



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

PELAKSANAAN KERJA MAGANG

3.1. Kedudukan dan Koordinasi

Kedudukan peserta magang di PT Indobest Artha Kreasi selama melaksanakan kerja magang adalah sebagai *software engineer* di bawah divisi Pengembangan Produk yang bertugas untuk mengembangkan layanan produk API, produk prabayar, dan produk pascabayar. Oleh karena divisi Pengembangan Produk ini sangat fokus dalam pengembangan API, divisi ini disebut sebagai divisi Pengembangan Produk API. Selama melaksanakan kerja magang, peserta magang bekerja di bawah bimbingan Eddy Christiandy selaku *Chief Technical Officer* dari Mobilepulsa dan Martha Saphira, selaku *Project Manager* divisi Pengembangan Produk API.

Selama melaksanakan kerja magang, anggota divisi Pengembangan Produk API melakukan koordinasi dengan cara melakukan komunikasi, baik melalui tatap muka di ruang kerja maupun melalui aplikasi pesan singkat, seperti Telegram. Selain itu, divisi Pengembangan Produk API juga melakukan rapat untuk melakukan koordinasi. Setidaknya satu kali dalam seminggu, divisi Pengembangan Produk API akan melakukan rapat yang membahas perkembangan proyek dan kendala yang dirasakan selama pengembangan proyek, guna memastikan setiap proyek memiliki perkembangan dan membantu anggota tim dalam mengembangkan proyek yang sedang dikembangkan. Dalam melakukan kontrol versi proyek, peserta magang dan tim menggunakan Bitbucket. Dalam melakukan koordinasi dan pembagian tugas, peserta magang dan tim menggunakan aplikasi Trello.

3.2. Tugas yang Dilakukan

Dalam melakukan kerja magang, peserta magang memiliki tugas-tugas yang harus dilaksanakan. Tugas-tugas tersebut adalah menambahkan fitur-fitur ke dalam panel admin Mobilepulsa, mengembangkan fitur pada proyek *website developer* API Mobilepulsa, dan menambahkan produk baru pada layanan pascabayar Mobilepulsa. Oleh karena ketiga tugas tersebut tidak saling terkait dan tugas yang paling banyak memakan waktu ialah tugas untuk menambahkan fitur-fitur ke dalam panel admin Mobilepulsa, maka tugas peserta magang yang akan dilaporkan adalah penambahan fitur-fitur pada panel admin Mobilepulsa. Fitur-fitur yang ditambahkan adalah fitur *blacklist product*, *broadcast product*, *secondary email*, dan *update supplier* PLN.

Dalam menjalankan tugasnya, peserta magang menggunakan bahasa pemrograman PHP yang dibantu dengan *framework* Laravel, HTML, CSS, dan Javascript. Selain itu, peserta magang juga menggunakan mySQL untuk mengakses data dalam *database* terkait. Setiap fitur setidaknya memiliki satu dari beberapa halaman *web* dan satu *controller* yang menunjang jalannya fitur ini dengan menggunakan *framework* Laravel. Beberapa halaman *web* tersebut adalah halaman *index*, *create*, dan *edit*. Halaman *index* berfungsi untuk menampilkan data-data umum terkait fitur. Halaman *create* berfungsi untuk menampilkan *form* yang ditujukan untuk memasukkan data baru ke dalam *database*. Halaman *edit* berfungsi untuk menampilkan *form* yang ditujukan untuk memodifikasi data yang sudah ada di dalam *database*. Selama pengembangan dilakukan, peserta magang menggunakan data *dummy* dari *database* yang sudah disediakan oleh perusahaan.

Tugas-tugas yang dilakukan oleh peserta magang selama kerja magang dapat dilihat pada tabel 3.1.

Tabel 3.1 Realisasi Kerja Magang

Minggu	Uraian Kegiatan
1	Instalasi dan persiapan pengerjaan proyek di Mobilepulsa
2	Pembuatan fitur <i>blacklist</i> dan <i>update supplier</i> PLN pada proyek panel admin Mobilepulsa
3	Pembuatan fitur <i>blacklist</i> dan <i>update supplier</i> PLN pada proyek panel admin Mobilepulsa
4	Pembuatan fitur <i>broadcast product</i> pada proyek panel admin Mobilepulsa
5	Pembuatan fitur <i>broadcast product</i> pada proyek panel admin Mobilepulsa
6	Pembuatan fitur <i>secondary email</i> pada proyek panel admin Mobilepulsa
7	Pembuatan fitur <i>secondary email</i> pada proyek panel admin Mobilepulsa
8	Memperbarui fitur <i>secondary email</i> pada proyek <i>developer API</i> Mobilepulsa
9	Memperbarui fitur <i>pricelist</i> pada proyek <i>developer API</i> Mobilepulsa
10	Memperbarui fitur <i>pricelist</i> pada proyek <i>developer API</i> Mobilepulsa
11	Memperbarui fitur <i>pricelist</i> pada proyek <i>developer API</i> Mobilepulsa
12	Integrasi produk baru pada layanan <i>postpaid</i> Mobilepulsa
13	Integrasi produk baru pada layanan <i>postpaid</i> Mobilepulsa
14	Integrasi produk baru pada layanan <i>postpaid</i> Mobilepulsa
15	Integrasi produk baru pada layanan <i>postpaid</i> Mobilepulsa

3.3. Uraian Pelaksanaan Kerja Magang

Pelaksanaan kerja magang yang dilakukan oleh peserta magang dapat diuraikan kembali menjadi tiga bagian, yaitu proses pelaksanaan, kendala yang ditemukan, dan solusi atas kendala yang ditemukan.

3.3.1. Proses Pelaksanaan

Pelaksanaan kerja magang membutuhkan perangkat keras dan perangkat lunak agar dapat dilaksanakan dengan baik. Perangkat lunak yang digunakan oleh peserta magang selama pelaksanaan magang dilaksanakan, antara lain.

- a. *Framework* Laravel 5.6
- b. Wampserver 3.1.9 (64 bit) yang memiliki Apache v2.4.37, PHP v7.1.3, MySQL v5.7.24, MariaDB v10.3.12, dan PhpMyAdmin v4.8.4
- c. PhpStorm v2019.2.3
- d. GitKraken v6.3.0
- e. Brave Browser v0.69.135
- f. Sistem Operasi Windows 10 Home (64 bit)

Perangkat keras yang digunakan selama pelaksanaan magang adalah ASUS A456UF dengan spesifikasi perangkat keras sebagai berikut.

- a. Processor Intel Core™ i5-6200U CPU @ 2.30GHz
- b. RAM 8GB DDR3
- c. Hard Disk Drive 1TB
- d. NVIDIA Graphic Card GeForce 930M

Proses pelaksanaan magang terbagi menjadi dua bagian, yaitu perancangan sistem dan implementasi.

A. Perancangan Sistem

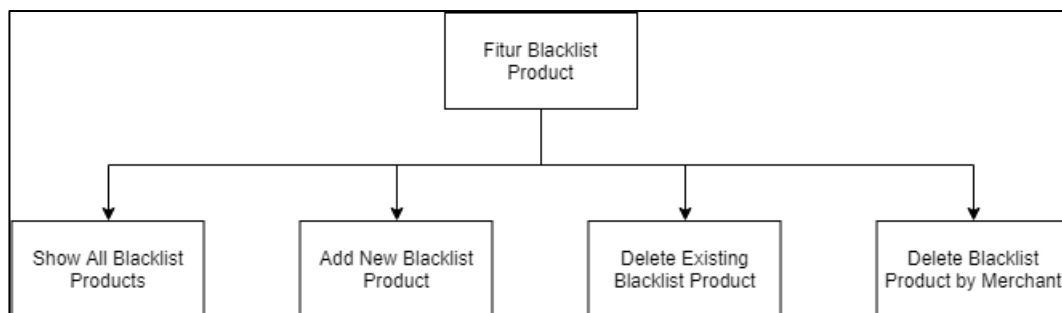
Pada bagian ini, perancangan dilakukan dengan pembuatan *flowchart* dan perancangan *user interface*.

A.1. Flowchart

Flowchart adalah suatu representasi grafik dari suatu logika pemrograman untuk menyelesaikan suatu permasalahan tertentu yang menggambarkan aliran proses dari penyelesaian permasalahan tersebut (Robertson, 2007).

A.1.1. Flowchart Fitur Blacklist Product

Gambar 3.1 menunjukkan hal-hal yang dapat dilakukan dan/atau dipilih oleh *user* ketika fitur *blacklist product* digunakan.

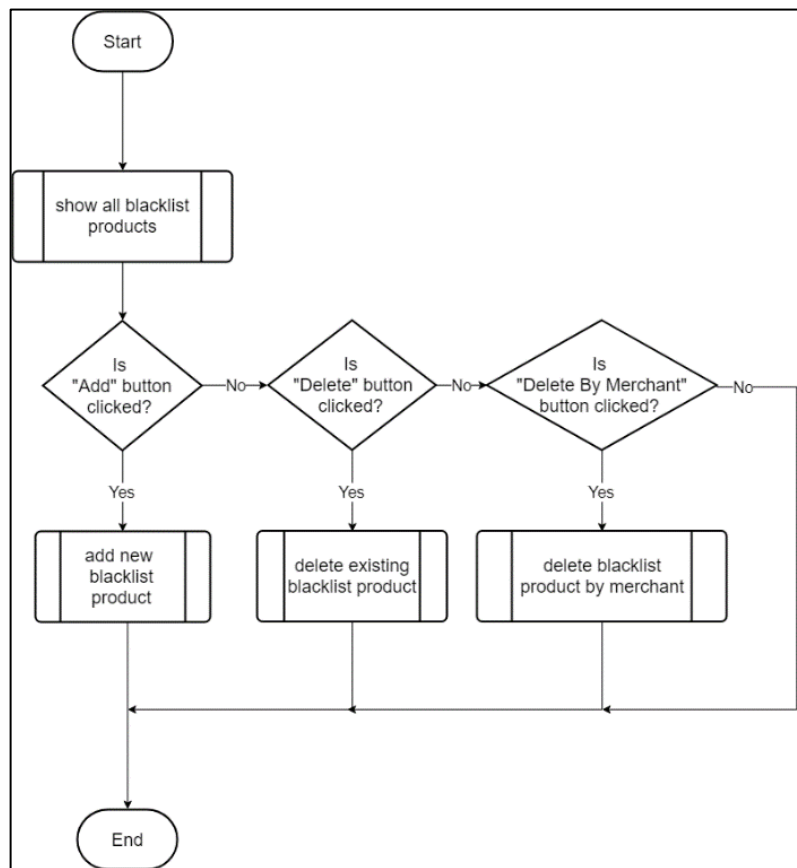


Gambar 3.1 Fitur *Blacklist Product*

Berdasarkan gambar 3.1, *user* dapat melihat semua produk yang didaftarhitamkan pada *client*. Selain itu, *user* juga dapat menambahkan produk yang hendak didaftarhitamkan pada *client* dan menghapus produk yang sudah didaftarhitamkan pada *client*, baik dengan satu per satu maupun beberapa sekaligus berdasarkan *client* yang dipilih.

Flowchart yang menggambarkan fitur *blacklist product* dalam gambaran besar dapat dilihat pada gambar 3.2. Berdasarkan gambar 3.2, semua produk yang sudah didaftarhitamkan sebelumnya akan ditampilkan pada panel admin saat *user* pertama kali mengunjungi halaman fitur ini. *User* dapat menambahkan produk yang ingin didaftarhitamkan pada *client* setelah menekan tombol “Add”.

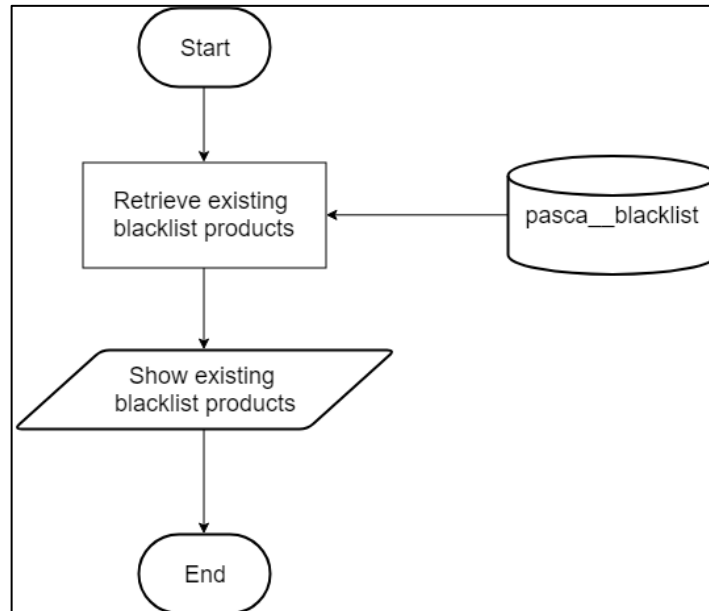
Selain itu, *user* juga dapat menghapus produk yang sudah didaftarhitamkan secara satu per satu setelah menekan tombol “Delete” dan menghapus beberapa produk yang sudah didaftarhitamkan sekaligus berdasarkan *client* yang dipilih setelah menekan tombol “Delete by Merchant”.



Gambar 3.2 *Flowchart* Fitur *Blacklist Product*

1) **Flowchart Show All Blacklist Products**

Sebelum data produk yang sudah didaftarhitamkan ditampilkan, data produk diambil terlebih dahulu dari tabel `pasca_blacklist`. Setelah data diambil, data akan ditampilkan dalam bentuk tabel pada halaman utama fitur *blacklist product*. Gambar 3.3 menggambarkan *flowchart* untuk menampilkan semua produk yang sudah didaftarhitamkan sebelumnya.



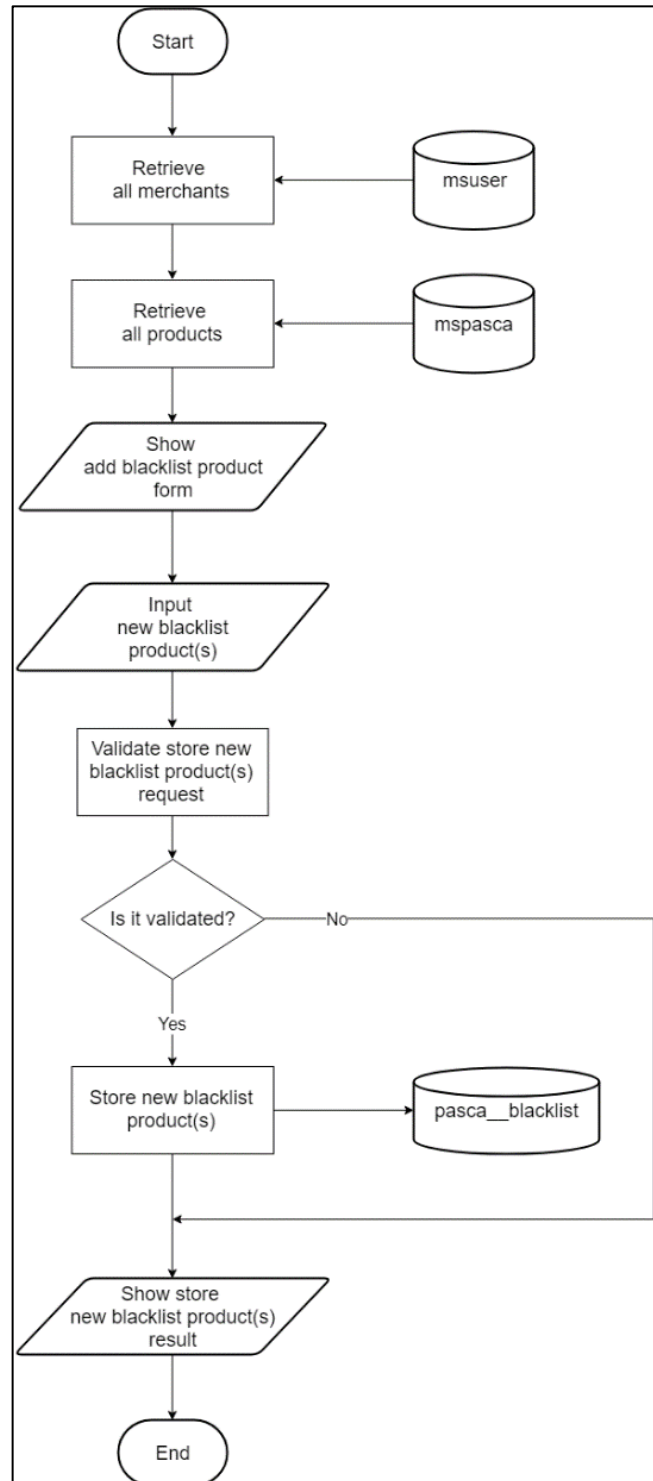
Gambar 3.3 *Flowchart Show All Blacklist Products*

2) **Flowchart Add Blacklist Product**

Setelah tombol “Add” pada halaman utama fitur *blacklist product* ditekan, *user* akan diarahkan menuju halaman yang memuat sebuah *form* untuk menambahkan produk yang akan didaftarhitamkan. Sebelum menampilkan *form*, panel admin akan mengakses *database* untuk mengambil data *client* dan data produk yang tersedia dari tabel *msuser* dan *mspasca*. Setelah *form* selesai diisi, masukan *user* akan divalidasi terlebih dahulu.

Jika validasi gagal, panel admin akan langsung mengarahkan *user* menuju halaman utama fitur *blacklist product* dan menunjukkan hasil bahwa penambahan produk yang didaftarhitamkan gagal. Sebaliknya, jika validasi berhasil, data yang sudah di-input oleh *user* akan dimasukkan ke dalam tabel *pasca_blacklist* dan *user* akan diarahkan kembali menuju halaman utama fitur *blacklist product* dengan menunjukkan hasil bahwa penambahan produk yang didaftarhitamkan berhasil.

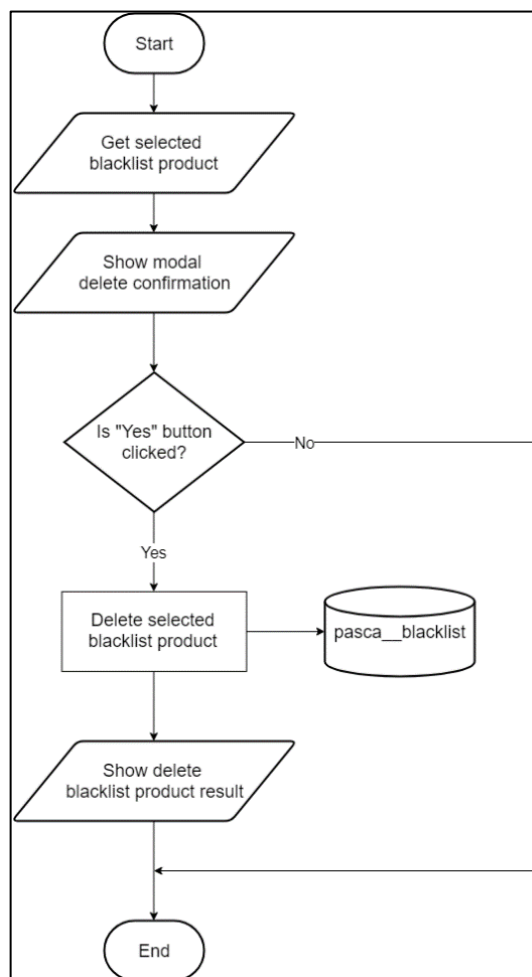
Flowchart untuk menambahkan produk yang didaftarhitamkan ditunjukkan pada gambar 3.4.



Gambar 3.4 *Flowchart Add Blacklist Product*

3) Flowchart Delete Blacklist Product

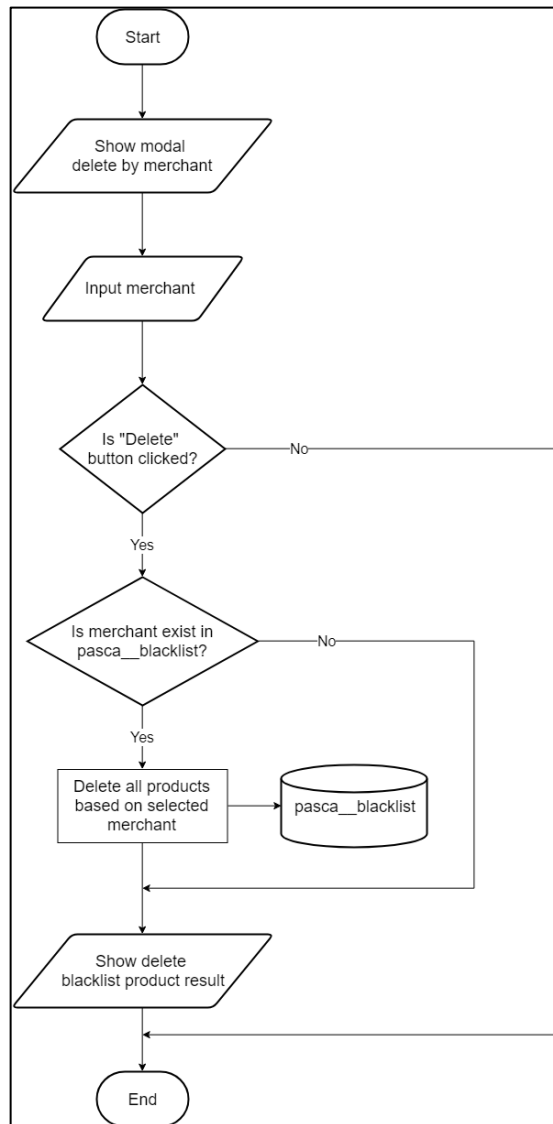
Setelah tombol “Delete” pada halaman utama fitur *blacklist product* ditekan, *modal* yang berisi konfirmasi untuk menghapus produk dan *client* dari daftar *blacklist* akan dimunculkan. Ketika *user* menekan tombol “Yes”, panel admin akan menghapus produk dan *client* tersebut dari daftar *blacklist* di dalam tabel *pasca_blacklist*. Setelah itu, hasil yang menunjukkan bahwa produk dan *client* berhasil dihapus atau tidak dari daftar *blacklist* akan dimunculkan pada halaman utama fitur *blacklist product*. *Flowchart* untuk menghapus produk dan *client* dari daftar *blacklist* dapat dilihat pada gambar 3.5.



Gambar 3.5 *Flowchart Delete Blacklist Product*

4) Flowchart Delete Blacklist Product by Merchant

Gambar 3.6 menunjukkan *flowchart* untuk menghapus produk yang didaftarhitamkan berdasarkan *client*.



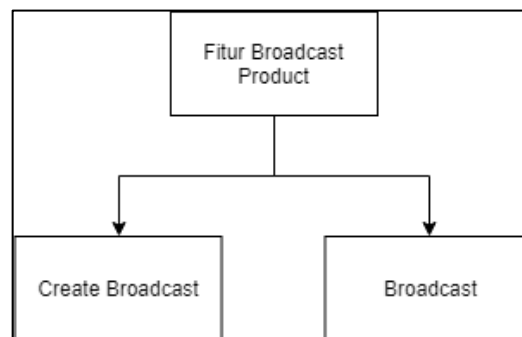
Gambar 3.6 *Flowchart Delete Blacklist Product by Merchant*

Berdasarkan gambar 3.6, fungsi untuk menghapus produk *blacklist* berdasarkan *client* dimulai dengan memunculkan sebuah *modal*. *Modal* tersebut berisi *form* yang ditujukan kepada *user* untuk mengisi nama *client* yang hendak dihapus dari daftar *blacklist*.

Setelah *user* mengisi nama *client* dan menekan tombol “Delete”, semua produk yang didaftarhitamkan pada *client* tersebut akan dihapus secara sekaligus dari daftar *blacklist* dalam tabel *pasca_blacklist*. Setelah semua data dihapus, hasil yang menunjukkan bahwa semua produk dan *client* tersebut berhasil dihapus atau tidak dari daftar *blacklist* akan ditampilkan pada halaman utama fitur *blacklist product*.

A.1.2. Flowchart Fitur Broadcast Product

Gambar 3.7 menunjukkan dua hal yang dapat dilakukan oleh *user* ketika menggunakan fitur *broadcast product*.

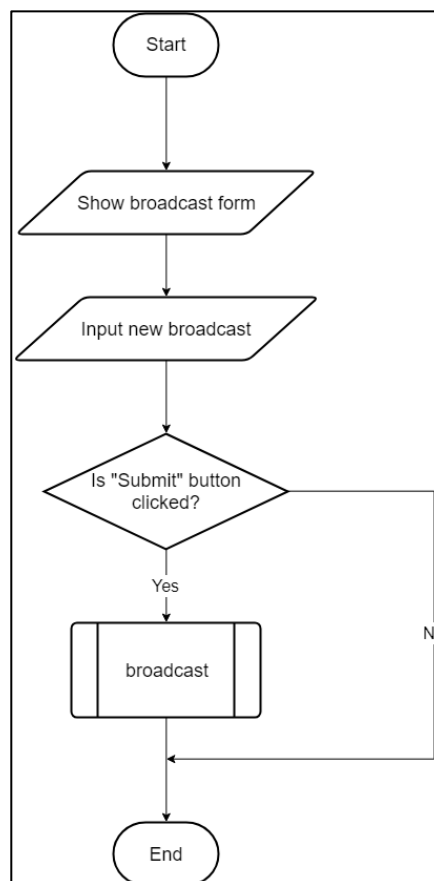


Gambar 3.7 Fitur *Broadcast Product*

Berdasarkan gambar 3.7, terdapat dua hal yang dapat dilakukan oleh *user* ketika menggunakan fitur *broadcast product*, yaitu membuat pesan yang akan disebarakan kepada *client* (create broadcast) dan menyebarkan pesan yang sudah dibuat (broadcast). Pada bagian *create broadcast*, *user* diharuskan untuk memilih *client* yang akan menerima pesan, menuliskan *subject* dari *broadcast* yang akan dikirimkan, dan mengisi pesan dalam *broadcast*. Pada bagian *broadcast*, pesan yang sudah dibuat akan dikirimkan kepada *client* yang sudah ditentukan secara satu per satu untuk menjaga kerahasiaan *email client*.

1) Flowchart Create Broadcast

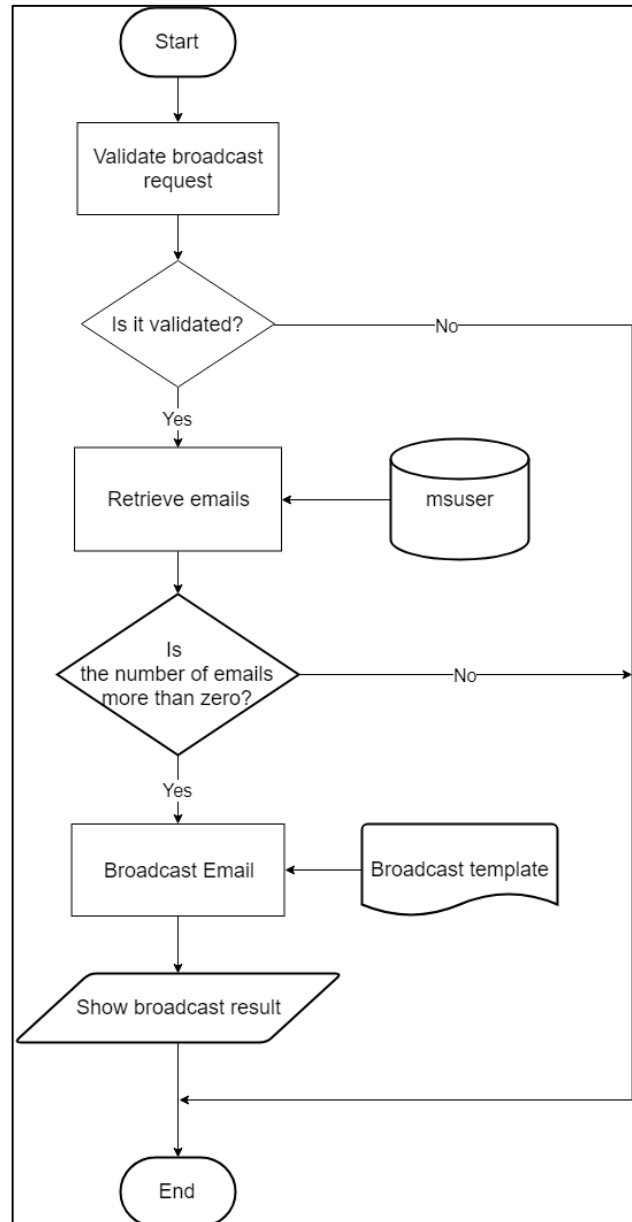
Ketika *user* memasuki halaman utama dari fitur *broadcast product*, *form* untuk membuat pesan yang akan di-broadcast akan ditampilkan. Setelah *form* tersebut diisi dan tombol “Submit” ditekan, fungsi *broadcast* akan dijalankan. Jika tombol “Submit” tidak ditekan, maka panel admin tidak melakukan fungsi *broadcast* dan fungsi apapun. *Flowchart* dari pembuatan pesan *broadcast* pada halaman utama fitur *broadcast product* dapat dilihat pada gambar 3.8.



Gambar 3.8 *Flowchart Create Broadcast*

2) Flowchart Broadcast

Gambar 3.9 menunjukkan *flowchart* untuk mengirimkan *broadcast* pada fitur *broadcast product*.



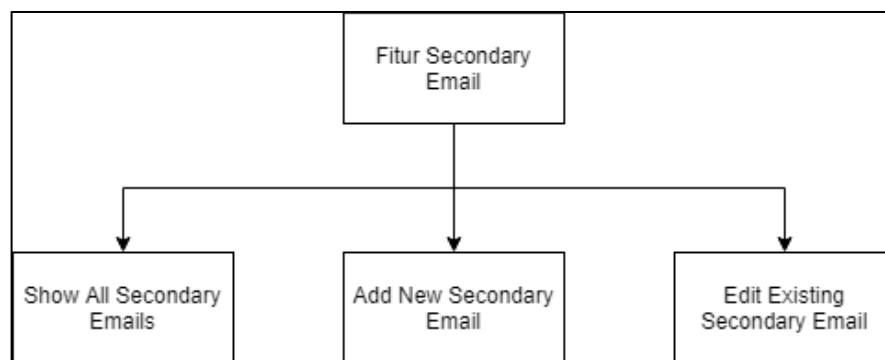
Gambar 3.9 *Flowchart Broadcast*

Berdasarkan gambar 3.9, hal pertama yang dijalankan pada fungsi *broadcast* adalah melakukan validasi terhadap masukan *user* dari *form* yang disediakan pada halaman utama fitur *broadcast product*. Jika validasi berhasil, data *email* setiap *client* berdasarkan masukan *user* akan diambil dari tabel *msuser*. Setelah itu, jumlah *email* yang didapat akan diperiksa. Jika jumlah *email* lebih besar dari nol, maka fungsi *broadcast* akan dilakukan.

Ketika melakukan *broadcast*, pesan yang sudah dimasukkan oleh *user* akan dimasukkan dalam sebuah *template broadcast* yang sudah dibuat. Pesan inilah yang akan diterima oleh *client*. Setelah itu, hasil pengiriman *broadcast* yang menunjukkan bahwa seluruh *broadcast* berhasil dikirimkan atau tidak akan ditampilkan pada halaman utama fitur *broadcast product*.

A.1.3. Flowchart Fitur Secondary Email

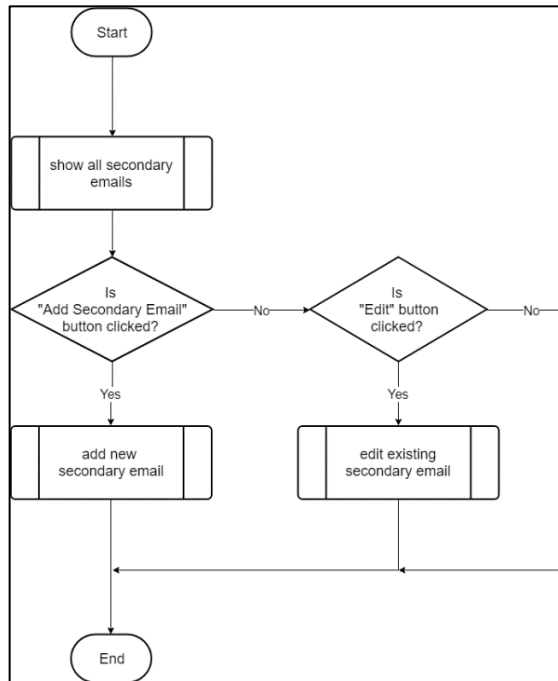
Gambar 3.10 menggambarkan hal-hal yang dapat dilakukan oleh *user* ketika menggunakan fitur *secondary email*.



Gambar 3.10 Fitur *Secondary Email*

Berdasarkan gambar 3.10, *user* dapat menampilkan semua *secondary email* yang terdaftar, menambahkan *secondary email*, dan memodifikasi *secondary email* yang sudah ada. *User* dapat menampilkan semua *secondary email* yang terdaftar ketika *user* mengunjungi halaman utama fitur *secondary email*. Selain itu, *user* juga dapat menambahkan dan memodifikasi *secondary email* yang sudah terdaftar setelah *user* menekan tombol tertentu. Bila *user* ingin menambahkan *secondary email*, *user* harus menekan tombol “Add Secondary Email”. Bila *user* ingin memodifikasi *secondary email*, *user* harus menekan tombol “Edit”.

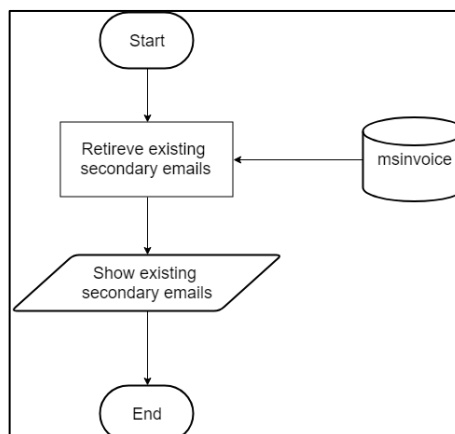
Flowchart yang menggambarkan dalam gambaran besar proses dari fitur *secondary email* ini dapat dilihat pada gambar 3.11.



Gambar 3.11 *Flowchart* Fitur *Secondary Email*

1) **Flowchart Show All Secondary Emails**

Gambar 3.12 menunjukkan *flowchart* untuk menampilkan semua data *secondary email* yang sudah tersimpan dalam *database*.

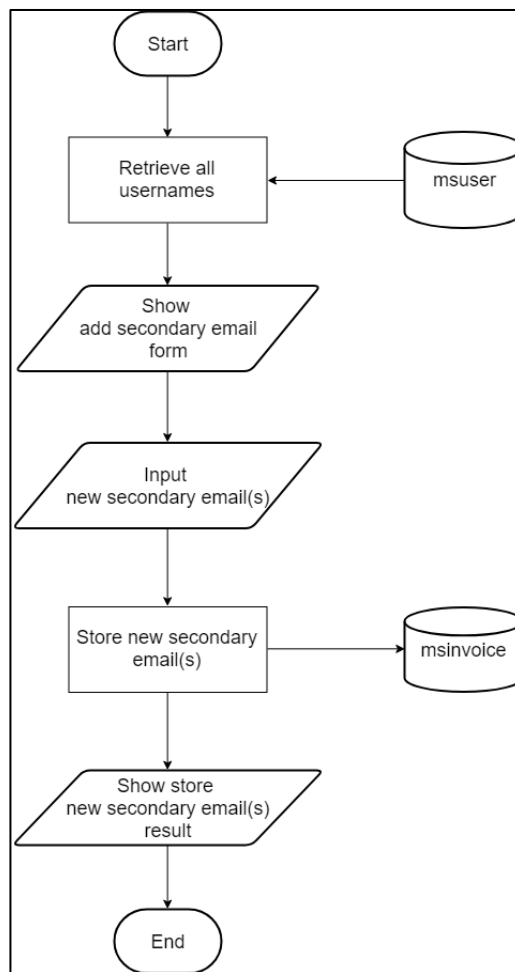


Gambar 3.12 *Flowchart* Show All Secondary Emails

Berdasarkan gambar 3.12, data *email* akan diambil terlebih dahulu dari tabel *msinvoice* sebelum ditampilkan pada halaman utama fitur *secondary email*. Selain data *email*, data *username client* juga diambil dari tabel tersebut. Setelah semua data tersebut diambil, data-data tersebut akan ditampilkan pada halaman utama fitur *secondary email*.

2) Flowchart Add New Secondary Email

Flowchart untuk menambahkan *secondary email* milik *client* yang baru dapat dilihat pada gambar 3.13.



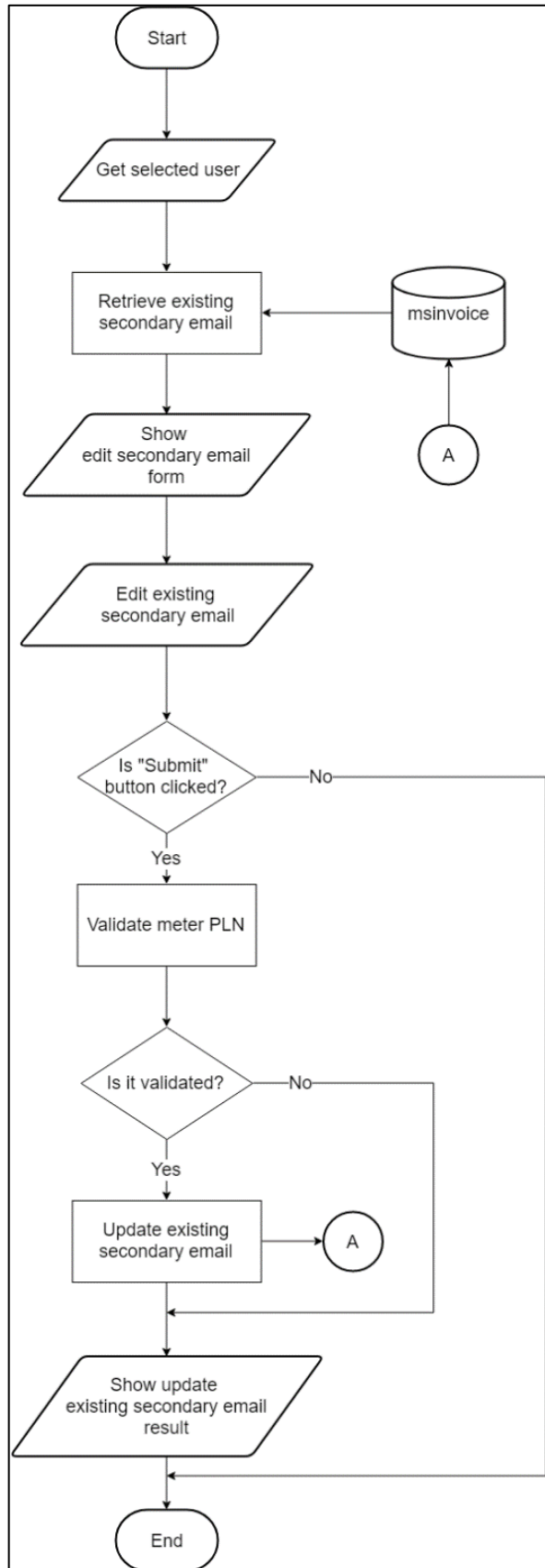
Gambar 3.13 *Flowchart Add New Secondary Email*

Berdasarkan gambar 3.13, data *username* milik *client* dari tabel *msuser* akan diambil terlebih dahulu sebelum *form* untuk menambahkan *email client* sebagai *secondary email* dimunculkan. Hal ini dilakukan supaya *user* hanya perlu memilih *username* dari *client* tersebut, sehingga kesalahan penulisan *username* milik *client* yang dilakukan oleh *user* dapat diminimalisir. *User* dapat memasukkan lebih dari satu *email* pada *form* tersebut. Setelah *form* selesai diisi dan tombol “Submit” ditekan, *email* akan dimasukkan ke dalam tabel *msinvoice*. Setelah itu, hasil yang menunjukkan bahwa seluruh *secondary email* berhasil dimasukkan atau tidak akan ditampilkan pada halaman utama fitur *secondary email*.

3) Flowchart Edit Existing Secondary Email

Setelah *user* menekan tombol “Edit”, *user* akan diarahkan menuju halaman baru yang berisi *form* untuk memodifikasi data *secondary email* yang sudah ada. Data yang dapat dimodifikasi dalam *form* ini hanya data *secondary email*, sehingga *user* tidak dapat mengubah data *username*. *User* dapat mengganti dan mengurangi *secondary email* yang dimiliki oleh *client*, tetapi *user* tidak dapat menghilangkan semua data *secondary email* dari *client* tersebut.

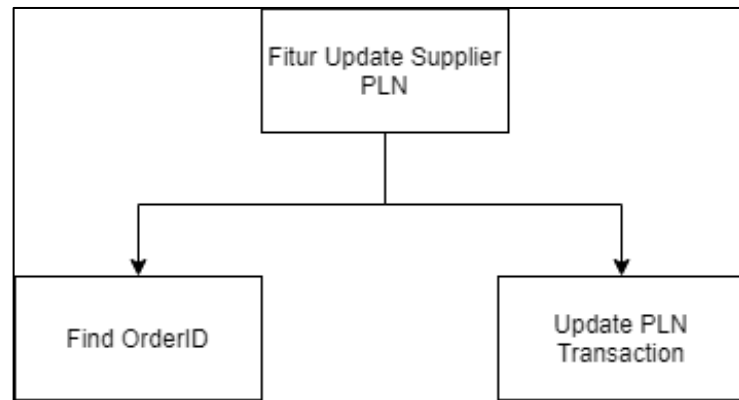
Setelah *user* mengisi *form* dan menekan tombol “Submit”, data masukkan *user* dari *form* akan divalidasi terlebih dahulu. Jika validasi berhasil, data *secondary email* pada tabel *msinvoice* akan diperbarui berdasarkan data masukkan *user*. Jika validasi gagal, maka tidak ada data yang diperbarui pada tabel *msinvoice*. Setelah itu, *user* akan menerima hasil yang menunjukkan bahwa modifikasi *secondary email* berhasil dilakukan atau tidak. *Flowchart* yang menggambarkan modifikasi data *secondary email* milik *client* dapat dilihat pada gambar 3.14.



Gambar 3.14 Flowchart Edit Existing Secondary Email

A.1.4. Flowchart Fitur Update Supplier PLN

Gambar 3.15 menggambarkan hal-hal yang dapat dilakukan oleh *user* ketika menggunakan fitur *update supplier* PLN.



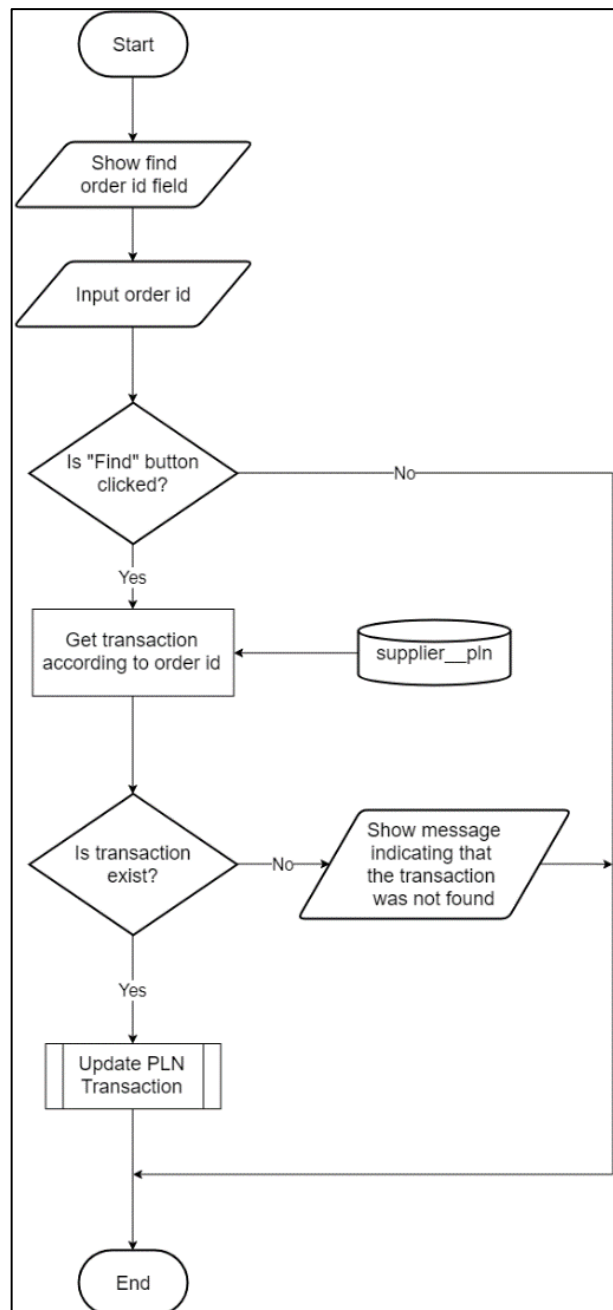
Gambar 3.15 Fitur *Update Supplier* PLN

Berdasarkan gambar 3.15, *user* dapat mencari orderID dari transaksi PLN yang sudah dilakukan oleh *client* dan memperbarui data transaksi PLN *client* tersebut berdasarkan data yang dimasukkan oleh *user* ketika menggunakan fitur *update supplier* PLN.

1) Flowchart Find OrderID

Ketika *user* mengunjungi halaman utama fitur *update supplier* PLN, *form* yang digunakan untuk mencari orderID dari transaksi produk PLN akan langsung ditampilkan pada halaman tersebut. *User* dapat memasukkan orderID dari transaksi *client* pada *form* tersebut. Setelah *user* selesai memasukkan orderID dan menekan tombol “Find”, data transaksi *client* tersebut akan diambil dari tabel *supplier__pln* berdasarkan orderID yang dimasukkan oleh *user*. Jika data transaksi berhasil didapat, maka fungsi *update PLN transaction* akan dilakukan.

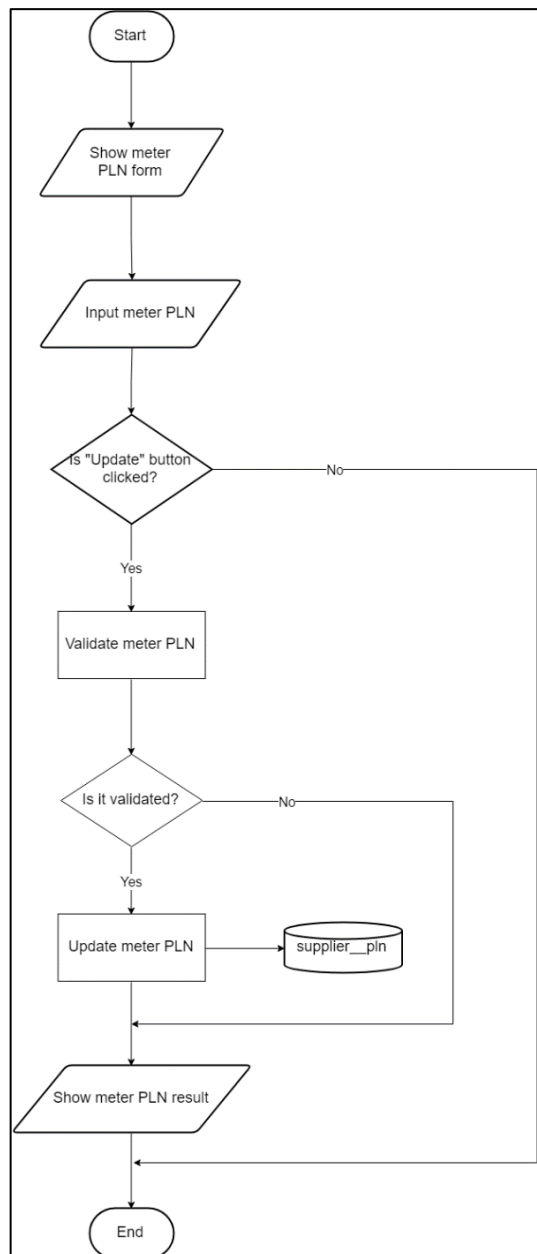
Jika data transaksi tidak berhasil didapat atau tidak ditemukan sesuai dengan orderID yang dimasukkan oleh *user*, maka *modal* yang berisi pesan bahwa transaksi tidak ditemukan akan ditampilkan pada halaman tersebut. *Flowchart* untuk mencari data transaksi PLN milik *client* berdasarkan masukan dari *user* dapat dilihat pada gambar 3.17.



Gambar 3.16 *Flowchart Find OrderID*

2) Flowchart Update PLN Transaction

Fungsi *update* PLN *transaction* hanya akan dijalankan apabila data transaksi dari tabel *supplier_pln* berhasil didapat pada fungsi *find* orderID. Gambar 3.17 menggambarkan *flowchart* untuk memperbarui data transaksi PLN milik *client* berdasarkan masukan dari *user*.



Gambar 3.17 *Flowchart Update PLN Transaction*

Setelah data transaksi PLN yang dicari didapatkan atau ditemukan, sebuah *form* akan dimunculkan di bawah *form* untuk mencari orderID. *Form* ini digunakan untuk memasukkan meteran PLN yang berhasil dibayarkan oleh *client*. Setelah *form* selesai diisi dan tombol “Update” pada *form* tersebut ditekan, masukkan dari *user* akan divalidasi terlebih dahulu. Jika validasi berhasil dilewati, data transaksi pada tabel *supplier_pln* akan diperbarui berdasarkan masukan dari *user*. Setelah itu, hasil yang menunjukkan data berhasil diperbarui atau tidak akan ditampilkan pada halaman utama fitur *update supplier* PLN.

A.2. Rancangan User Interface

A.2.1. Rancangan Tampilan Fitur Blacklist Product

Rancangan tampilan fitur *blacklist product* dibagi menjadi empat rancangan, yaitu rancangan tampilan untuk halaman *index*, halaman *add*, *modal delete*, dan *modal delete by merchant*.

1) Rancangan Halaman Index Fitur Blacklist Product

Pada saat *user* mengunjungi fitur *blacklist product*, *user* akan diarahkan menuju halaman *index* dari fitur *blacklist product*. Halaman *index* merupakan halaman utama fitur *blacklist product* yang disebutkan pada bagian *flowchart*. Halaman ini akan menampilkan nama *client* dan produk yang didaftarhitamkan. Selain itu, terdapat tombol yang dapat ditekan, yaitu tombol “Add” untuk menambahkan data produk dan *client* yang akan didaftarhitamkan, tombol “Delete” untuk menghapus data *client* dan produk dari daftar *blacklist*, dan tombol “Delete by Merchant” untuk menghapus semua data produk yang sudah didaftarhitamkan berdasarkan *client*.

Rancangan halaman *index* fitur *blacklist product* dapat dilihat pada gambar

3.18.

Mobilepulsa Hello, jason! Logout

Home / Pasca / Blacklist Products

Blacklist Products Delete by Merchant Add

Merchant	Product	Action
Merchant A	Product A	Delete
Merchant B	Product A	Delete
Merchant B	Product B	Delete
Merchant C	Product C	Delete
Merchant D	Product A	Delete
Merchant E	Product B	Delete

Gambar 3.18 Rancangan Halaman *Index* Fitur *Blacklist Product*

2) Rancangan Halaman Add Fitur Blacklist Product

Rancangan halaman *add* pada fitur *blacklist product* dapat dilihat pada gambar 3.19.

Mobilepulsa Hello, jason! Logout

Home / Pasca / Blacklist Products / Add Blacklist Product

Add Blacklist Products

Merchant

Method

Product

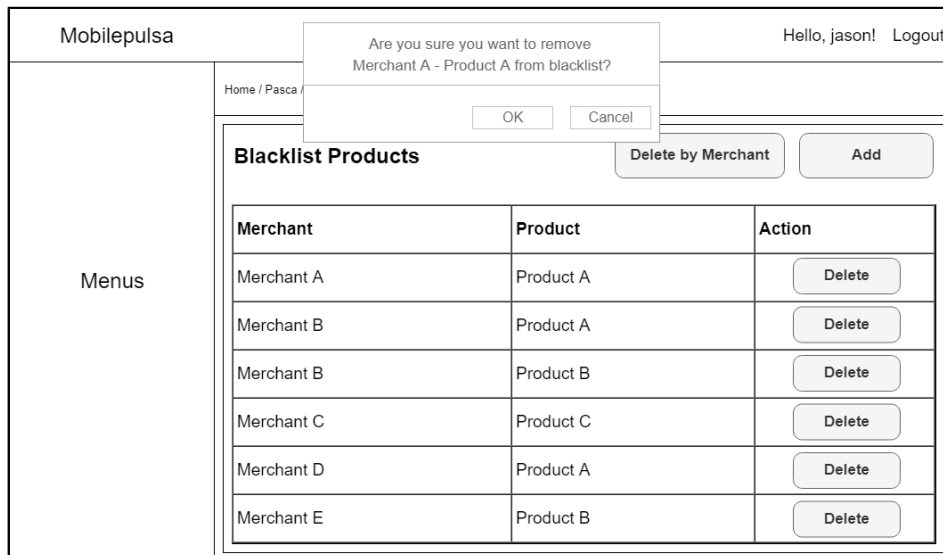
Submit Back

Gambar 3.19 Rancangan Halaman *Add* Fitur *Blacklist Product*

Terdapat dua tombol yang dapat ditekan oleh *user* pada *form* tersebut, yaitu tombol “Submit” untuk menambahkan data dan “Back” untuk kembali menuju halaman *index*. Selain itu, terdapat tiga *field* yang dicantumkan dalam *form*, yaitu *field merchant* untuk mengisi *username client* yang akan dimasukkan ke dalam daftar *blacklist*, *field method* untuk memilih antara memasukkan produk secara satu per satu atau sekaligus berdasarkan tipe produk, dan *field product* untuk memasukkan produk yang akan didaftarhitamkan bagi *client* yang dipilih pada *field merchant*.

3) Rancangan Modal Delete Fitur Blacklist Product

Rancangan *modal delete* pada fitur *blacklist product* dapat dilihat pada gambar 3.20.



Gambar 3.20 Rancangan *Modal Delete* Fitur *Blacklist Product*

Berdasarkan gambar 3.20, sebuah *modal* akan dimunculkan pada halaman *index* untuk mengonfirmasi bahwa suatu data akan dihapus setelah tombol “Delete” ditekan. *Modal* dimunculkan di daerah atas dari halaman *index*.

Nama *client* dan nama produk akan dimunculkan dalam *modal* tersebut. Terdapat dua tombol yang dapat ditekan oleh *user*, yaitu “OK” untuk menghapus data dan “Cancel” untuk menutup *modal* tersebut.

4) Rancangan Modal Delete by Merchant Fitur Blacklist Product

Setelah tombol “Delete by Merchant” pada halaman *index* ditekan, sebuah *modal* akan dimunculkan. *Modal* tersebut berisi *form* untuk memasukkan nama *client* yang akan dihapus dari daftar *blacklist* beserta dengan semua produk yang didaftarhitamkan pada *client* tersebut. Terdapat dua tombol yang dapat ditekan oleh *user*, yaitu tombol “Close” dan tombol “Delete”. Tombol “Close” memiliki fungsi untuk menutup *modal*. Tombol “Delete” memiliki fungsi untuk untuk menghapus *client* dan semua produk yang didaftarhitamkan berdasarkan *client* yang dimasukkan oleh *user*. Rancangan *modal delete by merchant* pada fitur *blacklist product* dapat dilihat pada gambar 3.21.

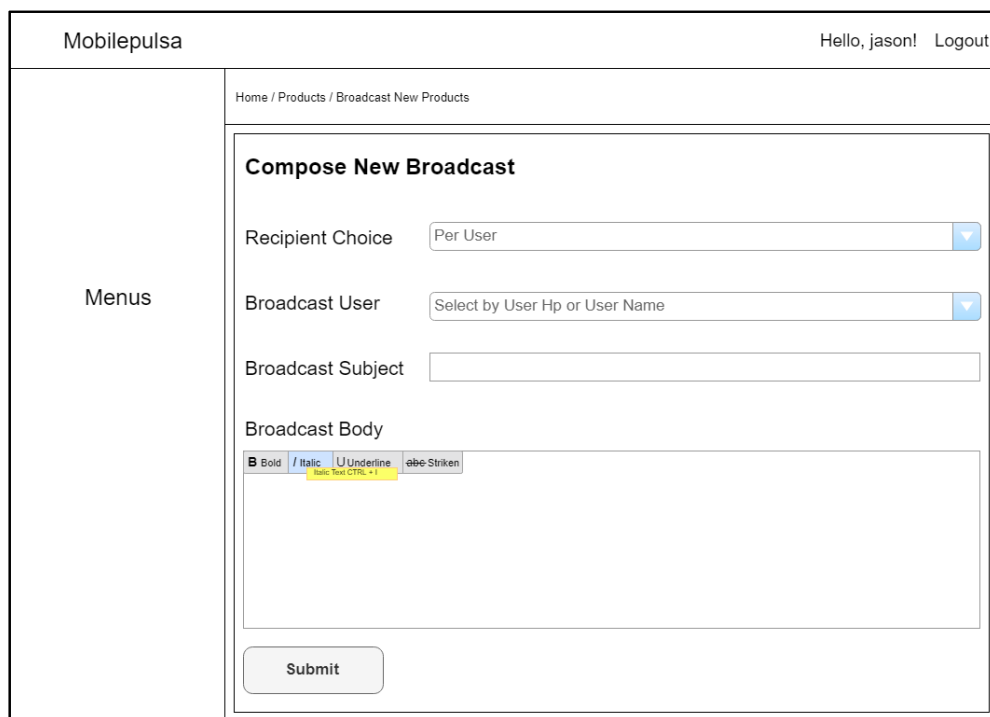
The screenshot shows a mobile application interface with a modal window. The modal is titled "Delete Blacklist Product by Merchant" and contains a form with a "Merchant" dropdown menu, an "Add" button, and "Close" and "Delete" buttons. Below the form is a table listing merchants and their products, each with a "Delete" button.

Merchant	Product	Action
Merchant A	Product A	Delete
Merchant B	Product A	Delete
Merchant B	Product B	Delete
Merchant C	Product C	Delete
Merchant D	Product A	Delete
Merchant E	Product B	Delete

Gambar 3.21 Rancangan *Modal Delete by Merchant* Fitur *Blacklist Product*

A.2.2. Rancangan Tampilan Fitur Broadcast Product

Pada fitur ini, hanya terdapat satu halaman yang dapat dikunjungi oleh *user*, yaitu halaman *create* yang berisi *form* untuk mengirimkan *broadcast* kepada *client*. Pada form tersebut, *user* dapat memilih penerima dari *broadcast* yang dikirimkan, mengisi *subject* dari *broadcast* yang dikirimkan, dan mengisi pesan yang akan dikirimkan dalam *broadcast*. Hanya terdapat satu tombol yang dapat ditekan pada halaman ini, yaitu tombol “Submit” untuk mengirimkan *broadcast* berdasarkan *form* yang telah diisi. Rancangan halaman *create* pada fitur *broadcast product* dapat dilihat pada gambar 3.22.



The screenshot shows a web interface for 'Compose New Broadcast'. At the top left is the logo 'Mobilepalsa' and at the top right is the user greeting 'Hello, jason!' and a 'Logout' link. Below the header is a breadcrumb trail: 'Home / Products / Broadcast New Products'. The main content area is titled 'Compose New Broadcast' and contains the following elements:

- Recipient Choice:** A dropdown menu currently showing 'Per User'.
- Broadcast User:** A dropdown menu with the text 'Select by User Hp or User Name'.
- Broadcast Subject:** A text input field.
- Broadcast Body:** A rich text editor with a toolbar containing 'B Bold', 'I Italic', 'U Underline', and 'abc Strikethrough'. The text area is currently empty.
- Submit:** A button at the bottom of the form.

A 'Menus' sidebar is visible on the left side of the form area.

Gambar 3.22 Rancangan Halaman *Create* Fitur *Broadcast Product*

A.2.3. Rancangan Tampilan Fitur Secondary Email

Rancangan tampilan fitur *secondary email* dibagi menjadi tiga, yaitu rancangan tampilan untuk halaman *index*, halaman *add*, dan halaman *edit*.

1) Rancangan Halaman Index Fitur Secondary Email

Ketika *user* mengunjungi fitur *secondary email*, *user* akan diarahkan menuju halaman *index*. Halaman *index* merupakan halaman utama fitur *secondary email* yang disebutkan pada bagian *flowchart*. Halaman ini akan menampilkan nama *client* dan *email* yang dijadikan sebagai *secondary email* dari *client* tersebut dalam bentuk tabel. Selain itu, terdapat tombol “Edit” untuk memodifikasi data *secondary email* yang sudah tersimpan dalam *database* dan tombol “Add Secondary Email” pada halaman *index* untuk menambahkan *secondary email* milik *client*. Rancangan halaman *index* pada fitur *secondary email* dapat dilihat pada gambar 3.23.

Username	Secondary Email	Action
Merchant A	email1, email2, email3	Edit
Merchant B	email1	Edit
Merchant C	email1, email2	Edit
Merchant D	email1, email2, email3	Edit
Merchant E	email1, email2	Edit
Merchant F	email1	Edit

Gambar 3.23 Rancangan Halaman *Index* Fitur *Secondary Email*

2) Rancangan Halaman Add Fitur Secondary Email

Setelah *user* menekan tombol “Add Secondary Email” pada halaman *index*, *user* akan diarahkan menuju halaman *add* yang berisi *form* untuk menambahkan *secondary email* milik *client*.

Terdapat dua *field* yang dapat diisi oleh *user* pada *form* tersebut, yaitu *field username* untuk memasukkan *username client* dan *field secondary email* untuk memasukkan *email client* sebagai *secondary email*. Terdapat lima tombol yang dapat ditekan oleh *user*, yaitu tombol “Add Email” untuk menambahkan satu *field secondary email*, tombol “Delete Email” untuk menghilangkan satu *field* terakhir dari *field secondary email*, tombol “Reset” untuk mengosongkan seluruh komponen *form*, tombol “Submit” untuk menambahkan *secondary email*, dan tombol “Back” untuk mengarahkan *user* kembali menuju halaman *index*. Rancangan halaman *add* pada fitur *secondary email* dapat dilihat pada gambar 3.24.

Mobilepalsa	Hello, jason! Logout
Menus	Home / Users / Secondary Email / Add
	<p>Add Secondary Email</p> <p>Username <input type="text"/></p> <p>Secondary Email <input type="text"/></p> <p><input type="button" value="Add Email"/> <input type="button" value="Delete Email"/> <input type="button" value="Reset"/> <input type="button" value="Submit"/> <input type="button" value="Back"/></p>

Gambar 3.24 Rancangan Halaman *Add* Fitur *Secondary Email*

3) Rancangan Halaman Edit Fitur Secondary Email

Setelah *user* menekan tombol “Edit” pada kolom “Action” pada salah satu baris dalam tabel, *user* akan diarahkan menuju halaman *edit* yang berisi *form* untuk memodifikasi data *secondary email* yang sudah tersimpan dalam *database*.

Field secondary email pada *form* tersebut sudah berisi *nilai* yang diisi dengan data yang tersimpan dalam *database*, sehingga jumlah *field secondary email* menyesuaikan dengan jumlah *secondary email* yang tersimpan dalam *database*. *User* hanya dapat memodifikasi bagian *secondary email* dalam *form* tersebut.

User dapat menghapus *field secondary email* dengan menekan tombol dengan logo tempat sampah dan dapat mengosongkan *field secondary email* dengan menekan tombol yang berbentuk silang. Selain itu, terdapat tiga tombol yang dapat ditekan oleh *user*, yaitu tombol “Reset”, tombol “Submit”, dan tombol “Back”. Tombol “Reset” untuk mengubah *value* pada *field* menjadi *value* ketika halaman *edit* ditampilkan. Tombol “Submit” untuk memodifikasi data *secondary email*. Tombol “Back” untuk mengarahkan *user* kembali menuju halaman *index*.

Rancangan halaman *edit* pada fitur *secondary email* dapat dilihat pada gambar 3.25.

Mobilepalsa	Hello, jason! Logout
	Home / Users / Secondary Email / Edit
Menus	Edit Secondary Email
	Username: 085693651653
	Secondary Email: <input type="text" value="email1@email.com"/> x
	Secondary Email: <input type="text" value="email2@email.com"/> x
	Secondary Email: <input type="text" value="email3@email.com"/> x
	<input type="button" value="Reset"/> <input type="button" value="Submit"/> <input type="button" value="Back"/>

Gambar 3.25 Rancangan Halaman *Edit* Fitur *Secondary Email*

A.2.4. Rancangan Tampilan Fitur Update Supplier PLN

Fitur *update supplier* PLN hanya memiliki satu halaman yang dapat dikunjungi oleh *user*, yaitu halaman *index*. Halaman *index* pada fitur ini akan menampilkan sebuah *form* yang digunakan untuk mencari transaksi produk PLN. *Form* tersebut hanya berisi satu *field* untuk mencari orderID. *User* dapat memasukkan orderID dari transaksi *client* pada *field* dalam *form* tersebut. Setelah *form* selesai diisi, *user* dapat menekan tombol “Find” untuk mencari transaksi PLN berdasarkan orderID yang dimasukkan oleh *user* pada *field* dalam *form* tersebut. Rancangan halaman *index* untuk menampilkan *form* untuk mencari orderID dapat dilihat pada gambar 3.26.

Mobilepulsa		Hello, jason! Logout	
Menus	Home / Pasca / Update Supplier PLN		
	Update Supplier PLN		
	Order ID	<input type="text"/>	Find

Gambar 3.26 Rancangan Halaman *Index* Fitur *Update Supplier* PLN untuk Mencari Transaksi

Jika transaksi ditemukan, *form* untuk memperbarui data transaksi PLN *client* akan dimunculkan di bawah *form* untuk mencari orderID. *User* dapat memasukkan meteran pertama dan terakhir sesuai dengan bulan yang dibayarkan.

User dapat menekan tombol “Update” untuk memperbarui data transaksi tersebut. Rancangan halaman *index* pada fitur *update supplier* PLN untuk memperbarui data transaksi dapat dilihat pada gambar 3.27.

Mobilepalsa	Hello, jason! Logout
	Home / Pasca / Update Supplier PLN
Menus	Update Supplier PLN
	Order ID <input type="text" value="123456789"/> <input type="button" value="Find"/>
	Update PLN
	August Period <input type="text" value="First Meter"/> - <input type="text" value="Last Meter"/>
	<input type="button" value="Update"/>

Gambar 3.27 Rancangan Halaman *Index* Fitur *Update Supplier* PLN untuk Memperbarui Data Transaksi

B. Implementasi

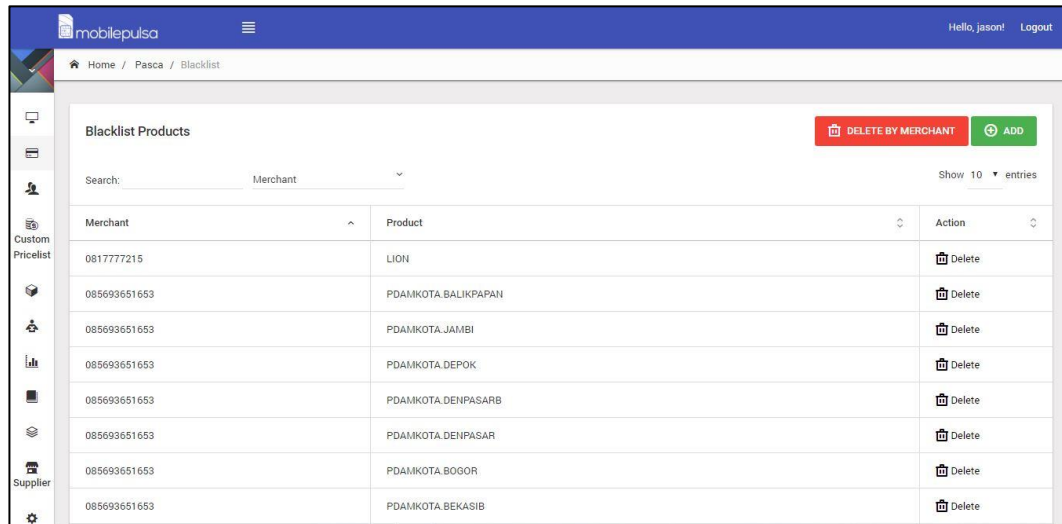
B.1. Fitur Blacklist Product

Implementasi tampilan fitur *blacklist product* dibagi menjadi empat, yaitu implementasi untuk halaman *index* halaman *add*, *modal delete*, dan *modal delete by merchant*.

B.1.1. Halaman Index

Halaman *index* memuat tabel dengan kolom *merchant* yang berisi *username client*, kolom *product* yang berisi nama produk, dan kolom *action* yang berisi tombol “Delete” untuk menghapus data pada barisan tersebut.

Implementasi dari rancangan tampilan halaman *index* pada fitur *blacklist product* dapat dilihat pada gambar 3.28.



The screenshot shows the 'Blacklist Products' page in the mobilepulsas application. The page has a blue header with the logo and user information. Below the header, there is a search bar and a table of products. The table has three columns: Merchant, Product, and Action. The Action column contains a 'Delete' button for each row. There are also buttons for 'DELETE BY MERCHANT' and 'ADD' at the top right of the table area.

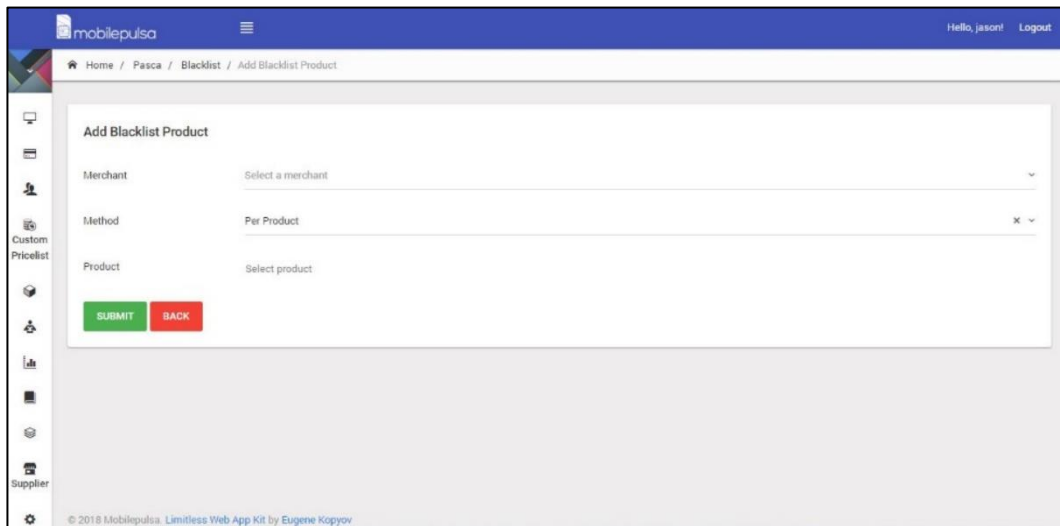
Merchant	Product	Action
0817777215	LION	Delete
085693651653	PDAMKOTA.BALIKPAPAN	Delete
085693651653	PDAMKOTA.JAMBI	Delete
085693651653	PDAMKOTA.DEPOK	Delete
085693651653	PDAMKOTA.DENPASARB	Delete
085693651653	PDAMKOTA.DENPASAR	Delete
085693651653	PDAMKOTA.BOGOR	Delete
085693651653	PDAMKOTA.BEKASIB	Delete

Gambar 3.28 Implementasi Halaman *Index* Fitur *Blacklist Product*

B.1.2. Halaman Add

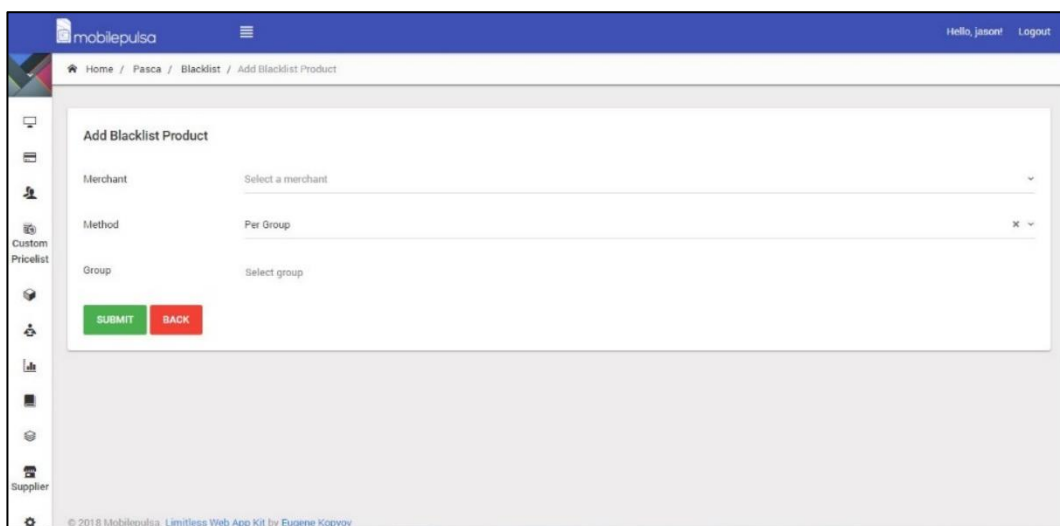
Implementasi dari rancangan halaman *add* pada fitur *blacklist product* yang memasukkan produk secara satu per satu dapat dilihat pada gambar 3.27. Terdapat tiga *field* yang dapat diisi oleh *user* pada *form* yang disediakan, yaitu *field merchant*, *field method*, dan *field product* atau *field group*. *Field merchant* sebagai tempat untuk mengisi *username client*. *Field method* sebagai pilihan *user* untuk memasukkan produk secara satu per satu atau beberapa produk sekaligus sesuai dengan tipe produk. *Field product* sebagai tempat untuk memasukkan produk yang didaftarhitamkan secara satu per satu. *Field group* untuk memasukkan tipe produk yang akan didaftarhitamkan pada *client* tersebut. *Field* yang digunakan pada *form* ini berupa komponen *select* untuk menghindarkan *user* dari kesalahan penulisan. Terdapat dua pilihan yang dapat dipilih oleh *user* pada *field method*, yaitu “Per Product” atau “Per Group”.

Jika *user* memilih pilihan “Per Product”, *field product* akan ditampilkan. Jika *user* memilih pilihan “Per Group”, *field group* akan ditampilkan. Halaman *add fitur blacklist product* akan ditampilkan seperti gambar 3.29 ketika *user* memilih pilihan “Per Product”.



Gambar 3.29 Implementasi Halaman *Add Per Product* Fitur *Blacklist Product*

Halaman akan ditampilkan seperti gambar 3.30 ketika *user* memilih pilihan “Per Group” untuk memilih produk berdasarkan tipe produk tersebut.

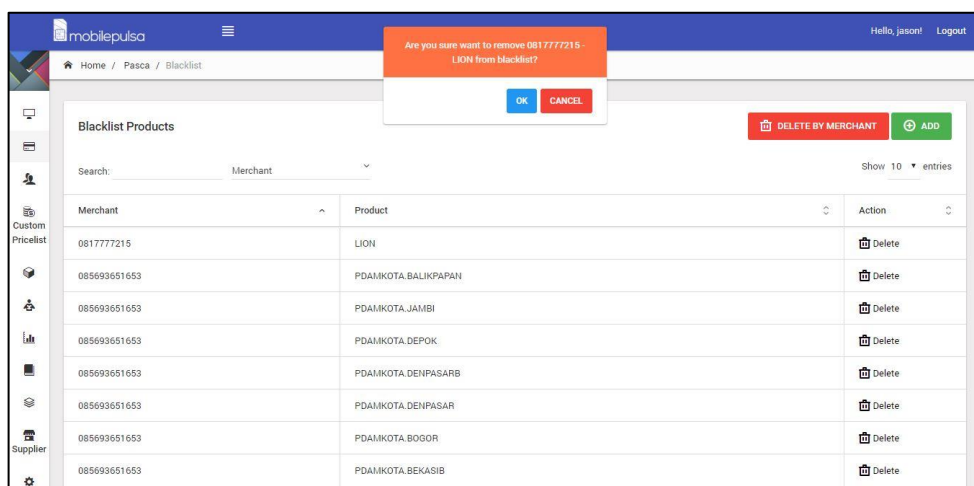


Gambar 3.30 Implementasi Halaman *Add Per Group* Fitur *Blacklist Product*

Perbedaan halaman ini dengan halaman sebelumnya adalah *field group* dan *field product*. Selain dari perbedaan tersebut, kedua halaman ini diimplementasikan dengan *field* dan cara yang sama.

B.1.3. Modal Delete

Implementasi dari *modal delete* pada fitur *blacklist product* dapat dilihat pada gambar 3.31.



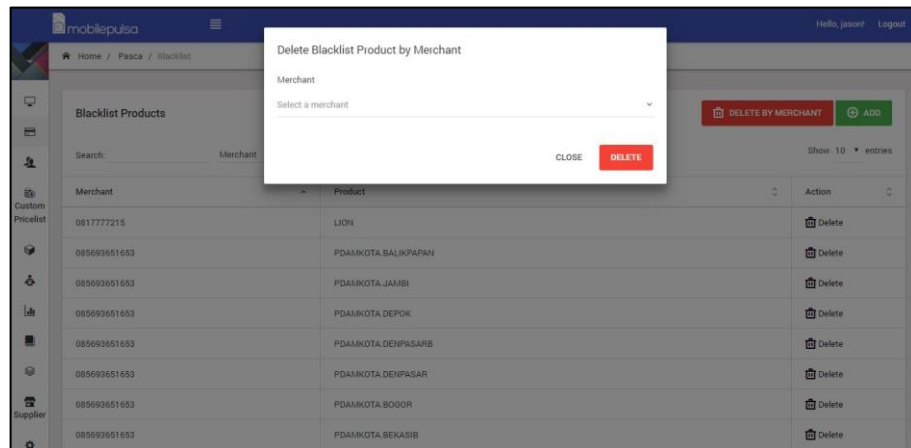
Gambar 3.31 Implementasi *Modal Delete* Fitur *Blacklist Product*

Berdasarkan gambar 3.31, *modal* diberi warna merah supaya *modal* dapat menarik perhatian *user*. *Modal* tersebut akan dihilangkan dari tampilan ketika *user* menekan tombol yang ada dalam *modal* tersebut, baik tombol “OK” maupun tombol “Cancel”.

B.1.4. Modal Delete by Merchant

Implementasi *modal delete by merchant* pada fitur *blacklist product* dapat dilihat pada gambar 3.32. Halaman *index* akan memunculkan *modal delete by merchant* ketika tombol “Delete by Merchant” ditekan.

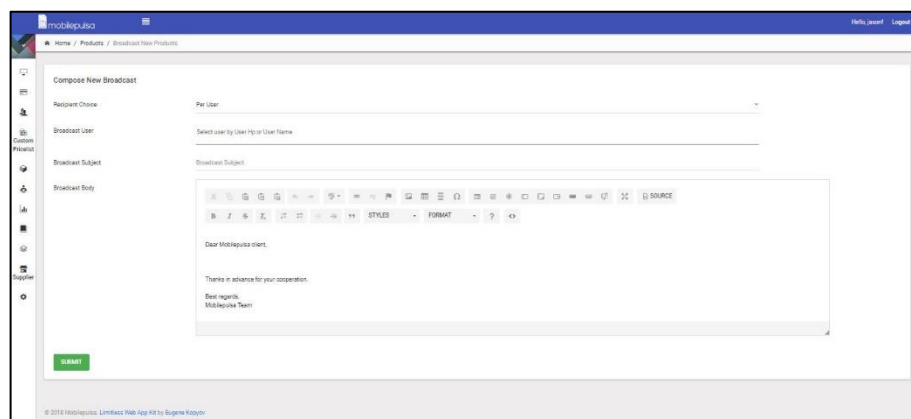
Komponen yang digunakan dalam memasukkan nama *client* adalah komponen *select* untuk menghindarkan *user* dari kesalahan penulisan *username client*. *Modal* akan dihilangkan ketika *user* menekan tombol “Delete” atau tombol “Close”.



Gambar 3.32 Implementasi *Modal Delete by Merchant* Fitur *Blacklist Product*

B.2. Fitur Broadcast Product

Fitur *broadcast product* hanya memiliki satu halaman yang dapat digunakan oleh *user*, yaitu halaman *create*. Implementasi dari halaman *create* pada fitur *broadcast product* dapat dilihat pada gambar 3.33.



Gambar 3.33 Implementasi Halaman *Create* Fitur *Broadcast Product*

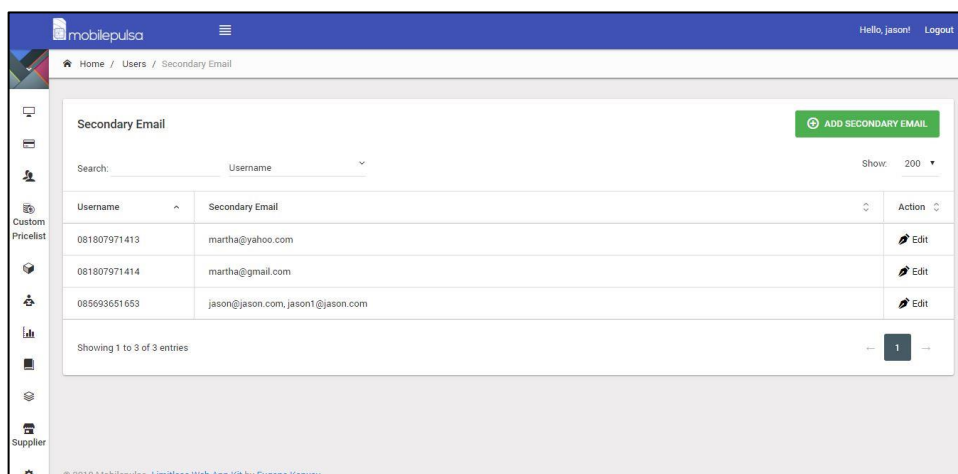
Field yang disediakan pada *form* ditujukan untuk pengiriman *email*, sehingga terdapat bagian judul *broadcast* (subject) dan isi *broadcast* (broadcast body). *User* hanya dapat memasukkan *client* yang memiliki *email* utama pada akun *client* tersebut dan *user* dapat melihat *email client* tersebut pada *field broadcast user*. *Field broadcast body* ditambahkan sebuah *toolbar* untuk memodifikasi tampilan pesan pada *broadcast*.

B.3. Fitur Secondary Email

Implementasi tampilan fitur *secondary email* dibagi menjadi tiga, yaitu implementasi untuk halaman *index* halaman *add*, dan halaman *edit*.

B.3.1. Halaman Index

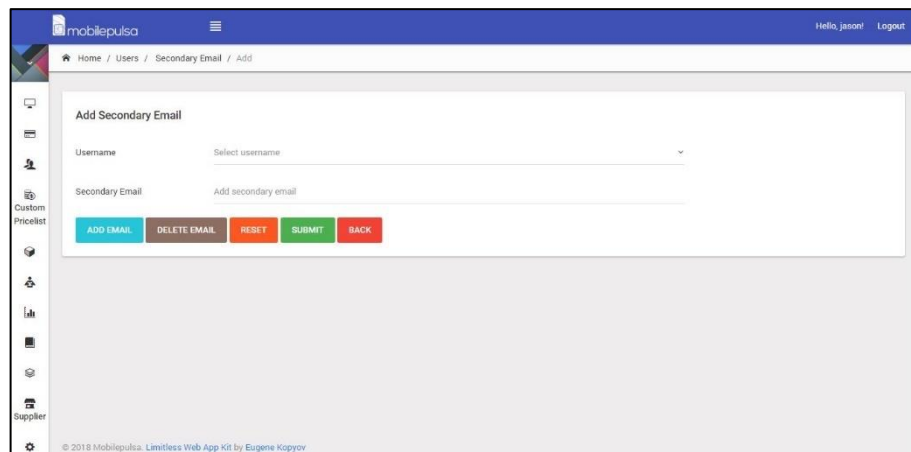
Implementasi halaman *index* pada fitur *secondary email* dapat dilihat pada gambar 3.34. Halaman *index* memuat tabel yang berisi kolom *username client*, kolom *secondary email*, dan kolom *actions* yang berisi tombol “Edit” untuk memodifikasi data pada barisan yang dipilih. Data *email* pada kolom *secondary email* yang berjumlah lebih dari satu akan diletakkan bersebelahan dengan tanda baca koma sebagai pemisah.



Gambar 3.34 Implementasi Halaman *Index* Fitur *Secondary Email*

B.3.2. Halaman Add

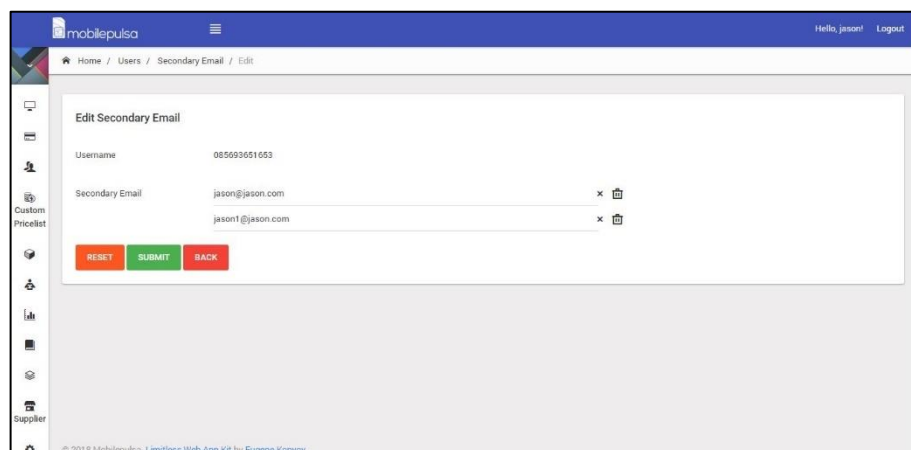
Implementasi halaman *add* pada fitur *secondary email* dapat dilihat pada gambar 3.35. Komponen yang digunakan dalam *form* ini adalah komponen *select* untuk field *username* dan komponen *input* bertipe *email* untuk field *secondary email*.



Gambar 3.35 Implementasi Halaman *Add* Fitur *Secondary Email*

B.3.3. Halaman Edit

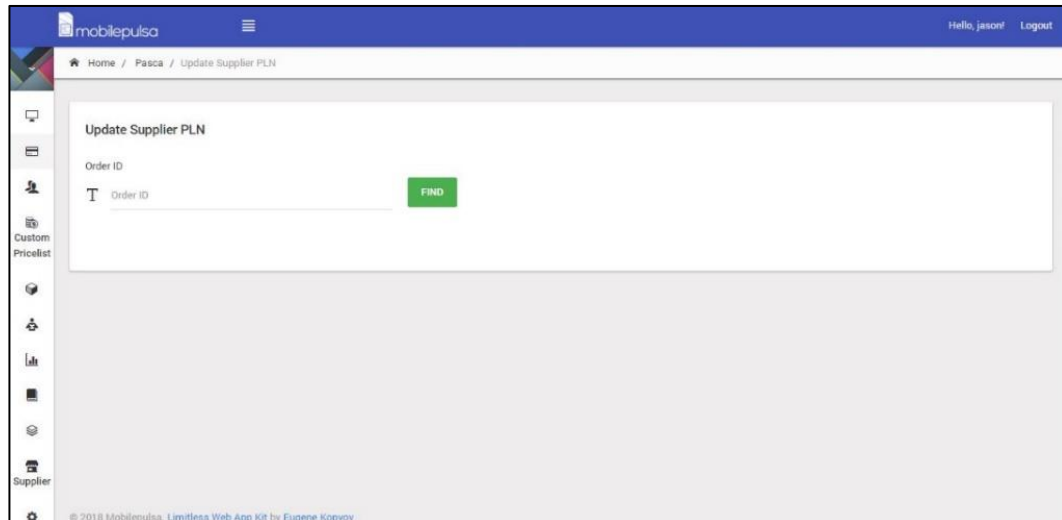
Implementasi halaman *edit* pada fitur *secondary email* dapat dilihat pada gambar 3.36.



Gambar 3.36 Implementasi Halaman *Edit* Fitur *Secondary Email*

B.4. Fitur Update Supplier PLN

Implementasi dari halaman *index* fitur *update supplier* PLN untuk mencari data transaksi dapat dilihat pada gambar 3.37.

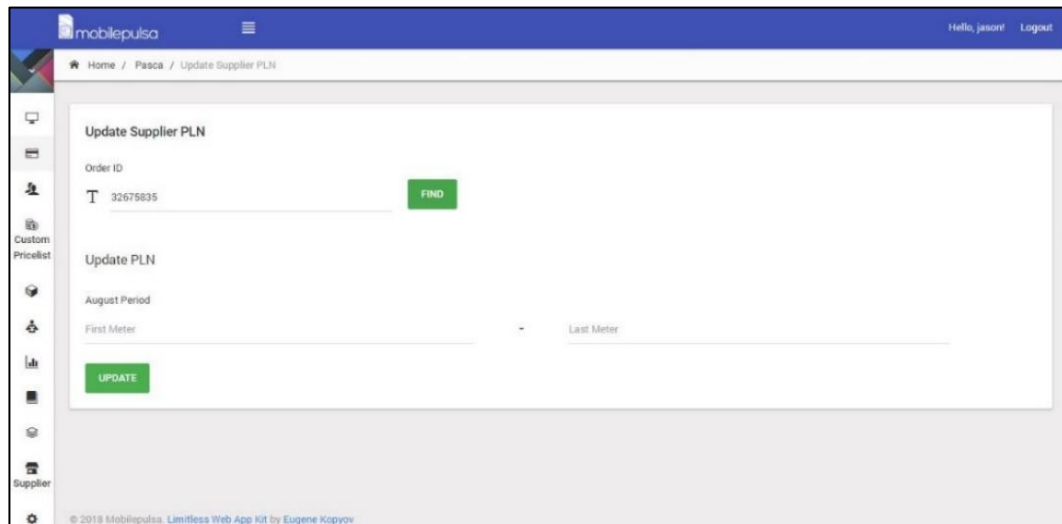


Gambar 3.37 Implementasi Halaman *Index* Fitur *Update Supplier* PLN untuk Mencari Transaksi

Berdasarkan gambar 3.37, *field* pada *form* tersebut hanya menerima masukan berupa angka dan harus bernilai lebih besar dari satu atau sama dengan satu. Jika *field* pada *form* tersebut diisi dengan masukan yang bukan angka atau berupa angka yang lebih kecil dari nol, tombol “Find” pada *form* tidak bisa ditekan. Hal ini untuk mencegah *user* memasukkan *value* pada *field* dengan *value* yang tidak seharusnya yang dapat menyebabkan terjadinya *error* pada panel admin.

Setelah *form* diisi, tombol “Find” ditekan, dan transaksi ditemukan, *form* untuk memperbarui data meteran dari tagihan yang berhasil dibayarkan oleh *client* akan dimunculkan. Aturan *field* pada *form* ini sama seperti aturan *field* pada *form* sebelumnya, yaitu masukan yang diterima pada *field* tersebut harus berupa angka dan bernilai lebih besar dari satu atau sama dengan satu.

Jika aturan dilanggar, tombol “Update” pada *form* tidak dapat ditekan, sehingga fitur untuk memperbarui data transaksi tidak akan dilakukan. Implementasi dari halaman *index* fitur *update supplier* PLN untuk memperbarui data transaksi dapat dilihat pada gambar 3.38.



Gambar 3.38 Implementasi Halaman *Index* Fitur *Update Supplier* PLN untuk Memperbarui Data Transaksi

3.3.2. Kendala yang Ditemukan

Kendala yang ditemukan selama melakukan kerja magang adalah sebagai berikut.

- a. Ppenambahan fitur harus menggunakan *framework* Laravel yang belum pernah dipelajari sebelumnya.
- b. Penambahan fitur harus mengikuti prinsip *clean code* yang tidak familiar dilakukan oleh peserta magang.
- c. Proses bisnis produk Mobilepulsas yang belum pernah diketahui dan dipelajari sebelumnya.

3.3.3. Solusi atas Kendala yang Ditemukan

Solusi yang dilakukan untuk mengatasi kendala yang ditemukan adalah sebagai berikut.

- a. Mempelajari *framework* Laravel dari berbagai sumber, termasuk menonton *video tutorial*, mengikuti *training* yang disediakan oleh perusahaan, membaca dokumentasi yang disediakan oleh Laravel, dan bertanya kepada rekan kerja dan *supervisor*.
- b. Mempelajari struktur *script* dari projek panel admin Mobilepulsa, mempelajari fitur-fitur penunjang prinsip *clean code* dari dokumentasi Laravel, dan bertanya kepada rekan kerja dan *supervisor*.
- c. Mempelajari proses bisnis produk Mobilepulsa dari situs developer API Mobilepulsa, dan bertanya kepada rekan kerja dan *supervisor*.