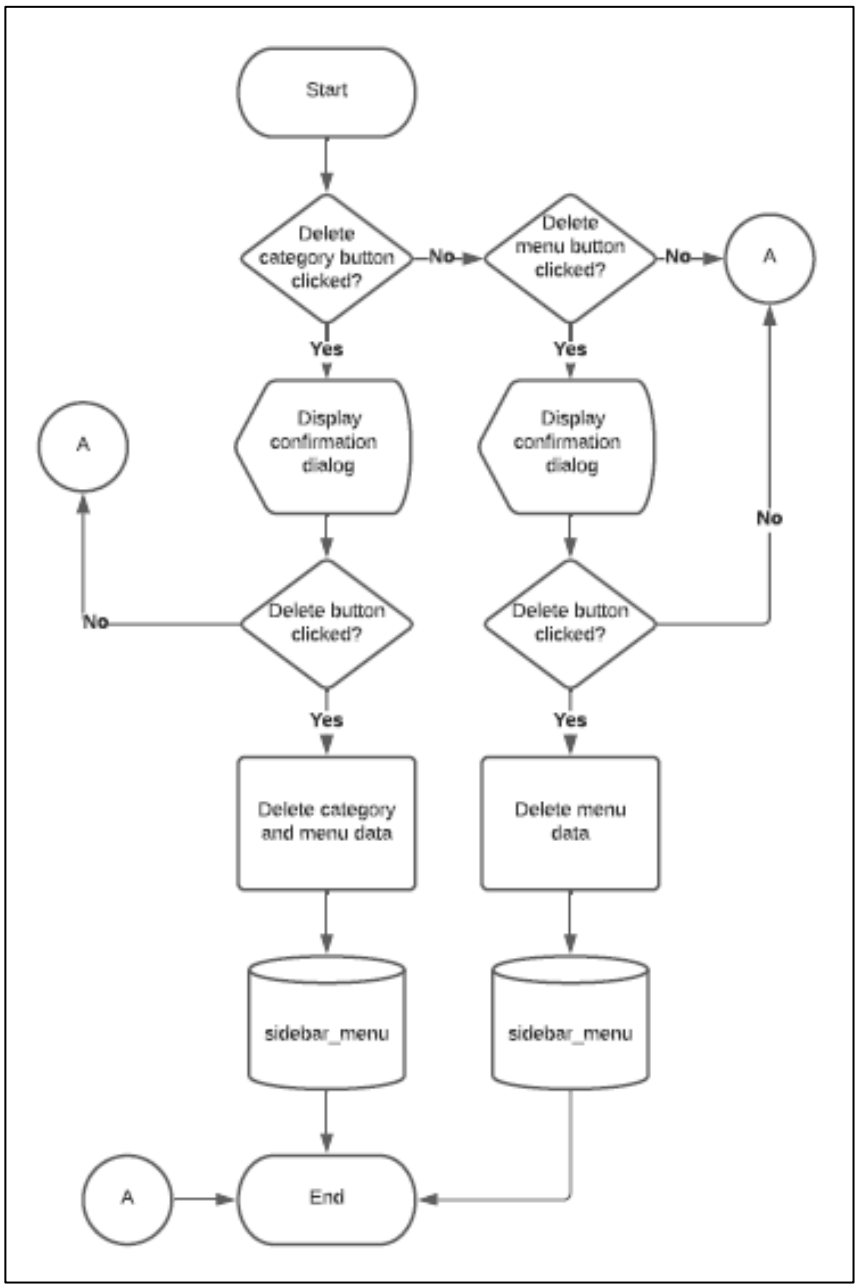
Gambar 3.5 Flowchart Halaman *Edit Menu*

Pengguna akan diarahkan ke halaman ini jika menekan tombol *edit menu* yang terdapat pada halaman utama fitur menu *sidebar custom*. Halaman ini memiliki fungsi agar pengguna dapat mengubah data – data menu setiap kategori. Setelah pengguna

memasukkan data - data menu, program akan melakukan validasi dan menyimpan data - data tersebut ke *database*. Kemudian halaman akan menampilkan pesan sukses atau tidaknya proses tersebut.

A.1.5 Flowchart Delete Item

Gambar 3.6 menggambarkan alur kerja program jika pengguna ingin menghapus kategori atau menu yang telah ada.



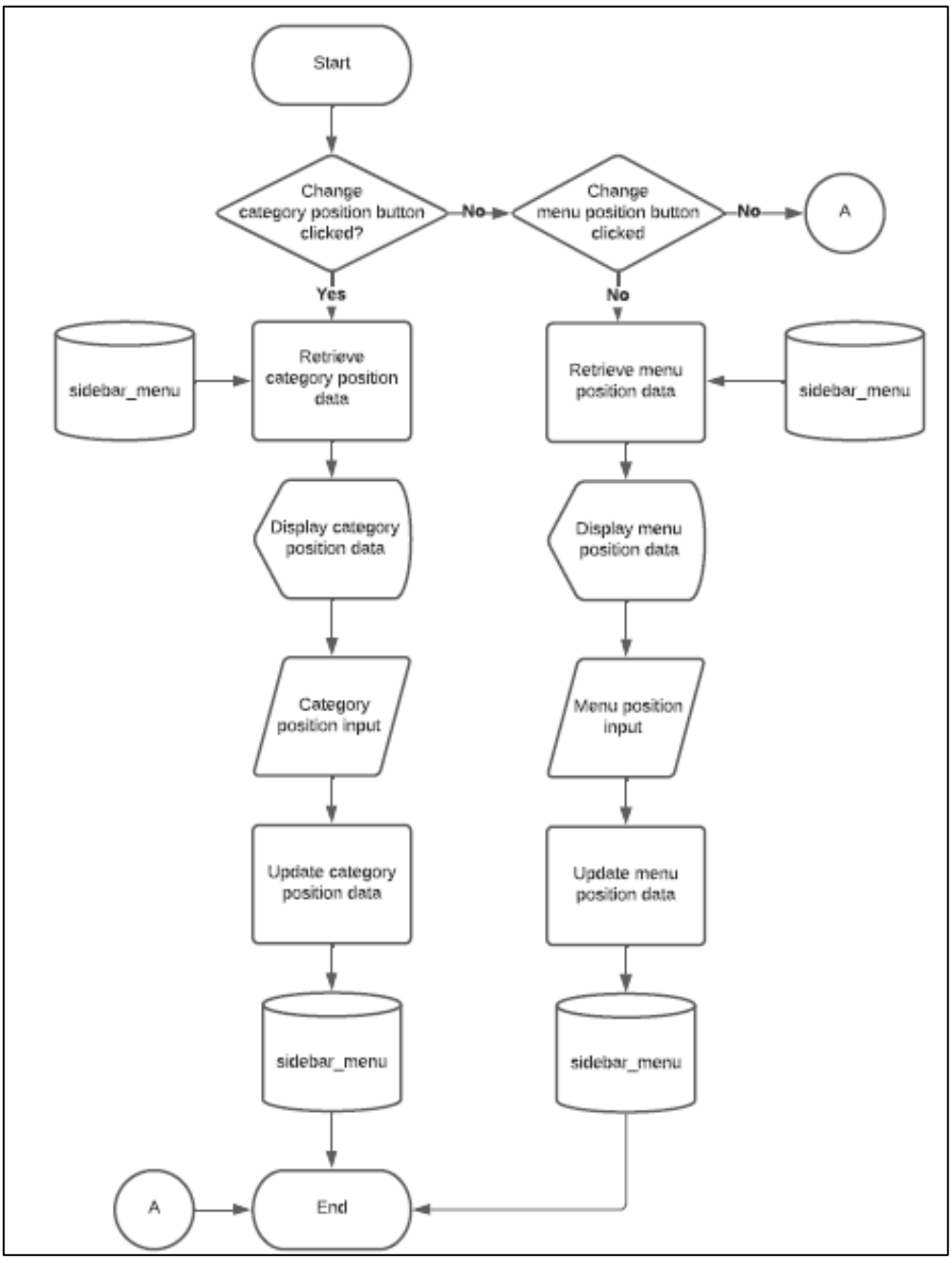
Gambar 3.6 Flowchart Delete Item

Jika pengguna menekan tombol hapus yang terdapat pada halaman utama fitur menu *sidebar custom*, sebuah *dialog* konfirmasi akan muncul. *Dialog* ini memiliki fungsi agar pengguna dapat melakukan konfirmasi penghapusan kategori atau menu

yang dipilih. Jika pengguna melakukan konfirmasi, maka program akan melakukan penghapusan kategori atau menu yang dipilih tersebut dari *database*. Jika pengguna menghapus kategori, maka menu – menu pada kategori tersebut juga akan dihapus. Halaman juga akan menampilkan pesan sukses atau tidaknya proses tersebut.

A.1.6 Flowchart Change Position

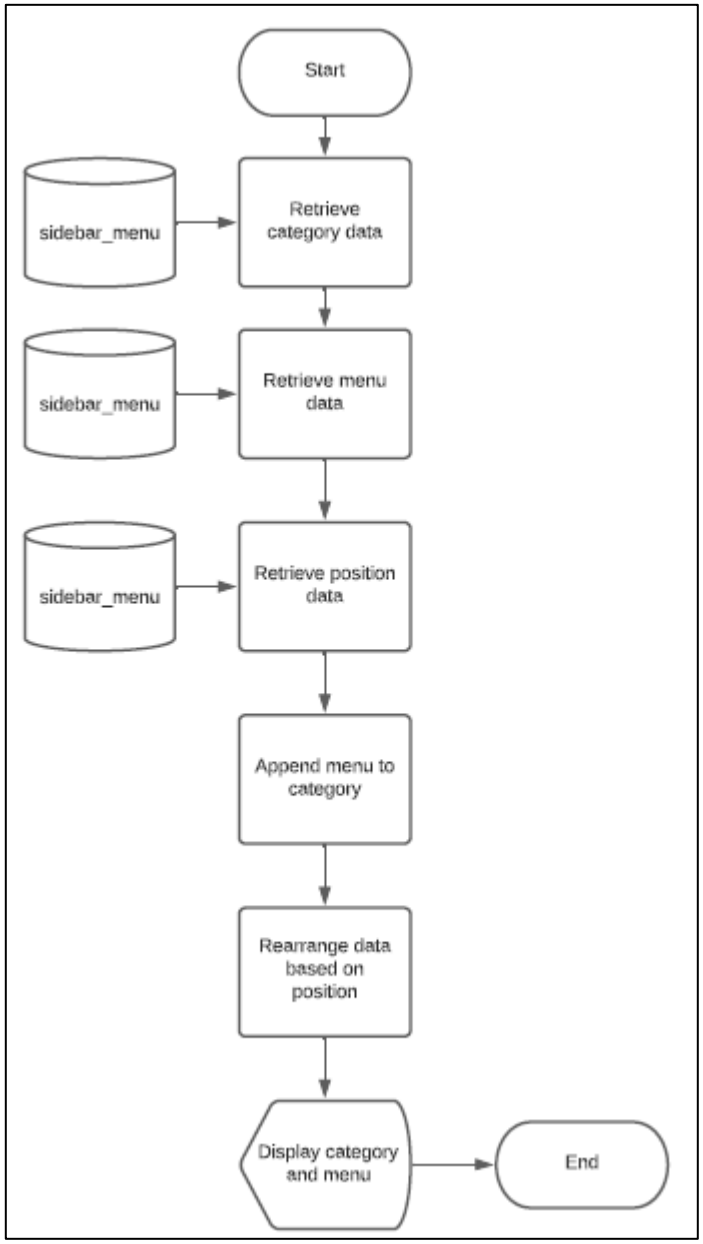
Gambar 3.7 menggambarkan alur kerja program jika pengguna ingin melakukan perubahan posisi tampil kategori atau menu yang telah dibuat. Pengguna akan diarahkan kepada halaman ini jika menekan tombol *change position* pada halaman utama fitur menu *sidebar custom*. Halaman ini memiliki fungsi agar pengguna dapat dengan bebas menentukan urutan posisi tampil kategori atau menu yang telah dibuat. Jika pengguna melakukan perubahan posisi, data urutan posisi tersebut akan disimpan pada *database,* kemudian halaman akan menampilkan pesan sukses atau tidaknya proses tersebut.



Gambar 3.7 Flowchart Change Position

A.1.7 Flowchart Penampilan Menu Sidebar Custom

Gambar 3.8 menggambarkan alur kerja program pada saat menampilkan kategori dan menu *custom* yang telah dibuat.



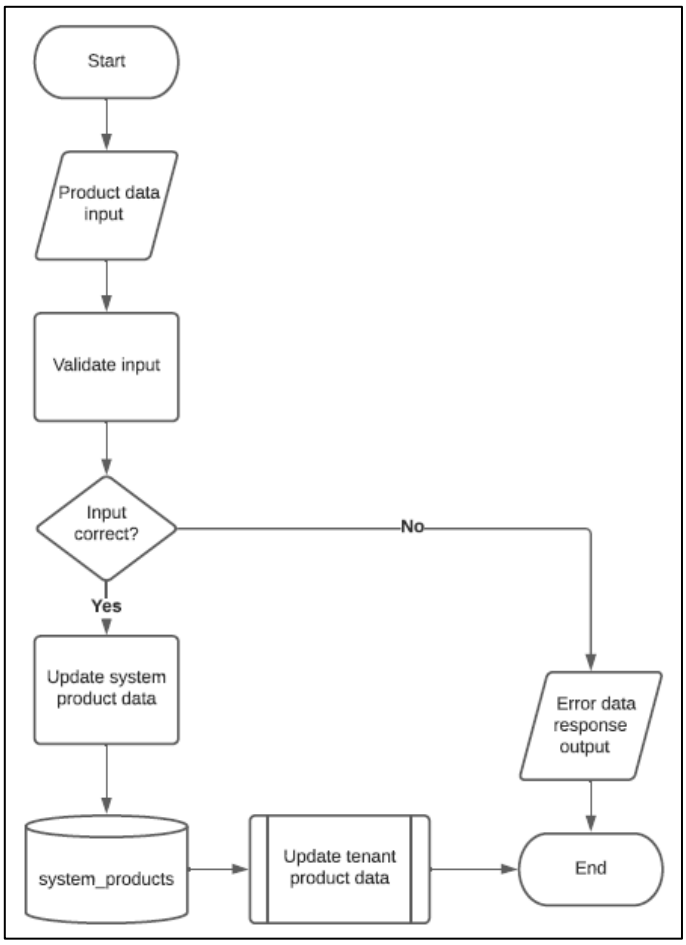
Gambar 3.8 Flowchart Penampilan Menu Sidebar Custom

Untuk menampilkan menu sidebar *custom* yang telah dibuat oleh pengguna, pertama – tama program akan melakukan pengambilan seluruh data kategori dan menu yang telah disimpan di *database*. Kemudian, data menu akan digabungkan ke data kategori yang sesuai. Terakhir, data akan diolah agar tampil sesuai dengan urutan yang telah ditentukan.

A.2 Flowchart Fitur Perubahan Otomatis Database

Fitur perubahan otomatis *database* adalah fitur yang berfungsi untuk mempermudah dan mempercepat proses perubahan data produk kepada seluruh *database tenant* Whitelabel yang aktif. Perubahan ini dapat terjadi jika *admin* Whitelabel PT. Indobest Artha Kreasi ingin melakukan pembaharuan data produk melalui portal *Superadmin* Whitelabel. Dengan adanya fitur ini, *admin* Whitelabel tidak perlu melakukan pembaharuan manual kepada setiap *database tenant* Whitelabel yang sedang aktif. Pengerjaan fitur ini hanya difokuskan pada bagian *back-end* dari situs *Superadmin* Whitelabel.

Gambar 3.9 menggambarkan alur kerja program secara keseluruhan pada saat sedang melakukan perubahan data otomatis kepada *database* sistem dan *database* tenant.



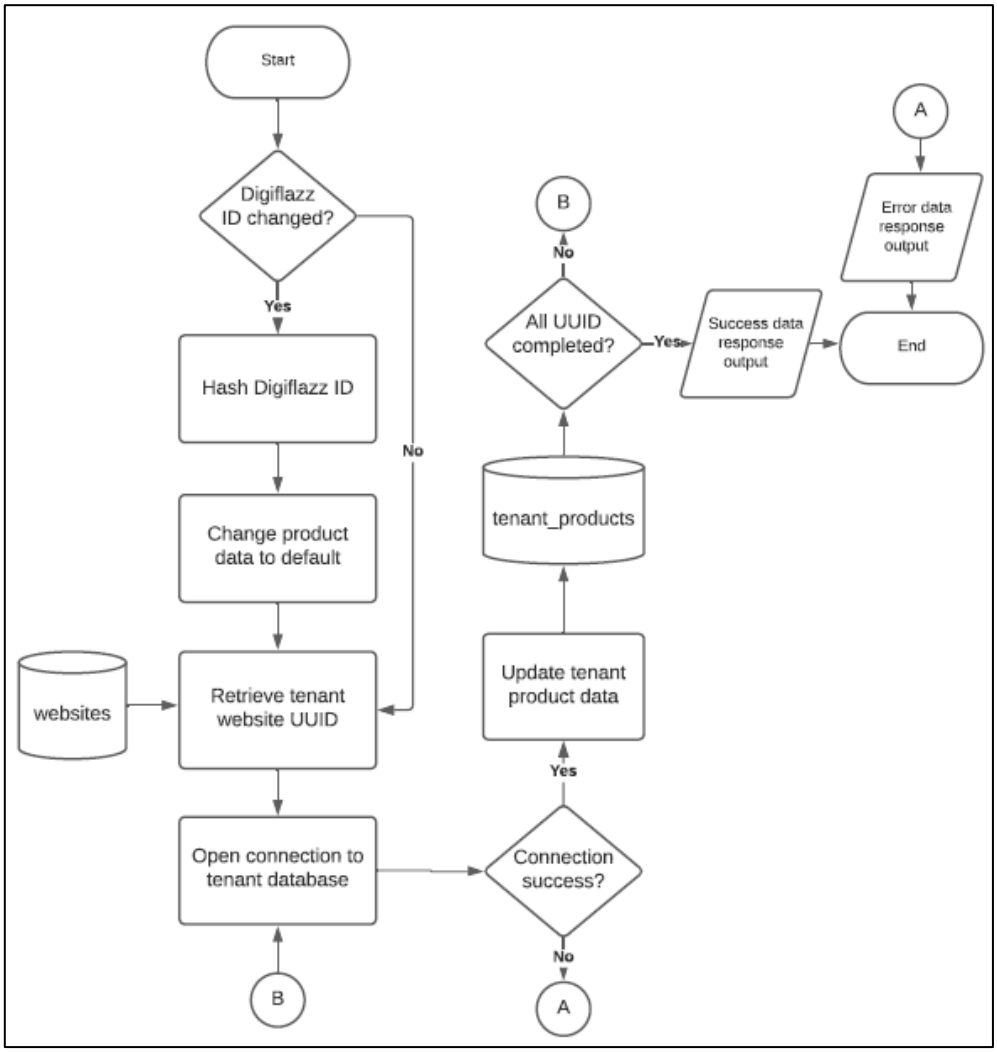
Gambar 3.9 Flowchart Perubahan Otomatis Database

Setelah menerima data perubahan produk, program akan melakukan validasi data yang diperlukan. Jika validasi gagal, maka program akan mengirim respon *error* kepada bagian *front-end* situs *Superadmin* Whitelabel. Jika validasi berhasil, program akan melakukan pembaharuan data produk kepada *database* sistem Whitelabel sesuai dengan data baru yang diterima. *Database* sistem Whitelabel merupakan sebuah *database* utama yang dimiliki oleh Whitelabel. *Database* ini menyimpan seluruh data sistem Whitelabel, terutama data terkait seluruh *tenant* yang ada, data seluruh operator

layanan yang tersedia, dan data seluruh produk yang tersedia. Setelah melakukan pembaharuan data produk di *database* sistem Whitelabel, program akan melakukan perubahan data produk di setiap *database* tenant yang ada.

A.2.1 Flowchart Perubahan Otomatis Database Tenant

Gambar 3.10 menggambarkan alur kerja pada saat program melakukan perubahan data otomatis kepada setiap *database* tenant.



Gambar 3.10 Flowchart Perubahan Otomatis Database

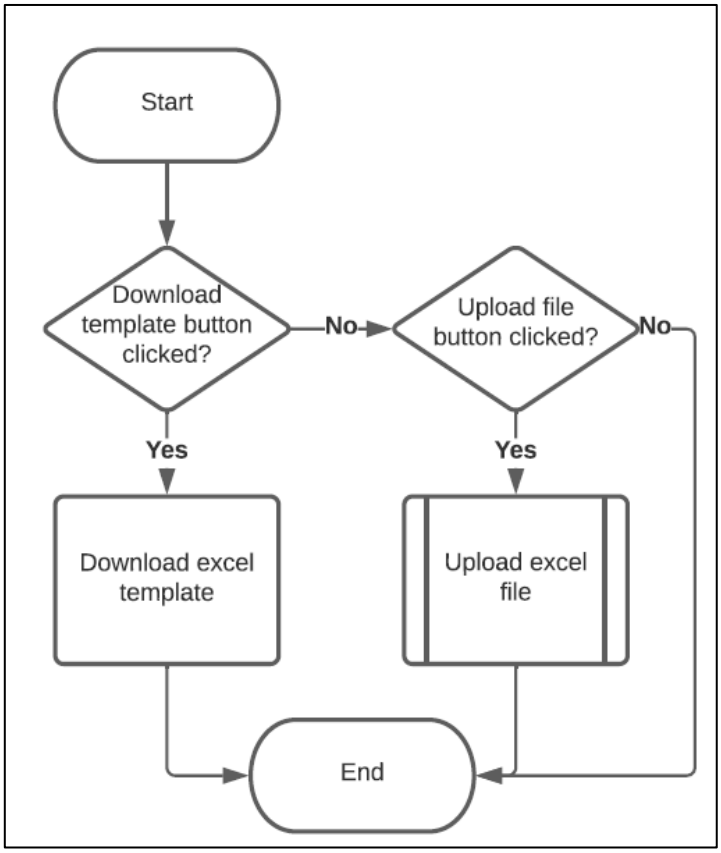
Perubahan otomatis *database tenant* dimulai dengan pemeriksaan perubahan ID Digiflazz. ID Digiflazz merupakan penanda unik pada sebuah produk jika produk tersebut memiliki pemasok Digiflazz. ID ini berfungsi untuk mempermudah integrasi dan transaksi dengan pemasok Digiflazz. Jika ID Digiflazz diperbaharui, maka ada langkah tambahan yang perlu dilakukan. Langkah ini adalah melakukan *hash* kepada ID Digiflazz baru tersebut, dan juga melakukan tambahan perbaharuan data produk, yaitu dengan mengubah data – data produk tersebut menjadi ke pengaturan awal atau *default* Whitelabel. *Hashing* perlu dilakukan kepada ID Digiflazz karena data tersebut akan digunakan pada saat melakukan transaksi dengan API Digiflazz. Jika ID Digiflazz tidak berubah, maka tidak ada langkah tambahan.

Proses perubahan dilanjutkan dengan melakukan pengambilan data *universally unique identifier* (UUID) dari seluruh situs *tenant* yang aktif. UUID ini diperlukan untuk melakukan koneksi kepada setiap *database tenant*. Setelah itu, koneksi dapat dilakukan kepada setiap *database tenant* yang aktif dan perubahan data produk dapat dilakukan kepada setiap *database tenant* tersebut. Jika ada koneksi kepada *database tenant* yang gagal, maka program akan memberikan respon *error* kepada bagian *front-end* situs *Superadmin* Whitelabel. Program akan mengirimkan respon sukses jika seluruh koneksi dan perubahan berhasil dilakukan kepada setiap *database tenant*.

A.3 Flowchart Fitur Top Up By Excel

Fitur *Top Up By* Excel merupakan fitur yang memungkinkan pelanggan *tenant* Whitelabel untuk melakukan transaksi seluruh produk prabayar dengan cara mengunggah *file* Excel. Fitur ini sangat bermanfaat untuk melakukan transaksi dalam jumlah besar dengan nomor tujuan yang berbeda – beda, karena pelanggan dapat dengan mudah dan cepat mengisi banyak data permintaan transaksi kedalam sebuah *file* Excel. Dengan adanya fitur ini, transaksi dalam jumlah banyak dapat dilakukan dengan lebih efektif dan efisien.

Gambar 3.11 menggambarkan alur kerja halaman *Top Up By* Excel secara keseluruhan.

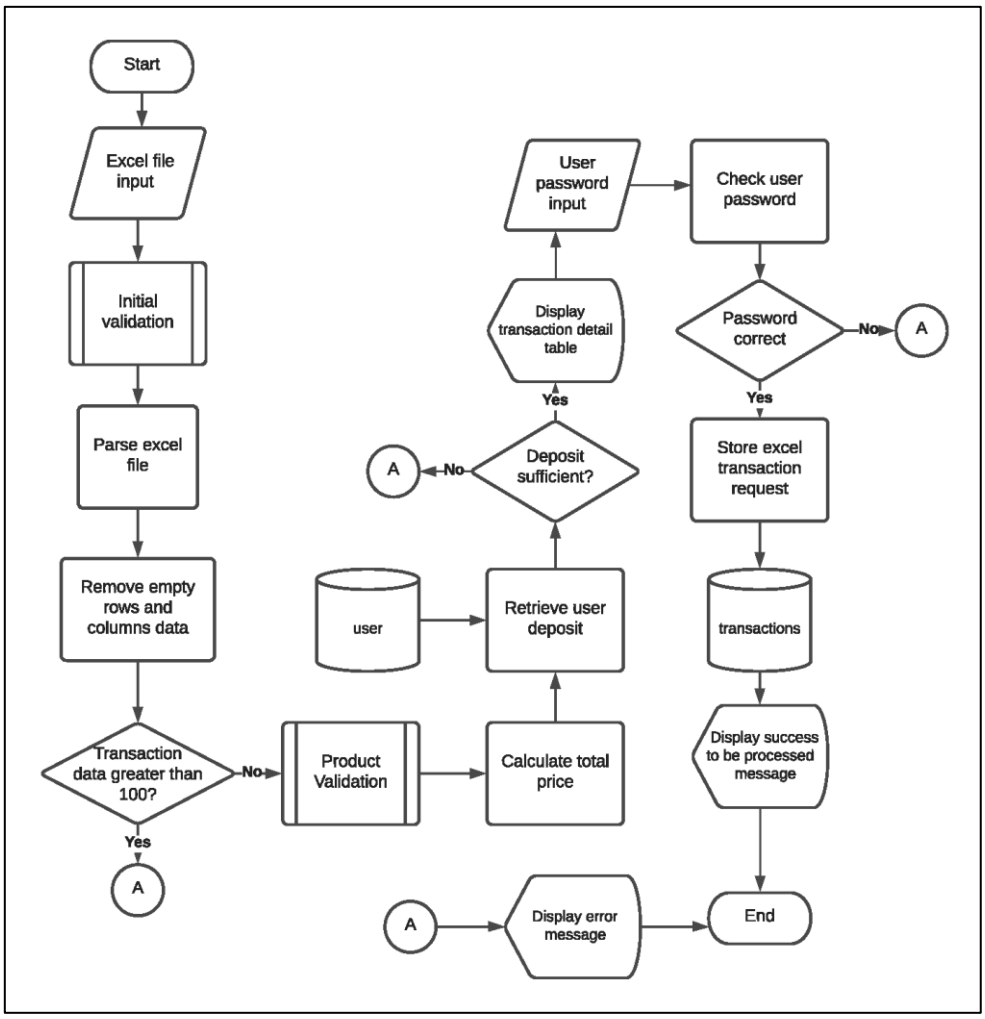


Gambar 3.11 Flowchart Halaman *Top Up By* Excel

Pada halaman ini, pengguna dapat melakukan dua hal, yaitu mengunduh contoh *template file* Excel, dan mengunggah *file* Excel untuk melakukan transaksi. Sebelum, melakukan transaksi, pengguna diinstruksikan untuk mengunduh *template file* Excel yang tersedia. Dengan melakukan pengunduhan *template file* Excel pengguna dapat menggunakan *file* tersebut untuk mengisi data – data permintaan transaksi sebelum diunggah kembali. Pada *template file* Excel tersebut juga tersedia berbagai panduan pengisian *file* Excel agar *format file* Excel yang akan diunggah sesuai dengan aturan *format* yang diterima.

A.3.1 Flowchart Pemrosesan File Excel

Gambar 3.12 menggambarkan alur kerja program secara umum setelah pengguna melakukan pengunggahan *file* Excel untuk melakukan permintaan *Top Up by* Excel.



Gambar 3.12 Flowchart Pemrosesan *File* Excel

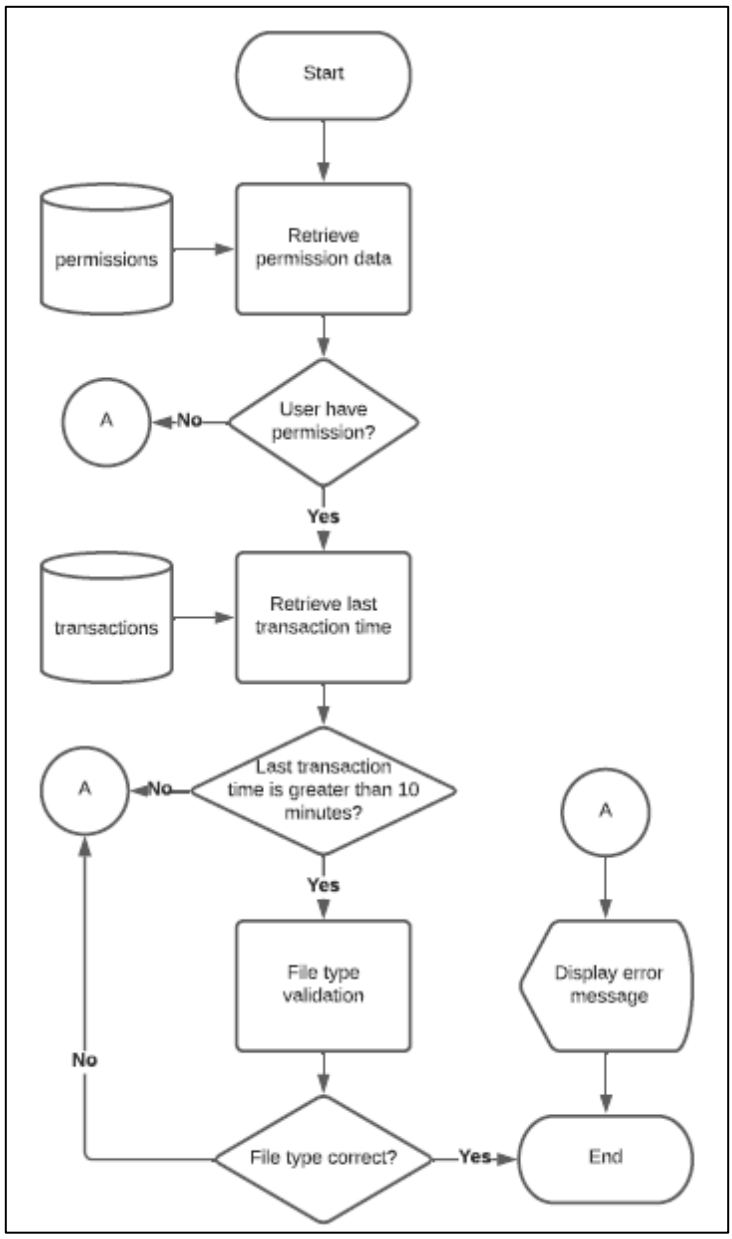
Setelah pengguna mengunggah *file* Excel, program akan melakukan berbagai macam validasi dan ekstraksi data dari *file* tersebut. Validasi pertama adalah validasi yang bersifat umum. Setelah selesai melakukan validasi pertama, ekstraksi data dari *file* akan dilakukan. Data yang telah didapatkan dari *file* tersebut kemudian akan di validasi kembali. Validasi selanjutnya merupakan validasi isi data, jumlah data, dan validasi produk yang diminta. Fitur *Top Up by* Excel Whitelabel akan menerima

maksimal 100 transaksi dalam sekali pengunggahan *file* Excel. Jika jumlah transaksi melebihi 100, maka pesan *error* akan ditampilkan.

Setelah seluruh validasi lanjutan selesai dilakukan, program akan melakukan perhitungan total harga seluruh transaksi. Total harga tersebut akan dibandingkan dengan jumlah saldo *deposit* pengguna. Jika saldo tidak mencukupi, maka pesan *error* akan ditampilkan. Jika saldo cukup, maka seluruh detail permintaan transaksi akan ditampilkan kepada pengguna sebagai bentuk konfirmasi produk yang ingin dibeli. Kemudian pengguna dapat melakukan validasi terakhir dalam pemrosesan *file* Excel, yaitu validasi *password* pengguna. Jika pengguna memasukkan *password* dengan salah, maka pesan *error* akan ditampilkan. Jika *password* yang dimasukkan pengguna benar program akan melakukan penyimpanan data permintaan transaksi ke *database* untuk menunggu diproses. Jika seluruh proses telah berjalan dengan benar, maka pesan sukses akan ditampilkan.

A.3.2 Flowchart Initial Validation

Gambar 3.13 menggambarkan alur kerja program ketika melakukan validasi *file* Excel secara umum.



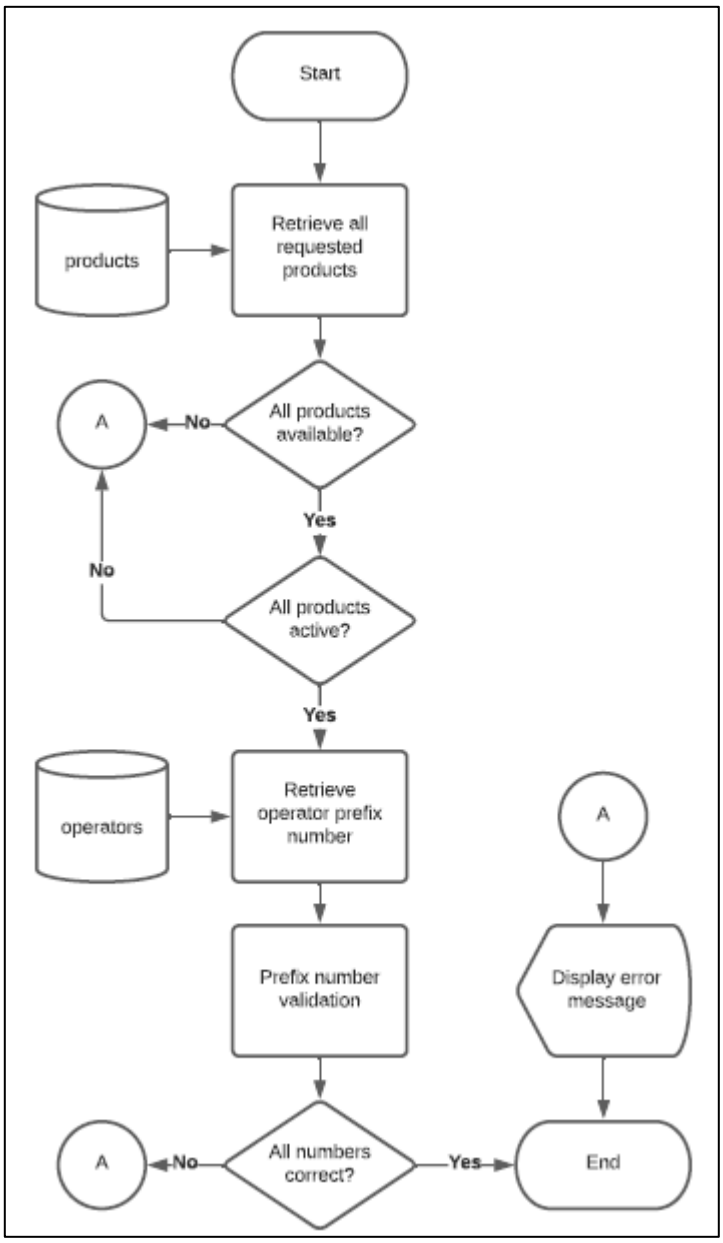
Gambar 3.13 Flowchart Initial Validation

Validasi pertama *file* Excel dimulai dari validasi *permission* yang dimiliki oleh pengguna, karena tidak semua pengguna dapat melakukan transaksi menggunakan *file* Excel. *Permission* ini dapat diatur oleh tenant situs Whitelabel yang digunakan

pengguna. Jika pengguna tidak memiliki *Permission* maka pesan *error* akan ditampilkan. Setelah itu, validasi jeda waktu akan dilakukan. Jeda minimal yang harus ada antar unggahan permintaan *Top Up by* Excel adalah 10 menit. Jika jeda waktu unggahan kurang dari 10 menit dari unggahan yang sebelumnya, maka pesan *error* akan ditampilkan. Validasi umum terakhir adalah validasi terkait tipe *file* yang diunggah. Jika pengguna melakukan pengunggahan *file* selain Excel, maka pesan error akan ditampilkan.

A.3.3 Flowchart Product Validation

Gambar 3.14 menggambarkan alur kerja program ketika melakukan validasi produk permintaan transaksi.



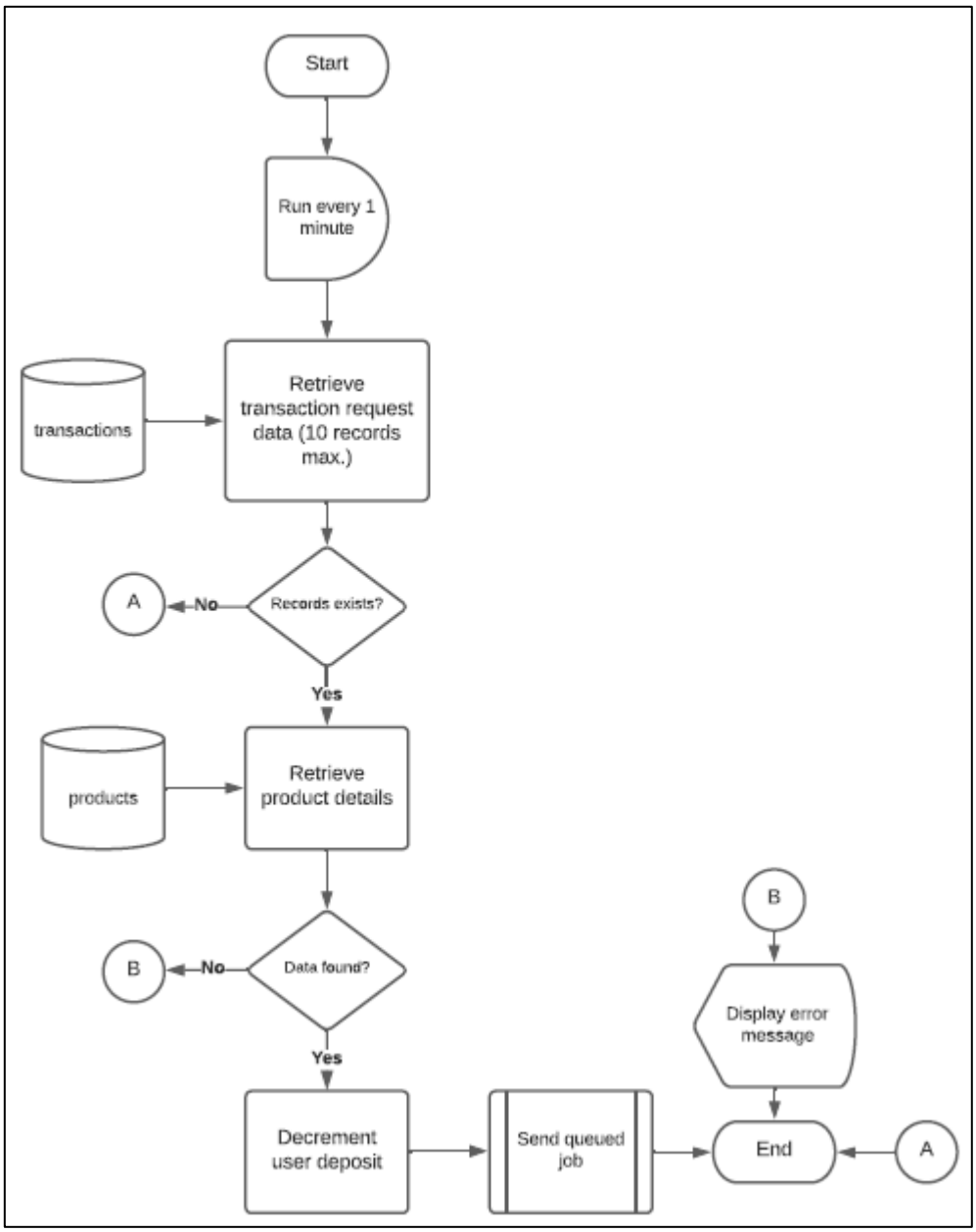
Gambar 3.14 Flowchart Product Validation

Validasi lain yang perlu dilakukan dalam fitur ini adalah validasi produk – produk transaksi yang diminta oleh pengguna di *file* Excel. Validasi produk diawali dengan pengambilan data setiap produk dari *database.* Jika ada salah satu produk yang

tidak ditemukan, maka pesan *error* akan ditampilkan. Validasi selanjutnya adalah validasi status produk yang telah ditemukan. Jika produk yang diminta ada di *database* namun tidak dalam kondisi aktif (dimatikan oleh *tenant*) maka pesan *error* akan ditampilkan. Validasi terakhir pada validasi produk adalah validasi setiap nomor tujuan yang di minta oleh pengguna, dengan awalan (*prefix*) nomor operator produk yang diminta. *Prefix* nomor adalah tiga sampai empat digit pertama dari nomor tujuan. Validasi ini diperlukan karena setiap operator memiliki *prefix* nomor yang berbeda – beda, sehingga nomor tujuan harus sesuai dengan nomor operator produk yang diminta. Jika nomor tujuan tidak sesuai dengan operator produk, pesan *error* akan ditampilkan.

A.3.4 Flowchart Cron Job Pemrosesan Permintaan Transaksi

Gambar 3.15 menggambarkan alur kerja program ketika melakukan pemrosesan permintaan transaksi.



Gambar 3.15 Flowchart *Cron Job* Pemrosesan Permintaan Transaksi

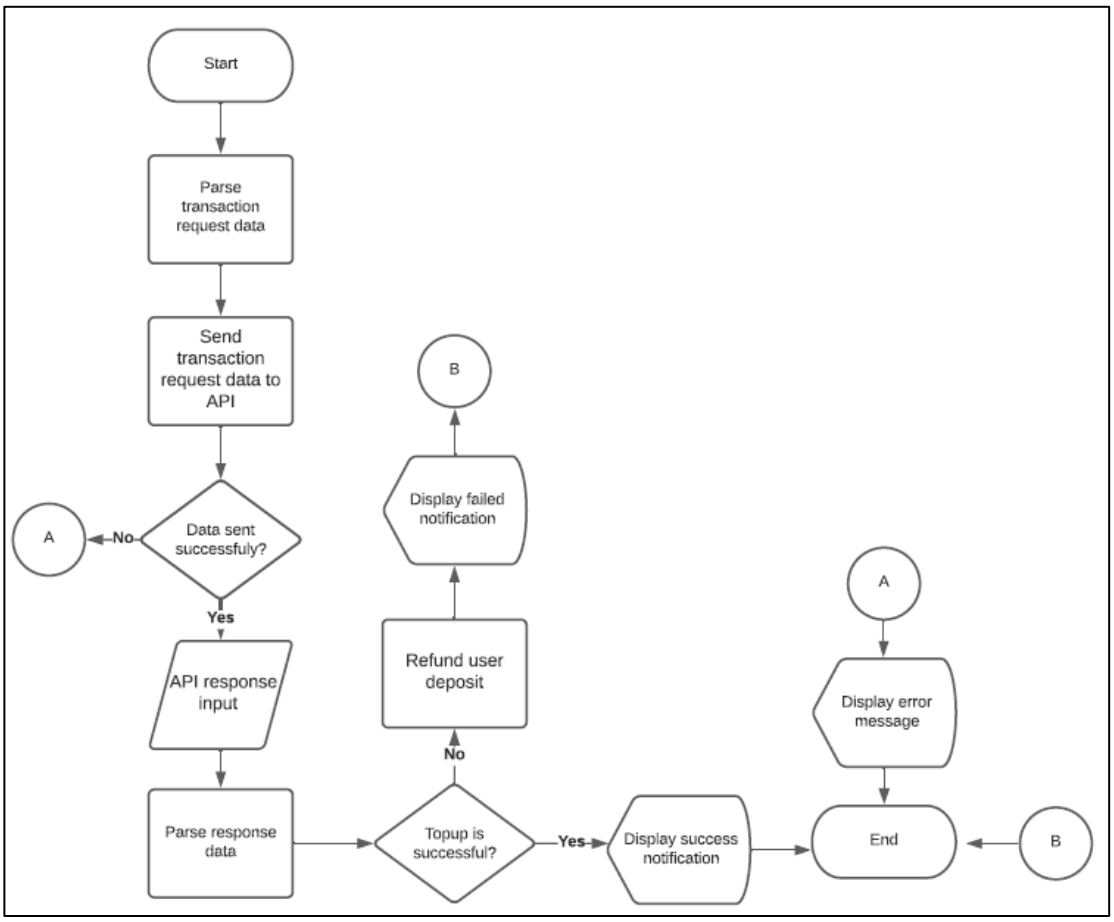
Pemrosesan permintaan transaksi melalui fitur *Top Up by* Excel pada whitelabel akan diproses menggunakan *Cron Job* yang disediakan oleh *Framework* Laravel. *Cron Job* ini akan dieksekusi setiap satu menit sekali. Dalam satu menit

tersebut, jika tersedia data transaksi yang sedang menunggu di proses, sebanyak maksimal 10 permintaan transaksi akan diproses. Hal ini bertujuan untuk menyeimbangkan beban *server* dan permintaan API yang perlu diproses oleh sistem Whitelabel. Sisa transaksi yang belum diproses kemudian akan diproses pada menit selanjutnya sesuai dengan aturan jumlah maksimal tiap menitnya.

Permintaan transaksi ini didapatkan dari data yang telah dimasukan ke dalam *database* pada langkah pengunggahan *file* Excel. Pada saat program melakukan pengambilan maksimal 10 permintaan transaksi, transaksi – transaksi tersebut juga akan diberikan penanda agar tidak diproses kembali pada menit selanjutnya. Data permintaan transaksi yang diambil tersebut kemudian akan digunakan untuk melakukan pengambilan data *detail* produk untuk nanti dikirimkan kepada API pemasok. Jika *detail* tidak berhasil diambil, maka pesan *error* akan muncul. Jika *detail* berhasil diambil, maka saldo *deposit* pengguna yang melakukan permintaan transaksi akan dipotong sesuai dengan harga total transaksi. Langkah terakhir adalah mengirimkan permintaan transaksi yang ada kepada antrian (*queue*) untuk nanti dikirimkan kepada API pemasok.

A.3.5 Flowchart Queue Pemrosesan Transaksi

Gambar 3.16 menggambarkan alur kerja program ketika melakukan pemrosesan transaksi.



Gambar 3.16 Flowchart *Queue* Pemrosesan Transaksi

Pemrosesan transaksi dilakukan menggunakan sistem antrian (*Queue*) yang disediakan oleh *Framework* Laravel. Hal ini dilakukan agar proses transaksi dapat dilakukan satu persatu dari awal hingga akhir tanpa ada gangguan atau *collision* dengan permintaan transaksi lain. Jika sebuah transaksi sedang diproses dan kemudian mendapatkan permintaan transaksi lain, permintaan transaksi tersebut akan disimpan untuk menunggu dalam antrian dan akan proses dengan sistem *First Come First Serve* (FCFS) atau nama lainnya *First In First Out* (FIFO).

Langkah pertama dalam pemrosesan transaksi adalah melakukan pengubahan data permintaan transaksi menjadi *format* JSON (*JavaScript Objet Notation*) agar dapat diproses oleh API pemasok. Data tersebut kemudian dikirim kepada pemasok melalui API. Jika data tidak berhasil dikirim, maka pesan *error* akan ditampilkan pada laporan transaksi. Jika data berhasil dikirim, maka program akan menunggu respon balasan dari API pemasok tersebut.

Setelah program mendapatkan balasan dari API pemasok, program akan kembali melakukan pengubahan *format* data agar dapat diolah dengan baik. Melalui data respon tersebut, program dapat mengetahui apakah transaksi yang telah diproses tersebut berhasil atau tidak. Jika transaksi berhasil, maka pesan berhasil dan notifikasi beserta laporan akan ditampilkan, namun jika transaksi gagal, maka saldo *deposit* pengguna akan dikembalikan. Selain itu pengguna juga akan mendapatkan notifikasi beserta laporan yang menginformasikan *detail* alasan kegagalan transaksi tersebut.

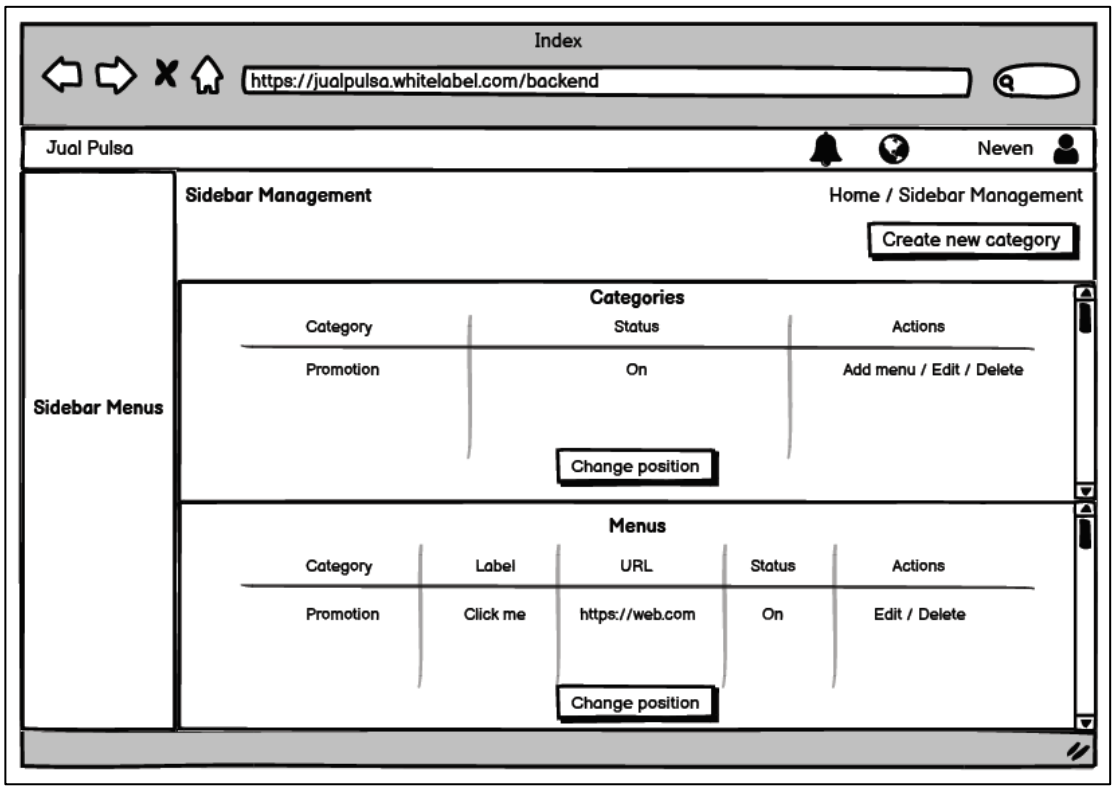
B. Rancangan User Interface

B.1 Rancangan User Interface Fitur Menu Sidebar Custom

Rancangan tampilan fitur menu *sidebar custom* dibagi menjadi beberapa bagian sesuai dengan halaman setiap fungsionalitas yang secara garis besar dibagi menjadi *create*, *edit*, dan *delete*.

B.1.1 Rancangan User Interface Halaman Utama

Gambar 3.17 menggambarkan tampilan halaman utama dari fitur menu *sidebar custom*.

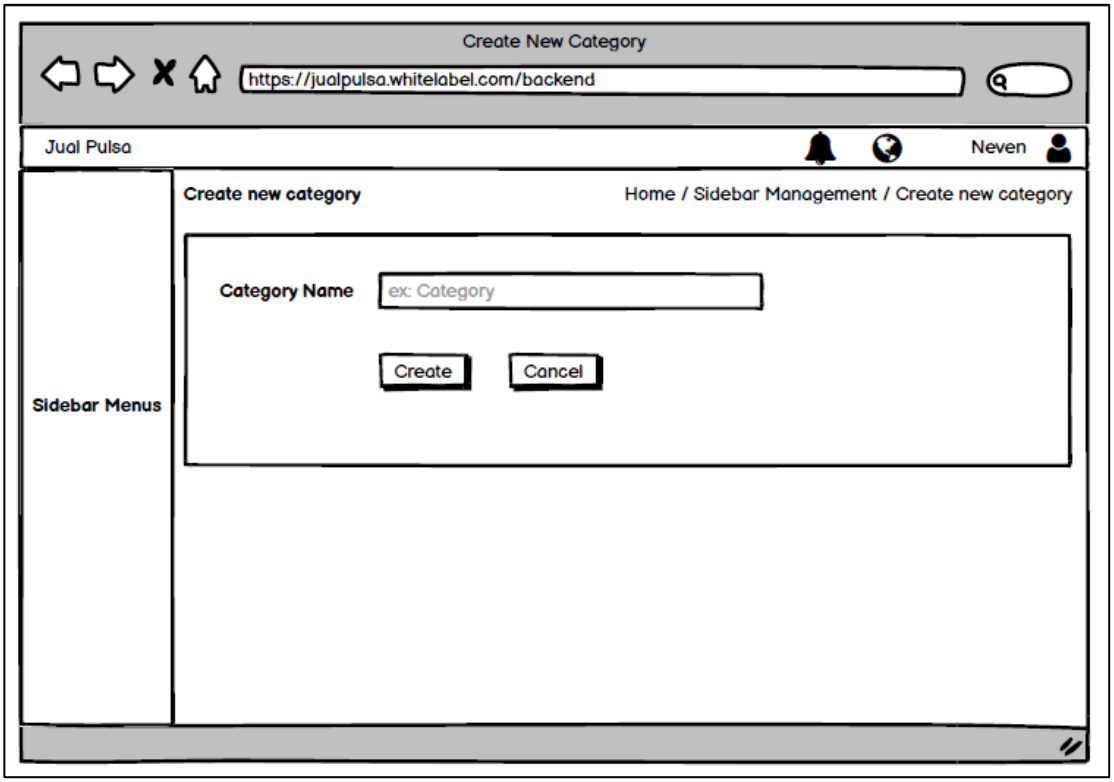


Gambar 3.17 Rancangan Halaman Utama Fitur Menu *Sidebar Custom*

Pada menu utama ini, pengguna dapat melihat seluruh data kategori dan menu yang telah ada. Pengguna juga dapat melihat detail seluruh data kategori dan menu yang telah dibuat yang disajikan dalam bentuk tabel. Tombol "*Create new category*" digunakan untuk membuat kategori baru. Tombol "*Change position*" digunakan untuk melihat posisi kategori atau menu yang aktif dan merubah urutan posisi tersebut. Pada kolom *actions* terdapat beberapa tombol yang dapat melakukan fungsionalitas seperti menambahkan menu ke dalam kategori, mengubah data kategori, mengubah data menu, menghapus kategori, dan juga menghapus menu.

B.1.2 Rancangan User Interface Halaman Create New Category

Gambar 3.18 menggambarkan tampilan halaman *create new category* dari fitur menu *sidebar custom*.

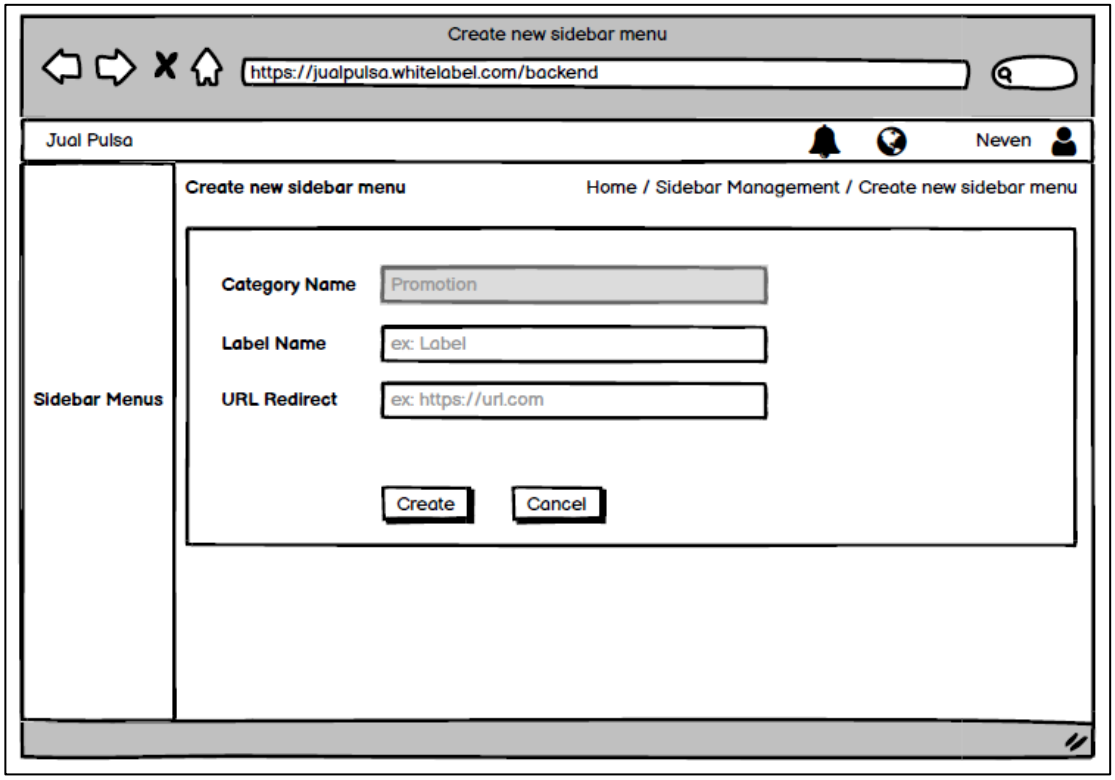


Gambar 3.18 Rancangan Halaman *Create New Category* Fitur Menu *Sidebar Custom*

Pada menu ini, pengguna dapat menambah data kategori baru dengan cara memasukkan nama kategori pada kolom *input* yang telah disediakan. Jika telah mengisinya, pengguna dapat menekan tombol "*Create*" untuk membuat kategori baru. Tombol "*Cancel*" digunakan untuk membawa pengguna ke halaman utama.

B.1.3 Rancangan User Interface Halaman Create New Menu

Gambar 3.19 menggambarkan tampilan halaman *create new menu* dari fitur menu *sidebar custom*.

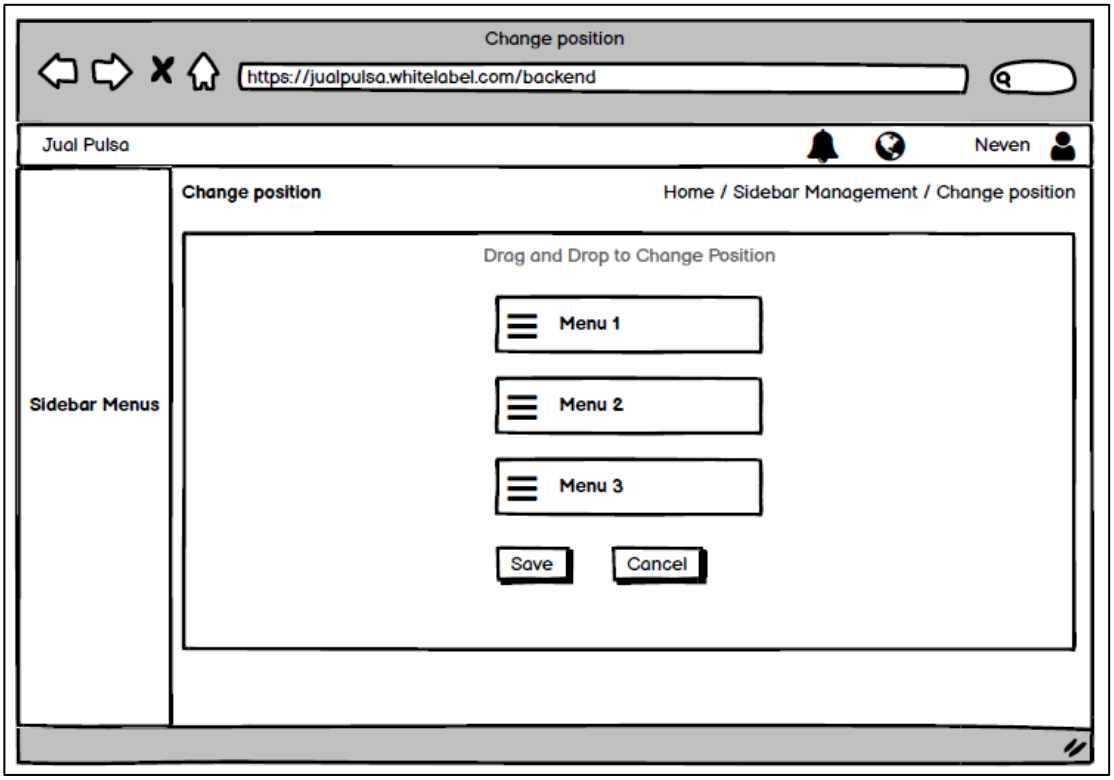


Gambar 3.19 Rancangan Halaman *Create New Menu* Fitur Menu *Sidebar Custom*

Pada menu ini, pengguna dapat menambah data menu baru dengan cara memasukkan nama label dan URL *redirect* pada kolom *input* yang telah disediakan. Jika telah mengisinya, pengguna dapat menekan tombol "*Create*" untuk membuat menu baru. Tombol "*Cancel*" digunakan untuk membawa pengguna ke halaman utama. Nama kategori dari menu yang akan dibuat juga ditampilkan sebagai pelengkap informasi.

B.1.4 Rancangan User Interface Halaman Change Position

Gambar 3.20 menggambarkan tampilan halaman *Change Position* dari fitur menu *sidebar custom*.

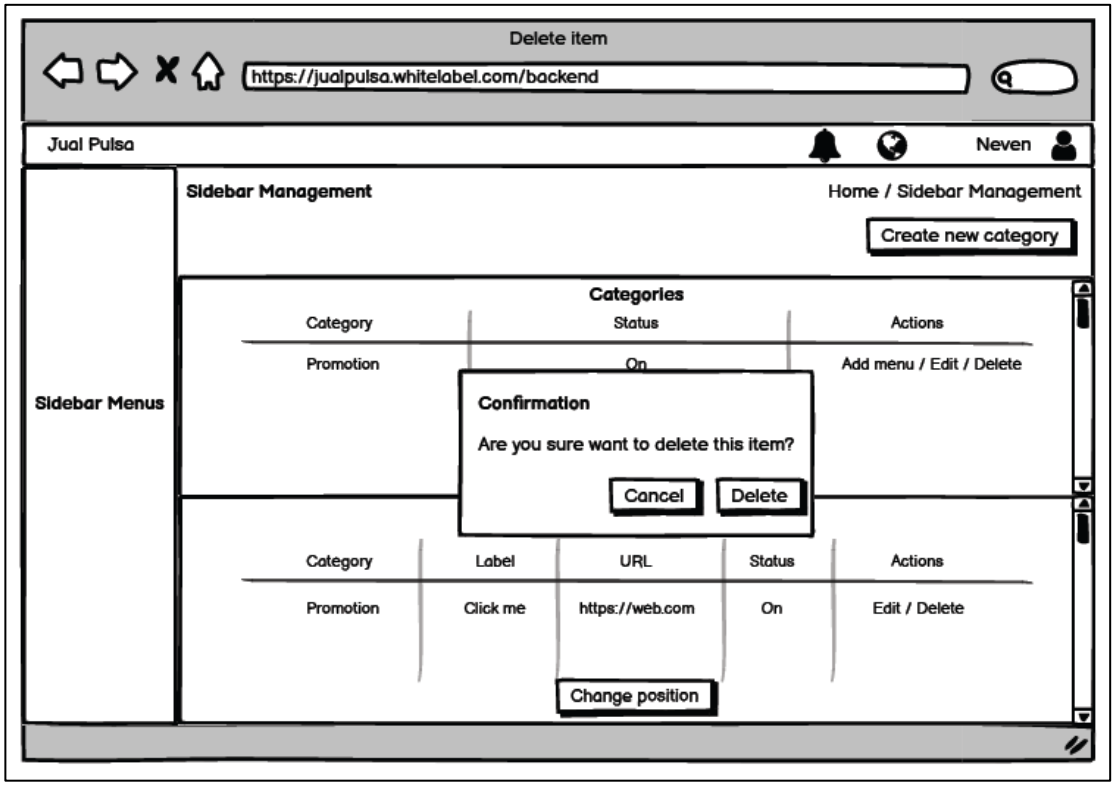


Gambar 3.20 Rancangan Halaman *Change Position* Fitur Menu *Sidebar Custom*

Pada menu ini, pengguna dapat melihat posisi urutan kategori atau menu yang sedang tampil. Untuk merubah posisi urutan tampil kategori atau menu tersebut, pengguna dapat melakukan *drag and drop* ke arah atas dan bawah pada komponen kategori atau menu yang ditampilkan sesuai dengan keinginan pengguna. Tombol "*Save*" digunakan untuk menyimpan perubahan posisi yang telah ditentukan pengguna. Tombol "*Cancel*" digunakan untuk membawa pengguna ke halaman utama.

B.1.5 Rancangan User Interface Dialog Konfirmasi Penghapusan

Gambar 3.21 menggambarkan tampilan *dialog* konfirmasi penghapusan dari fitur menu *sidebar custom*.



Gambar 3.21 Rancangan *Dialog* Konfirmasi Penghapusan Fitur Menu *Sidebar Custom*

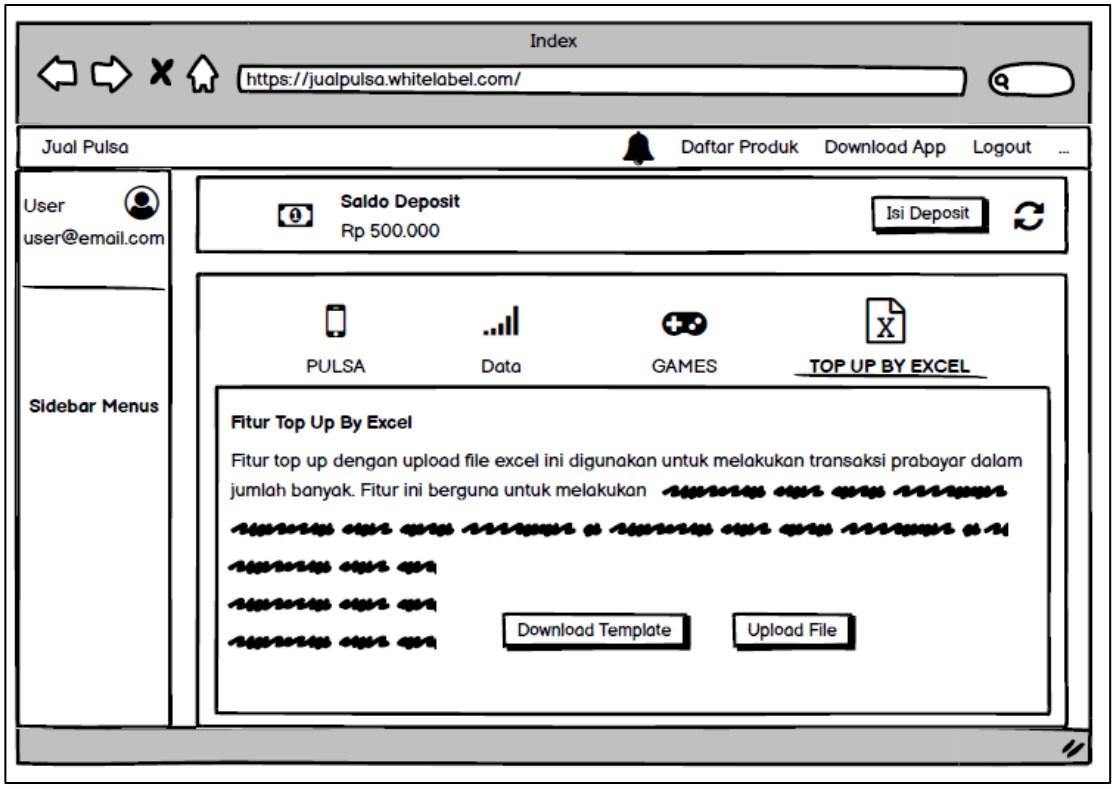
Jika pengguna menekan tombol hapus pada kategori atau menu, sebuah *dialog* konfirmasi penghapusan akan muncul. Tombol "*Delete*" digunakan untuk menghapus kategori atau menu yang dipilih. Tombol "*Cancel*" digunakan untuk membatalkan penghapusan kategori atau menu yang dipilih.

B.2 Rancangan User Interface Fitur Top Up By Excel

Rancangan tampilan fitur *Top Up by* Excel dibagi menjadi beberapa bagian yaitu rancangan tampilan halaman awal, rancangan tabel konfirmasi, dan rancangan pesan validasi.

B.2.1 Rancangan User Interface Halaman Awal

Gambar 3.22 menggambarkan tampilan halaman awal dari fitur *Top Up by* Excel.

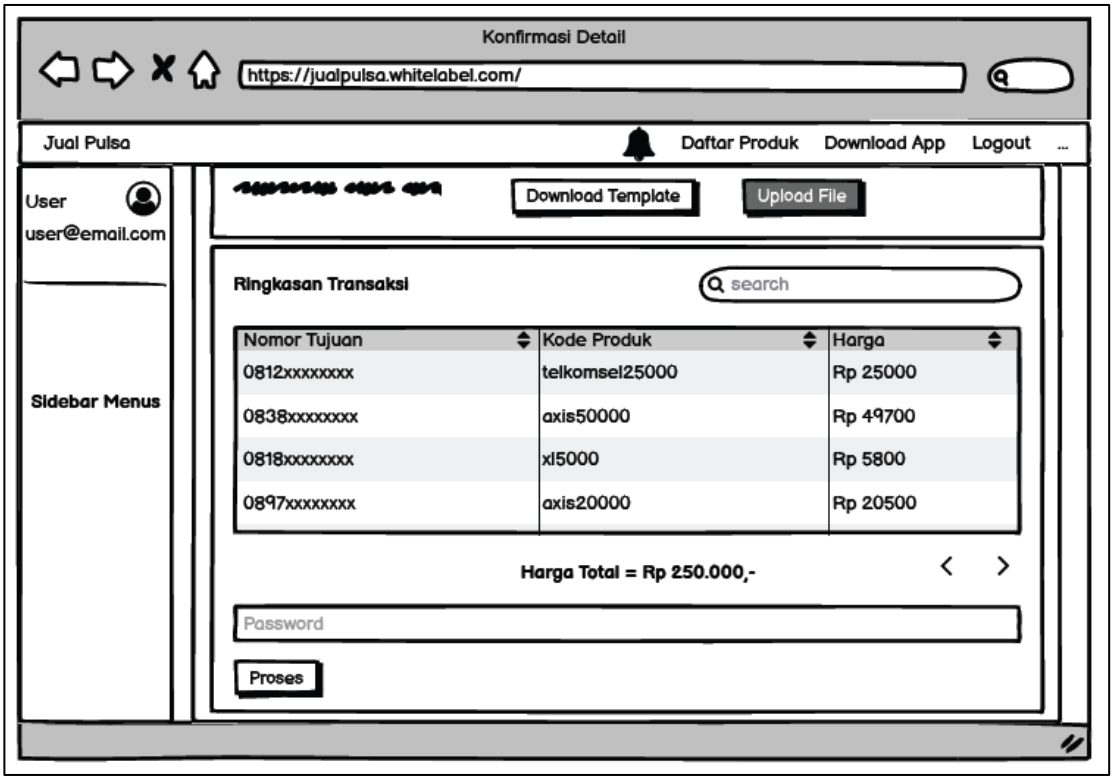


Gambar 3.22 Rancangan Halaman Awal Fitur *Top Up by* Excel

Pada halaman awal, disediakan penjelasan singkat terkait fungsi fitur *Top Up by* Excel, tata cara menggunakannya, dan beberapa himbauan terkait penggunaan fitur ini. Pada halaman ini, pengguna juga dapat mengunduh *file template* Excel yang telah disediakan, dan juga melakukan pengunggahan *file* Excel yang telah diisi. Tombol "*Download Template*" digunakan untuk mengunduh *file template*. Tombol "*Upload File*" digunakan untuk mengunggah *file* Excel permintaan transaksi prabayar.

B.2.2 Rancangan User Interface Tabel Konfirmasi Transaksi

Gambar 3.23 menggambarkan tampilan tabel konfrmasi transaksi dari fitur *Top Up by* Excel.

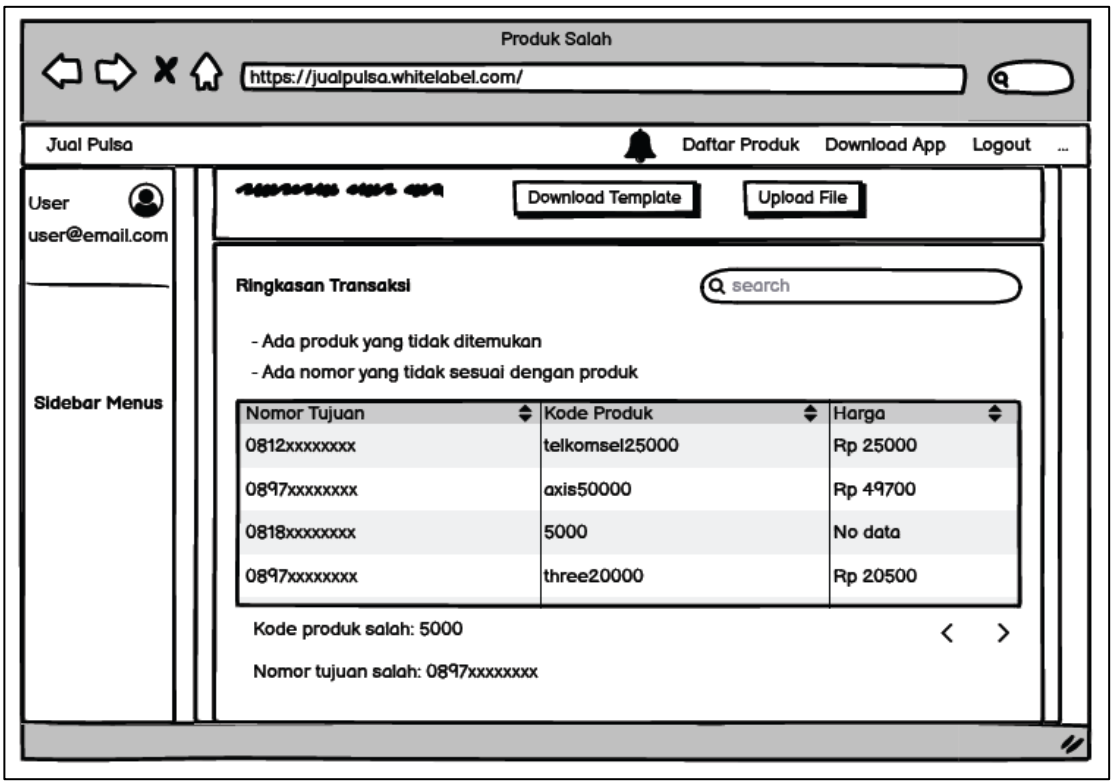


Gambar 3.23 Rancangan Tabel Konfirmasi Fitur *Top Up by* Excel

Setelah pengguna mengunggah *file* Excel permintaan transaksi dengan benar, tabel konfirmasi transaksi akan muncul pada bagian bawah halaman awal. Tabel ini menyediakan informasi terkait transaksi yang akan dilakukan pengguna. Informasi tersebut berupa nomor tujuan, kode produk, harga setiap produk, dan harga total. Pengguna juga dapat melakukan pengurutan data pada setiap kolom dan melakukan pencarian data dengan kata kunci pada kolom *input* yang telah disediakan. Jika pengguna telah mengkonfirmasi data – data permintaan transaksi tersebut, pengguna dapat mengisi *password* akun miliknya pada kolom *input* yang telah disediakan. Tombol "Proses" digunakan untuk mengkonfirmasi permintaan transaksi dan memproses transaksi tersebut.

B.2.3 Rancangan User Interface Validasi Permintaan Transaksi

Gambar 3.24 menggambarkan tampilan pesan validasi permintaan transaksi dari fitur *Top Up by* Excel.



Gambar 3.24 Rancangan Pesan Validasi Fitur *Top Up by* Excel

Setelah pengguna mengunggah *file* Excel permintaan transaksi, jika ada *format* penulisan, nomor tujuan, dan kode produk yang salah, maka pesan kesalahan akan muncul. Pesan kesalahan tersebut akan menjelaskan secara singkat kesalahan apa yang terjadi. Jika saldo pengguna tidak cukup dan jika terjadi kesalahan lainnya, pesan kesalahan yang sesuai juga akan muncul. Jika muncul pesan kesalahan, pengguna tidak dapat memasukkan *password* dan memproses transaksi, sehingga pengguna perlu melakukan pengunggahan *file* Excel kembali.

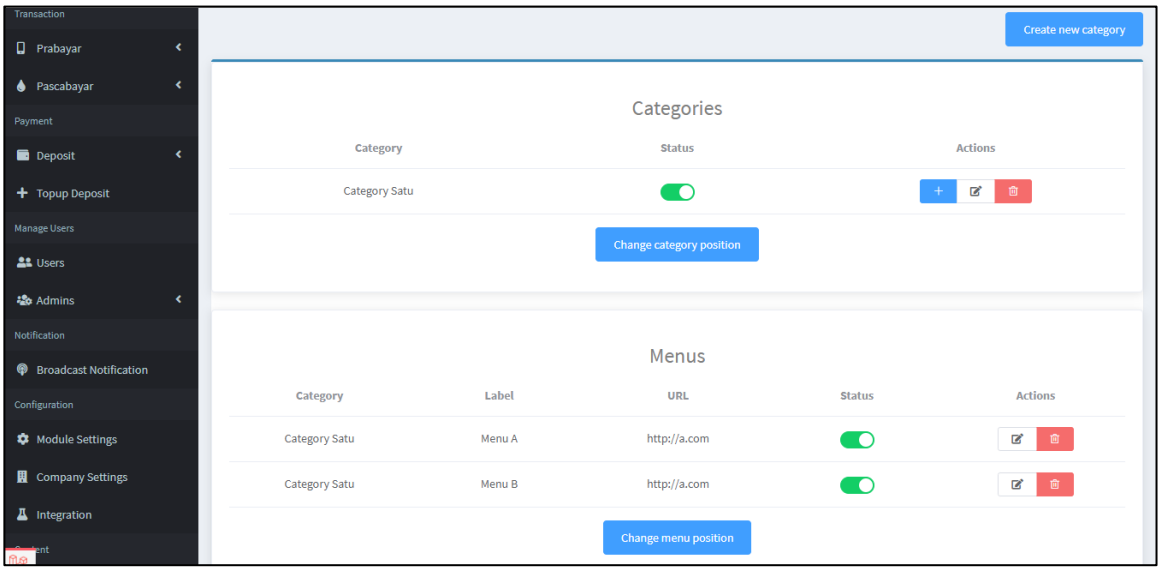
3.3.4 Implementasi

A. Implementasi Fitur Menu Sidebar Custom

Implementasi fitur menu *sidebar custom* dibagi menjadi beberapa bagian sesuai dengan halaman setiap fungsionalitas yang ada. Fungsionalitas – fungsionalitas tersebut adalah memperlihatkan data seluruh kategori dan menu (halaman utama), *create new category*, *create new menu*, *edit category*, *edit menu*, *change item position*, dan *delete item*.

A.1 Implementasi Halaman Utama Fitur Menu Sidebar Custom

Gambar 3.25 merupakan implementasi halaman utama fitur menu *sidebar custom*.



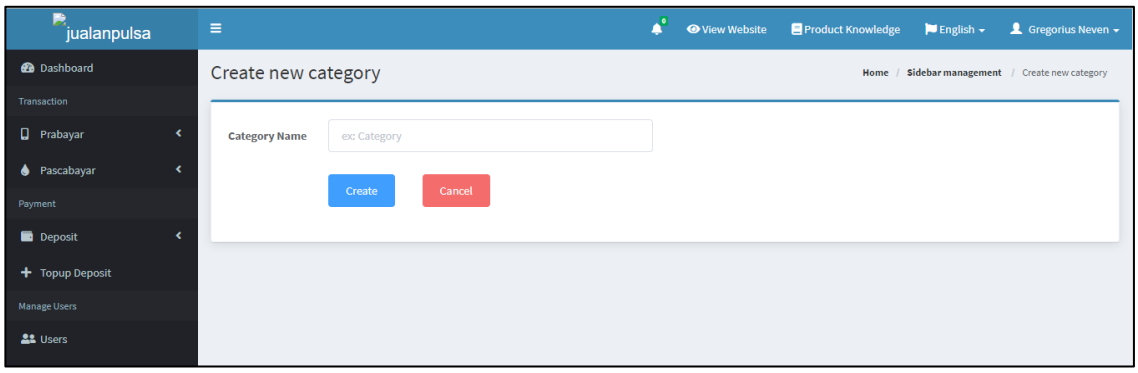
Gambar 3.25 Implementasi Halaman Utama Fitur Menu *Sidebar Custom*

Halaman utama pada fitur Menu *Sidebar Custom* akan memperlihatkan daftar seluruh kategori dan menu yang telah dibuat oleh pengguna yang disajikan dalam bentuk tabel. Tabel *Categories* menampilkan nama kategori, status kategori, dan tombol – tombol aksi yang dapat ditekan oleh pengguna untuk melakukan aksi tertentu pada kolom *actions*. Pada kolom *actions*, Tombol berwarna biru digunakan untuk menambahkan data menu ke dalam kategori tersebut. Tombol berwarna putih digunakan untuk merubah data kategori yang telah dibuat. Tombol berwarna merah digunakan untuk menghapus kategori. Tombol "*Change category position*" digunakan untuk melakukan perubahan urutan tampil kategori yang telah dibuat. Untuk menambah kategori, pengguna dapat menggunakan tombol "*Create new category*" yang terdapat pada bagian kanan atas halaman.

Tabel *Menus* menampilkan data – data menu yang telah dibuat seperti, nama kategori menu tersebut, nama menu, URL *redirect* menu, status menu, dan tombol – tombol aksi untuk melakukan aksi tertentu pada kolom *actions*. Tombol berwarna putih digunakan untuk merubah data menu. Tombol berwarna merah digunakan untuk menghapus menu. Tombol "*Change menu position*" digunakan untuk melakukan perubahan urutan tampil kategori yang telah dibuat. Selain itu, pengguna juga dapat menekan *switch* pada kolom status pada kedua tabel untuk melakukan perubahan status kategori atau menu yang dipilih (aktif / tidak aktif).

A.2 Implementasi Halaman Create New Category Menu Sidebar Custom

Gambar 3.26 merupakan implementasi halaman *create new category* fitur menu *sidebar custom*.



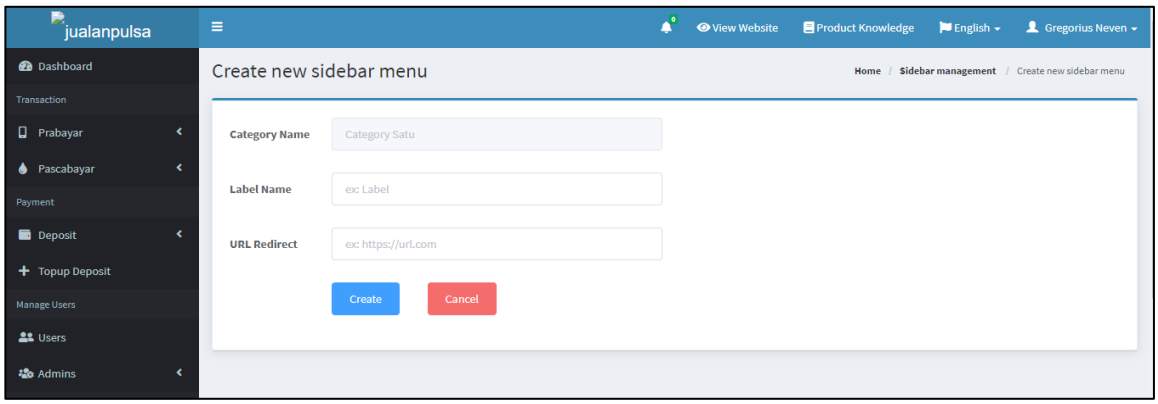
Gambar 3.26 Implementasi Halaman *Create New Category*

Fitur Menu *Sidebar Custom*

Pada halaman *create new category* pengguna dapat menambahkan data kategori menu. Pengguna dapat mengisi nama kategori pada kolom *input* yang telah disediakan dan menekan tombol "*Create*" untuk menambahkan data kategori baru. Tombol "*Cancel*" digunakan untuk kembali ke halaman utama. Batas maksimal kategori yang dapat dibuat oleh pengguna adalah satu. Jika pengguna memasukkan nama kategori dengan *format* yang salah atau melakukan pembuatan kategori lebih dari satu, maka pesan kesalahan akan diperlihatkan, dan proses penambahan data kategori akan gagal.

A.3 Implementasi Halaman Create New Menu Fitur Menu Sidebar Custom

Gambar 3.27 merupakan implementasi halaman *create new menu* fitur menu *sidebar custom*.



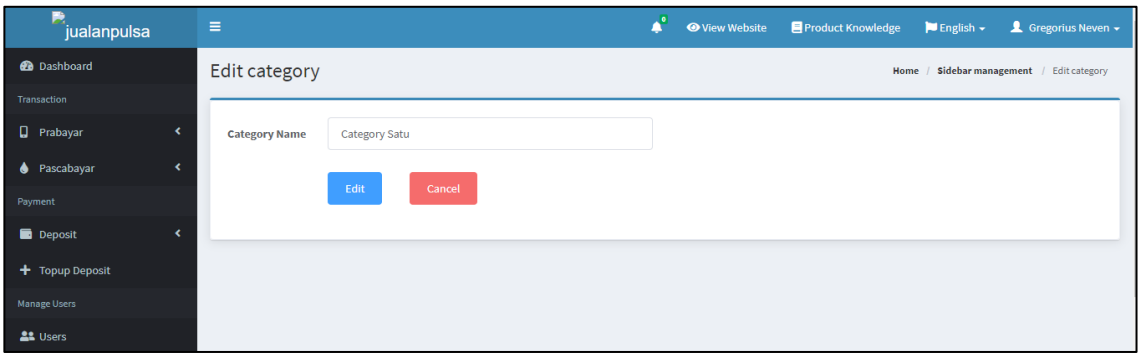
Gambar 3.27 Implementasi Halaman *Create New Menu*

Fitur Menu *Sidebar Custom*

Pada halaman *create new menu* pengguna dapat menambahkan data menu baru pada kategori yang telah ditentukan. Nama kategori menu yang akan ditambahkan ini dapat dilihat pada kolom *input* berlabel "*Category Name*". Untuk menambahkan data menu baru, pengguna dapat mengisi nama menu dan URL *redirect* menu tersebut pada kolom – kolom *input* yang telah disediakan. Setelah itu pengguna dapat menekan tombol "*Create*" untuk menambahkan data menu. Tombol "*Cancel*" digunakan untuk kembali ke halaman utama. Jika pengguna memasukkan nama menu dan URL dengan *format* yang salah, maka pesan kesalahan akan diperlihatkan, dan proses penambahan data menu akan gagal.

A.4 Implementasi Halaman Edit Category Fitur Menu Sidebar Custom

Gambar 3.28 merupakan implementasi halaman *edit category* fitur menu *sidebar custom*.

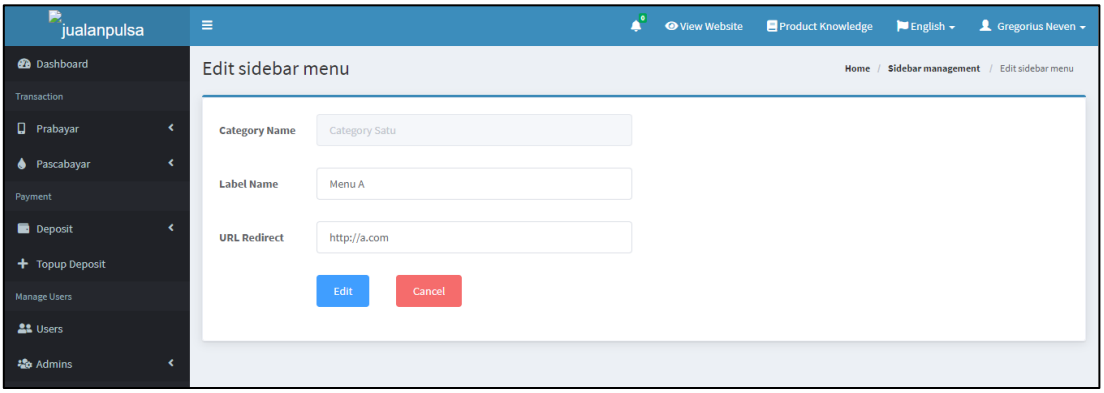


Gambar 3.28 Implementasi Halaman *Edit Category* Fitur Menu *Sidebar Custom*

Pada halaman *edit category* pengguna dapat melakukan perubahan data kategori menu. Pengguna dapat merubah nama kategori pada kolom *input* yang telah disediakan dan menekan tombol "*Edit*" untuk merubah data kategori. Tombol "*Cancel*" digunakan untuk kembali ke halaman utama. Jika pengguna memasukkan nama kategori dengan *format* yang salah, maka pesan kesalahan akan diperlihatkan, dan proses perubahan data kategori akan gagal.

A.5 Implementasi Halaman Edit Menu Fitur Menu Sidebar Custom

Gambar 3.29 merupakan implementasi halaman *edit menu* fitur menu *sidebar custom*.

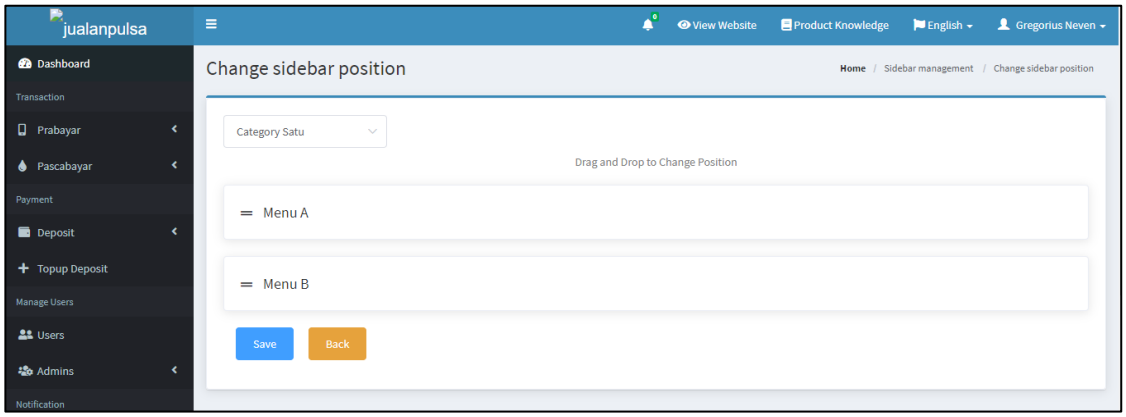


Gambar 3.29 Implementasi Halaman *Edit Menu* Fitur Menu *Sidebar Custom*

Pada halaman *edit menu* pengguna dapat melakukan perubahan data menu yang telah dipilih. Untuk melakukan perubahan data menu, pengguna dapat merubah nama menu dan URL *redirect* menu tersebut pada kolom – kolom *input* yang telah disediakan. Setelah itu pengguna dapat menekan tombol "*Edit*" untuk melakukan perubaan data menu. Tombol "*Cancel*" digunakan untuk kembali ke halaman utama. Jika pengguna memasukkan nama menu dan URL dengan *format* yang salah, maka pesan kesalahan akan diperlihatkan, dan proses perubahan data menu akan gagal.

A.6 Implementasi Halaman Change Position Fitur Menu Sidebar Custom

Gambar 3.30 merupakan implementasi halaman *change position* fitur menu *sidebar custom*.



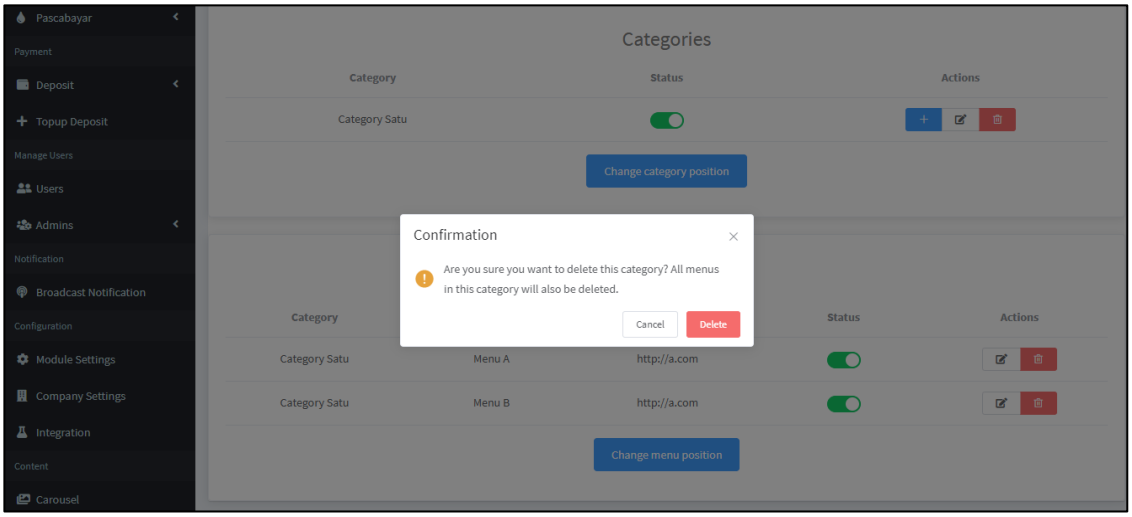
Gambar 3.30 Implementasi Halaman *Change Position* Fitur Menu *Sidebar Custom*

Pada halaman *change position*, pengguna dapat melakukan perubahan posisi urutan tampil kategori atau menu yang sudah dibuat. Untuk melakukan perubahan

tersebut, pengguna dapat melakukan *drag and drop* pada komponen kategori atau menu ke arah atas atau bawah sesuai keinginan pengguna dalam meletakan urutan kategori atau menu tersebut. Tombol "*Save*" digunakan untuk menyimpan perubahan urutan tampil kategori atau menu. Tombol "*Back*" digunakan untuk kembali ke halaman utama.

A.7 Implementasi Dialog Konfirmasi Penghapusan Menu Sidebar Custom

Gambar 3.31 merupakan implementasi *dialog* konfirmasi penghapusan kategori atau menu fitur menu *sidebar custom*.



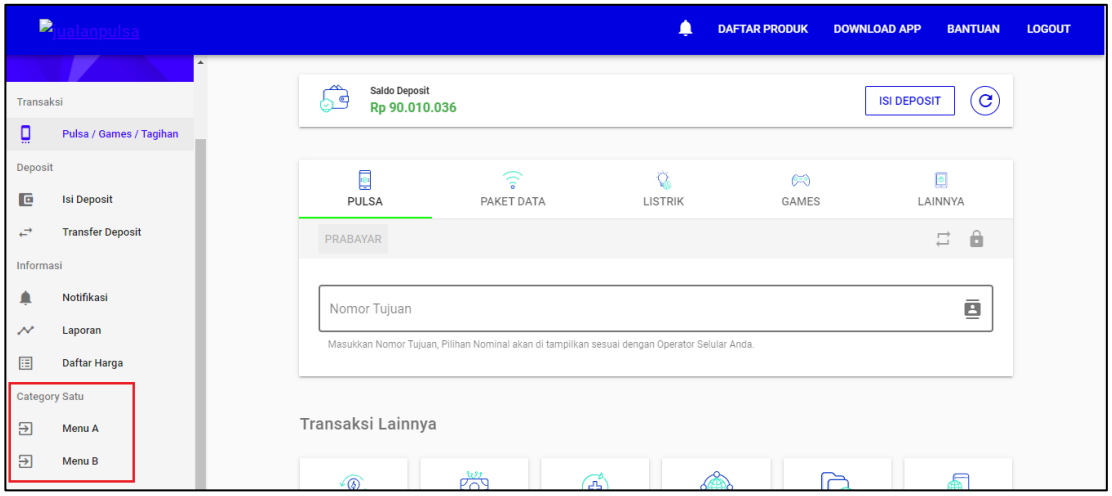
Gambar 3.31 Implementasi *Dialog* Konfirmasi Penghapusan Fitur Menu *Sidebar Custom*

Jika pengguna menekan tombol untuk menghapus kategori atau menu yang telah dibuat, kategori atau menu tersebut tidak akan langsung terhapus. Sebelum sistem melakukan penghapusan, halaman akan menampilkan *dialog* konfirmasi penghapusan.

Jika pengguna menghapus kategori, maka seluruh menu yang ada pada kategori tersebut akan ikut terhapus, namun jika pengguna menghapus sebuah menu, maka hanya menu tersebut yang akan hilang. Tombol "*Delete*" digunakan untuk mengkonfirmasi penghapusan. Tombol "*Cancel*" digunakan untuk membatalkan penghapusan dan kembali ke halaman utama.

A.8 Implementasi Penampilan Menu Sidebar Custom

Gambar 3.32 merupakan implementasi penampilan menu *sidebar custom*.



Gambar 3.32 Implementasi Penampilan Menu *Sidebar Custom*

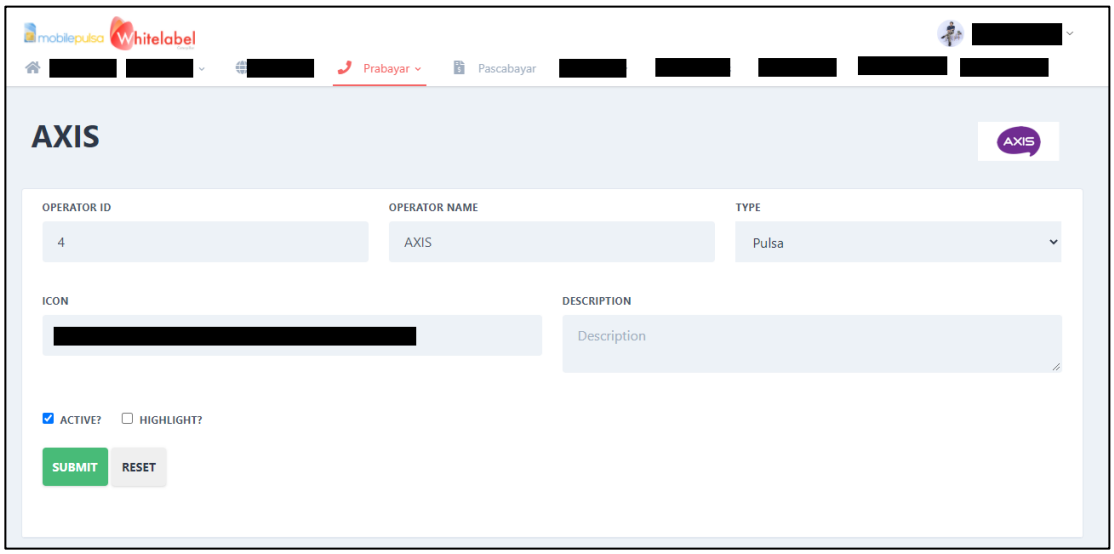
Kategori dan menu – menu yang telah dibuat akan tampil pada bagian *sidebar* halaman situs Whitelabel milik pengguna (*tenant*). Menu – menu ini dapat ditekan oleh pelanggan situs Whitelabel milik *tenant* dan akan mengarahkan pelanggan kepada URL yang telah ditentukan oleh *tenant* pada saat melakukan penambahan atau perubahan data menu.

B. Implementasi Fitur Perubahan Otomatis Database Tenant

Implementasi fitur perubahan otomatis *database tenant* direpresentasikan menggunakan tampilan situs *superadmin* Whitelabel untuk melakukan perubahan data, dan menggunakan tampilan situs *admin* Whitelabel untuk melihat data yang telah diubah.

B.1 Implementasi Halaman Utama Fitur Perubahan Otomatis Database

Gambar 3.33 merupakan salah satu contoh halaman utama yang dapat diakses untuk melakukan perubahan data operator produk fitur perubahan otomatis *database*.

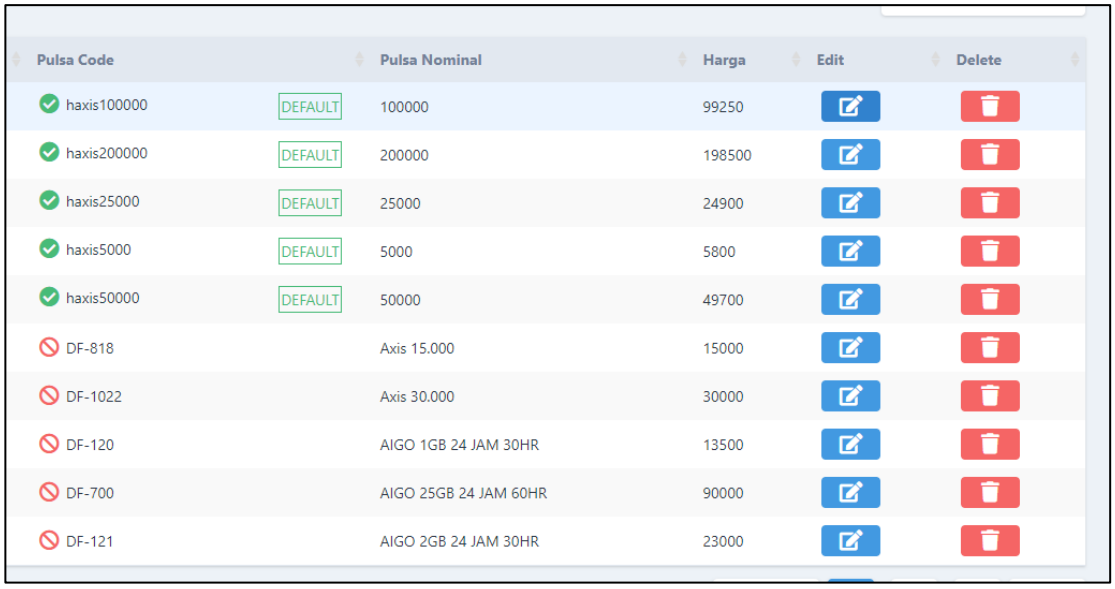


Gambar 3.33 Implementasi Halaman Utama Perubahan Operator Fitur Perubahan Otomatis *Database Tenant*

Pada halaman ini, pengguna (*superadmin* Whitelabel) dapat melakukan perubahan data operator produk dengan mengisi kolom *input* yang sudah disediakan. Kolom *input* dengan label "*Operator Name*" dapat diisi dengan nama operator. Label

"*type*" dapat diisi dengan tipe operator (pulsa, game, dsb). Label "*icon*" dapat diisi dengan URL *icon* yang digunakan. Label "*description*" dapat diisi dengan deskripsi operator. Tombol "*Submit*" digunakan untuk melakukan perubahan. Tombol "*Reset*" digunakan untuk membatalkan perubahan. Jika pengguna melakukan perubahan terhadap data operator ini, maka secara otomatis perubahan tersebut akan disimpan dan disesuaikan ke seluruh *database tenant* Whitelabel yang aktif.

Gambar 3.34 merupakan salah satu contoh halaman utama yang dapat diakses untuk melakukan perubahan data produk fitur perubahan otomatis *database*.



Pulsa Code		Pulsa Nominal	Harga	Edit	Delete
haxis100000	DEFAULT	100000	99250		
haxis200000	DEFAULT	200000	198500		
haxis25000	DEFAULT	25000	24900		
haxis5000	DEFAULT	5000	5800		
haxis50000	DEFAULT	50000	49700		
DF-818		Axis 15.000	15000		
DF-1022		Axis 30.000	30000		
DF-120		AIGO 1GB 24 JAM 30HR	13500		
DF-700		AIGO 25GB 24 JAM 60HR	90000		
DF-121		AIGO 2GB 24 JAM 30HR	23000		

Gambar 3.34 Implementasi Halaman Utama Perubahan Produk

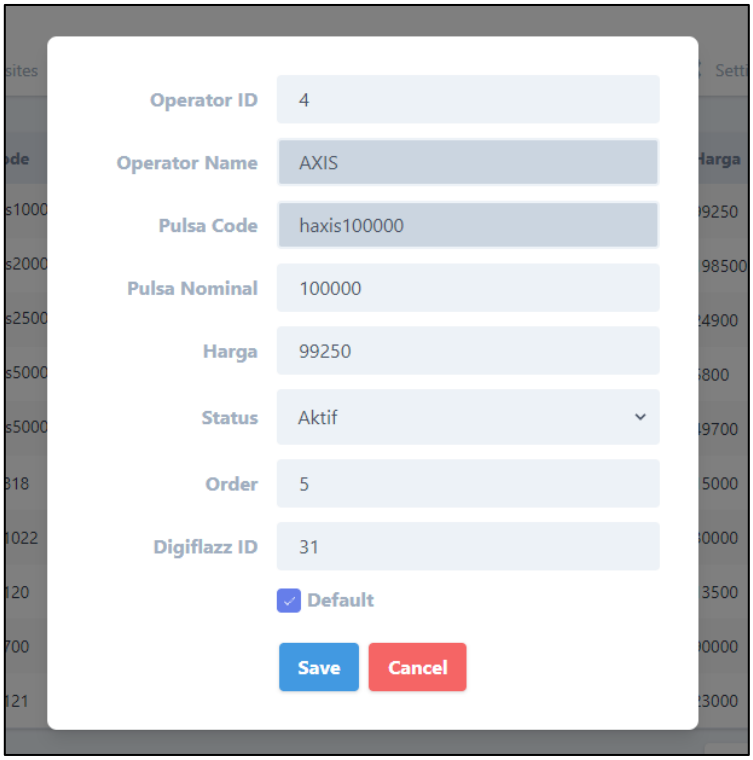
Fitur Perubahan Otomatis *Database Tenant*

Pada halaman ini, pengguna (*superadmin* Whitelabel) dapat melihat seluruh produk sesuai dengan operator yang telah dipilih yang ditampilkan dalam bentuk tabel.

Pada tabel, kolom “Pulsa *Code*” merupakan kode produk, kolom “Pulsa Nominal” merupakan nominal produk, kolom “Harga” merupakan harga satuan dari setiap produk yang ada. Pada setiap baris juga terdapat tombol untuk melakukan pengubahan dan penghapusan data produk. Tombol berwarna biru digunakan untuk melakukan perubahan data produk. Tombol berwarna merah digunakan untuk menghapus data produk.

B.2 Implementasi Halaman Edit Produk Fitur Perubahan Otomatis Database

Gambar 3.35 merupakan salah satu contoh halaman untuk melakukan perubahan data produk fitur perubahan otomatis *database*.



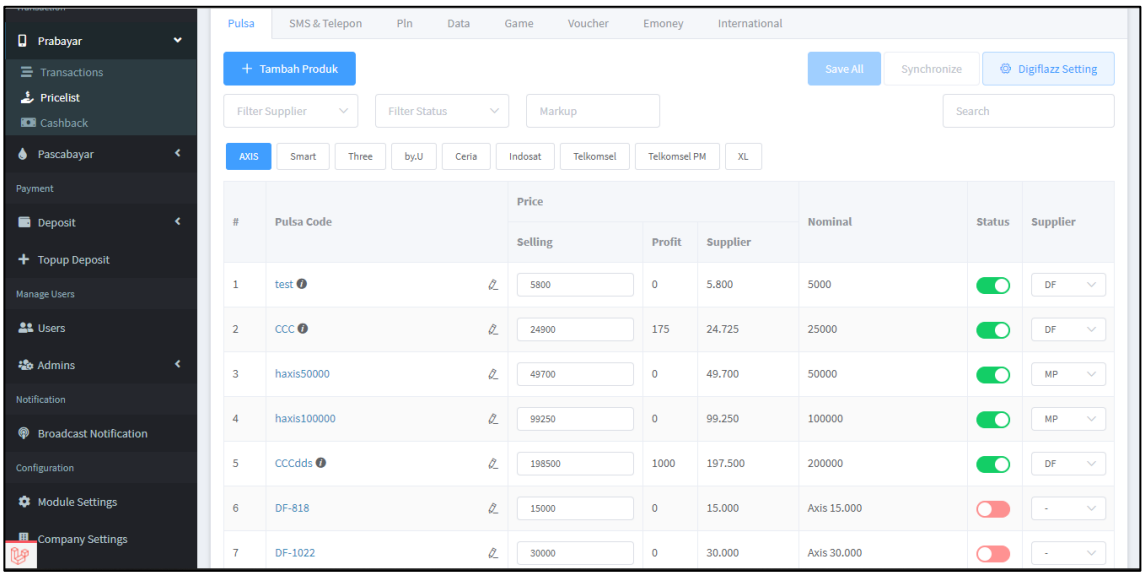
Gambar 3.35 Implementasi Halaman *Edit* Produk

Fitur Perubahan Otomatis *Database Tenant*

Pada halaman ini, pengguna dapat melakukan perubahan pada produk yang telah dipilih. Pengguna dapat melakukan perubahan data nominal pulsa, harga produk, status produk, urutan tampil produk, dan ID Digiflazz. Untuk melakukan perubahan tersebut, pengguna dapat mengisi perubahan pada kolom *input* yang telah disediakan sesuai dengan label deskripsi masing – masing. Tombol "*Save*" digunakan untuk menyimpan perubahan. Tombol "*Cancel*" digunakan untuk membatalkan perubahan dna kembali ke halaman utama. Jika pengguna melakukan perubahan terhadap data produk ini, maka secara otomatis perubahan tersebut akan disimpan dan disesuaikan ke seluruh *database tenant* Whitelabel yang aktif.

B.3 Implementasi Halaman Daftar Produk Perubahan Otomatis Database

Gambar 3.36 merupakan contoh halaman daftar produk yang dapat diakses oleh masing – masing *tenant* Whitelabel pada halaman *admin* situs Whitelabel.



Gambar 3.36 Implementasi Halaman Daftar Produk

Fitur Perubahan Otomatis *Database Tenant*

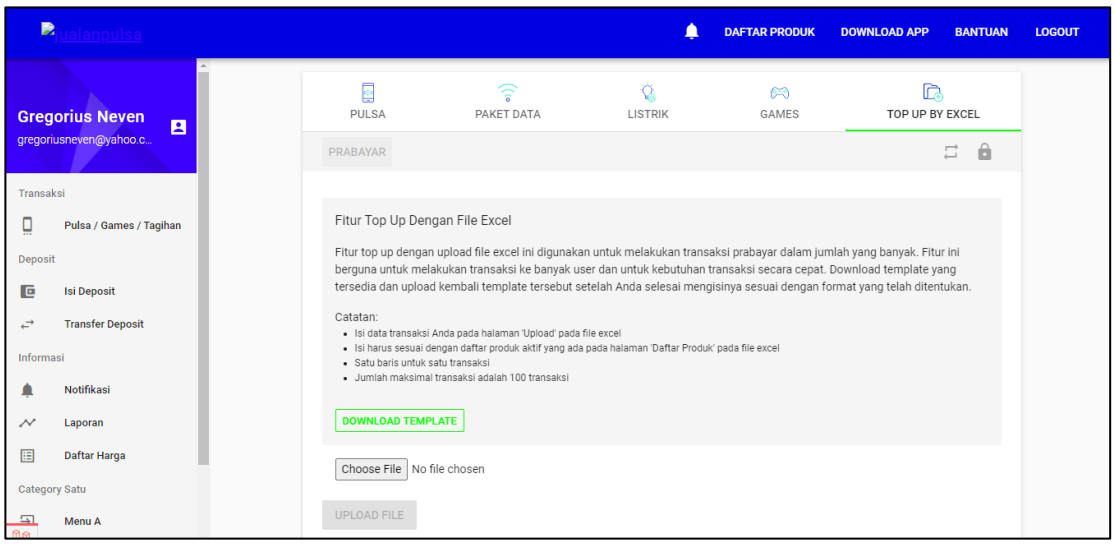
Pada halaman ini, pengguna (*tenant* Whitelabel) dapat melihat seluruh produk yang ada. Pengguna juga dapat melihat seluruh detail produk dengan lengkap, seperti kode produk, nominal produk, dan harga produk. Segala perubahan yang dilakukan pada halaman *superadmin* Whitelabel akan dapat langsung dilihat oleh setiap *tenant* Whitelabel yang aktif melalui halaman ini. Dengan adanya fitur ini, segala perubahan yang ada dapat dengan cepat diimplementasikan kepada semua *tenant* Whitelabel sehingga dapat mengurangi kerugian transaksi yang dapat terjadi, jika ada perubahan data dalam *database* setiap tenant.

C. Implementasi Fitur Top Up by Excel

Implementasi fitur menu *Top Up by* Excel dibagi menjadi beberapa bagian yang terdiri dari halaman awal, tabel konformasi transaksi, dan berbagai pesan validasi transaksi.

C.1 Implementasi Halaman Awal Fitur Top Up By Excel

Gambar 3.37 merupakan implementasi halaman awal fitur *Top Up by* Excel.



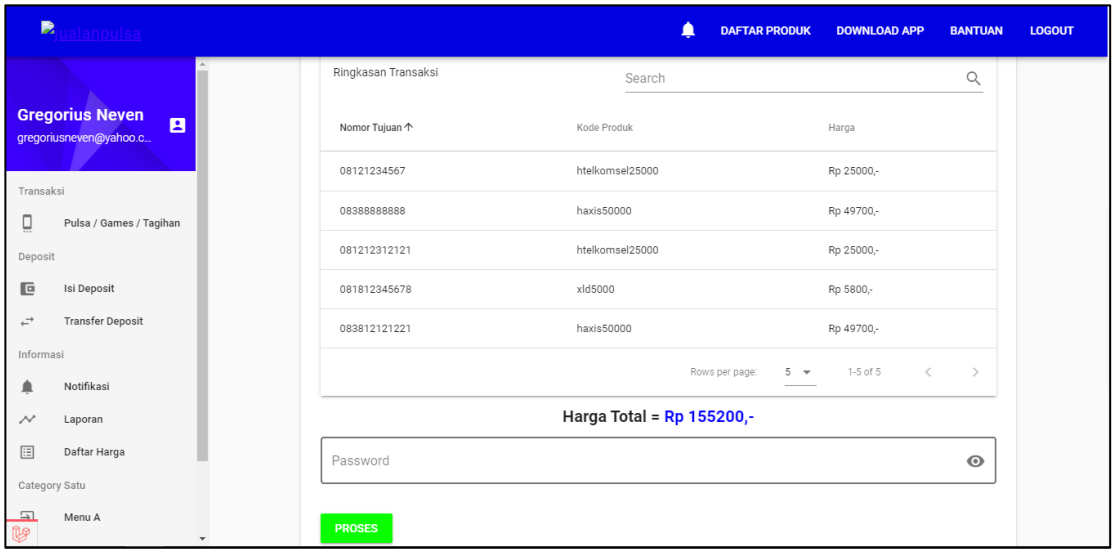
Gambar 3.37 Implementasi Halaman Awal Fitur *Top Up by* Excel

Halaman awal fitur *Top Up by* Excel menampilkan penjelasan singkat terkait fungsi dan cara kerja fitur ini. Selain itu, juga ditampilkan catatan – catatan yang berupa himbauan, aturan, dan prosedur penggunaan fitur ini secara singkat untuk membimbing pengguna (pelanggan situs Whitelabel) untuk bertransaksi dengan menggunakan *file* Excel. Tombol "*Download Template*" berfungsi untuk mengunduh *file template* Excel yang dapat digunakan pengguna untuk mengisi permintaan transaksi yang diinginkan.

Tombol "*Choose File*" berfungsi untuk memilih *file* yang akan diunggah oleh pengguna. *File template* Excel yang telah diunduh pengguna idealnya dapat digunakan untuk diunggah kembali setelah pengguna selesai mengisi *file* tersebut dengan permintaan transaksi. Tombol "*Upload File*" digunakan untuk mengunggah *file* yang telah dipilih oleh pengguna.

C.2 Implementasi Tabel Konfirmasi Transaksi Fitur Top Up By Excel

Gambar 3.38 merupakan implementasi tabel konfirmasi transaksi fitur *Top Up by* Excel.



Gambar 3.38 Implementasi Tabel Konfirmasi Transaksi Fitur *Top Up by* Excel

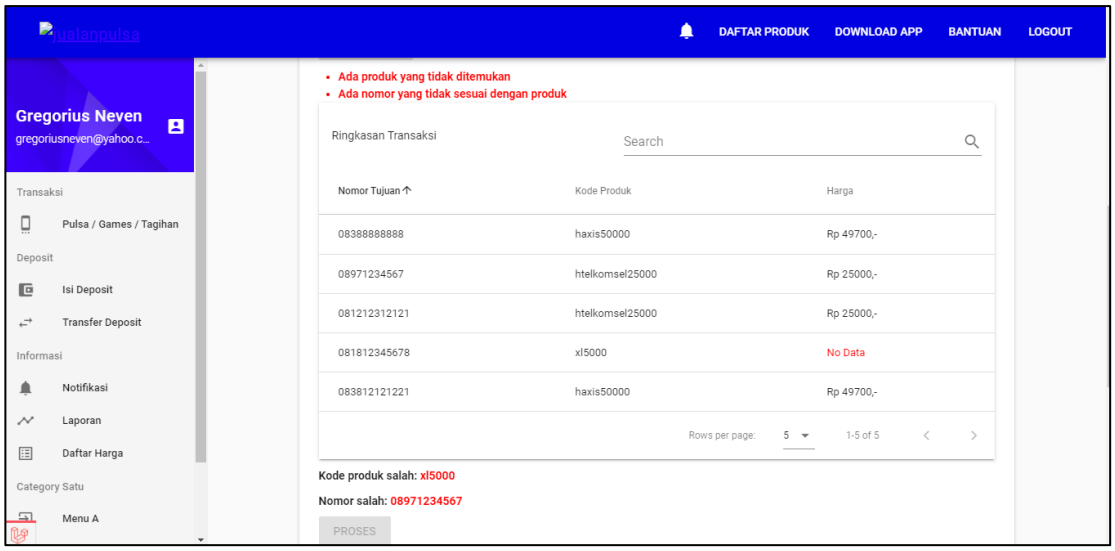
Tabel konfirmasi transaksi akan muncul setelah pengguna melakukan pengunggahan *file* Excel yang telah diisi dengan permintaan transaksi. Tabel konfirmasi transaksi terdiri dari tiga buah kolom, yaitu kolom "Nomor Tujuan" untuk yang memperlihatkan nomor tujuan transaksi, kolom "Kode Produk" yang

memperlihatkan kode produk yang diminta pengguna, kolom “Harga” yang menunjukkan harga satuan produk. Satu baris pada tabel konfirmasi transaksi ini merupakan satu buah transaksi. Tabel ini mendukung pencarian transaksi dengan kata kunci, pengurutan data menurut kolom, dan juga dapat menampilkan banyak data yang dapat dipisah berdasarkan halaman – halaman pada tabel.

Di bagian bawah tabel konfirmasi transaksi terdapat label “Harga Total” yang menampilkan harga total transaksi yang diminta pengguna. Jika pengguna telah selesai melakukan pengecekan dan konfirmasi permintaan transaksi, pengguna dapat mengisi *password* pengguna pada kolom *input password* yang telah disediakan. Setelah itu, pengguna dapat menekan tombol “Proses” untuk memproses permintaan transaksi.

C.3 Implementasi Validasi Permintaan Transaksi Fitur Top Up By Excel

Gambar 3.39 merupakan implementasi validasi permintaan transaksi fitur *Top Up by* Excel.

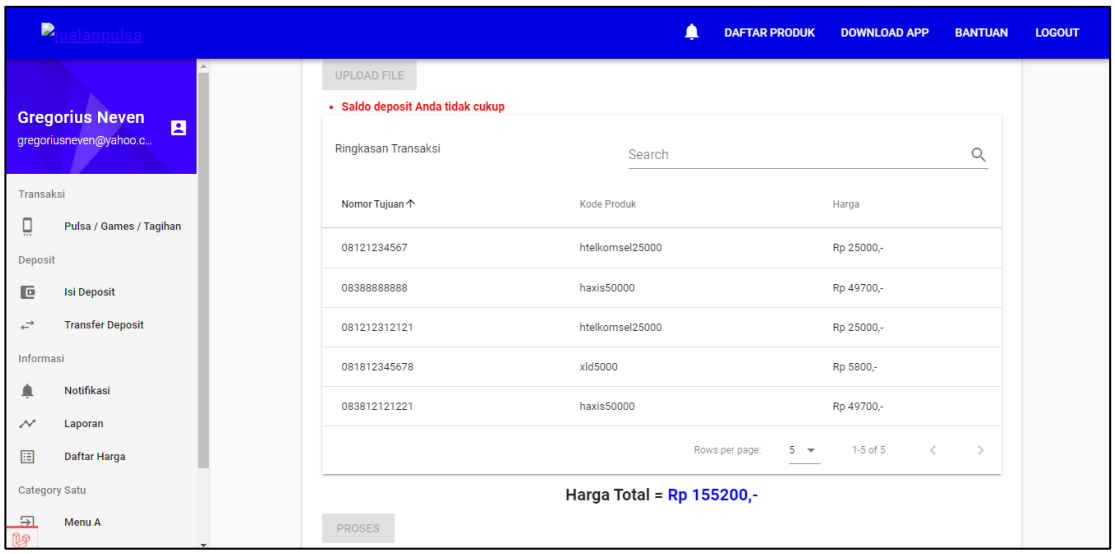


Gambar 3.39 Implementasi Validasi Permintaan Transaksi Fitur *Top Up by* Excel

Jika terdapat kesalahan dalam pengisian permintaan transaksi pada *file* Excel, maka pesan validasi akan ditampilkan. Kolom *input password* tidak akan muncul dan tombol "Proses" tidak akan dapat ditekan jika pesan kesalahan muncul. Jika pengguna salah memasukkan kode produk, maka pesan kesalahan "produk tidak ditemukan" akan ditampilkan. Kode produk yang salah pun akan ditampilkan untuk memberi informasi kepada pengguna terkait produk yang tidak ditemukan tersebut. Jika pengguna salah memasukkan nomor tujuan, maka pesan kesalahan "nomor tujuan tidak sesuai" akan ditampilkan. Nomor tujuan yang tidak sesuai tersebut juga akan ditampilkan kepada pengguna. Kesalahan nomor tujuan yang tidak sesuai dapat terjadi jika pengguna memasukkan nomor tujuan dengan awalan atau *prefix* nomor yang tidak sesuai dengan produk yang diminta. Jika mendapatkan pesan kesalahan ini, pengguna dapat memperbaiki *file* Excel, dan mengunggah kembali *file* Excel tersebut untuk divalidasi kembali.

C.4 Implementasi Validasi Saldo Fitur Top Up By Excel

Gambar 3.40 merupakan implementasi validasi saldo fitur *Top Up by* Excel.



Gambar 3.40 Implementasi Validasi Saldo Fitur *Top Up by* Excel

Jika jumlah saldo *deposit* pengguna lebih kecil dari total harga transaksi yang diminta pengguna, maka pesan kesalahan "saldo deposit tidak cukup" akan ditampilkan. Kolom *input password* dan tombol "Proses" juga tidak dapat digunakan jika pesan kesalahan ini muncul. Jika mendapatkan pesan kesalahan ini, pengguna dapat menambah saldo *deposit* terlebih dahulu melalui menu yang sudah disediakan, ataupun memperbaiki *file* Excel untuk mengurangi jumlah permintaan transaksi, sebelum mengunggah kembali *file* Excel tersebut untuk divalidasi kembali.

3.3.5 Kendala yang Ditemukan

Selama melakukan pengerjaan tugas kerja magang, terdapat beberapa kendala yang dihadapi. Kendala – kendala tersebut adalah sebagai berikut.

1. Pengembangan fitur yang sebagian dilakukan menggunakan *framework* Vue.js yang belum pernah dipelajari dan digunakan sebelumnya.
2. Pengembangan fitur yang sebagian dilakukan menggunakan berbagai jenis *libraries* seperti Vuetify, Element, dan tailwindcss yang belum pernah dipelajari dan digunakan sebelumnya.
3. Struktur proyek Whitelabel yang besar dan berbeda dengan struktur proyek pada umumnya.
4. Struktur *database* proyek Whitelabel yang besar berbeda dengan struktur proyek pada umumnya.

3.3.6 Solusi atas Kendala yang Ditemukan

Solusi yang dilakukan untuk mengatasi kendala – kendala yang dihadapi selama pengerjaan tugas kerja magang adalah sebagai berikut.

1. Mempelajari *framework* Vue.js dengan melakukan studi dokumentasi, melihat contoh implementasi kode yang sudah ada, dan berdiskusi dengan rekan dan mentor kerja magang.
2. Mempelajari berbagai jenis *libraries* yang digunakan pada proyek dengan melakukan studi dokumentasi, melihat contoh implementasi kode yang sudah ada, dan berdiskusi dengan rekan dan mentor kerja magang.

3. Mempelajari struktur proyek Whitelabel dengan melakukan studi dokumentasi serta berdiskusi dengan rekan dan mentor kerja magang.
4. Mempelajari struktur *database* proyek Whitelabel dengan berdiskusi dengan rekan dan mentor kerja magang.