



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

Peserta magang diberi kedudukan sebagai *software engineer* di dalam divisi Whitelabel. Divisi Whitelabel bertanggung jawab untuk memberikan pelayanan bagi *customer* Whitelabel dan pengembangan produk Whitelabel baru. Pelayanan Whitelabel untuk memberikan permintaan fitur yang diinginkan ataupun menyelesaikan kendala yang mungkin dialami. Tugas yang diberikan dari perusahaan dilakukan selama kerja magang dipimpin oleh Eddy Christiandy selaku CTO (*Chief Technical Officer*) dan Dominicus Jovian Owen Hoetama selaku *Senior Programmer*. Peserta magang berada dibawah bimbingan dan pengawasan dari Dominicus Jovian Owen Hoetama. Koordinasi tugas menggunakan Trello dan kemudian pindah kepada Fresh Release karena ingin bereksperimen kepada divisi Whitelabel untuk menggunakan fitur *sprint* yang ada pada Fresh Release.

3.2 Tugas yang Dilakukan

Tugas yang diberikan kepada peserta magang terdiri dari pembuatan fitur laporan pada panel admin Whitelabel baru. Fitur laporan ini akan memiliki halaman utama yang menyediakan laporan-laporan dari modul terkait. Modul-modul terkait tersebut terdiri dari empat modul, yaitu modul pascabayar, prabayar, pengguna, dan pembayaran. Bagian pascabayar dan prabayar memiliki laporan yang sama yaitu laporan keuntungan, laporan penjual per produk, laporan penjualan per agen. Bagian pembayaran terdapat laporan

transfer deposit dan laporan penambahan deposit. Bagian pengguna hanya berisi laporan mengenai perkembangan pengguna.

Laporan keuntungan akan menampilkan pendapatan keuntungan yang didapatkan dari transaksi pada tanggal tertentu. Laporan penjualan per agen adalah laporan transaksi yang telah dilakukan oleh masing-masing agen dalam jarak waktu tertentu. Laporan penjualan per produk menunjukkan jumlah penjualan per produk yang telah terjadi. Laporan transfer deposit akan menunjukkan jumlah deposit yang telah ditransfer dari seorang agen ke agen lainnya. Laporan penambahan deposit adalah laporan yang menampilkan jumlah deposit yang telah diterima pada suatu jarak tanggal. Laporan perkembangan pengguna adalah laporan yang menunjukkan jumlah agen yang telah mendaftar pada tanggal tertentu.

Seluruh tabel yang yang ditampilkan pada halaman laporan direpresentasikan menggunakan *plugin* DataTables. Data diambil dari *database* yang telah disediakan oleh perusahaan. Seluruh laporan akan melakukan *query* berdasarkan *range* tanggal yang akan di-*input*-kan oleh admin. Setiap laporan memiliki fungsi untuk menggeserkan kolom yang ada pada tabel sesuka admin. Laporan juga dapat dibuat kedalam bentuk PDF dan Excel. Kolom yang ada pada laporan juga akan dapat dihilangkan sementara dan kemudian dapat dinyalakan kembali melalui sebuah *toggle*. *User Interface* untuk tombol *fetch data* dan *range of date* akan diambil dari *component* Element UI. Pembuatan halaman laporan menggunakan *framework* Laravel dengan bahasa pemrograman PHP. *Database*

yang digunakan adalah dengan MySQL dan MongoDB. Tugas-tugas yang dilakukan selama kerja magang disertakan melalui Tabel 3.1.

Tabel 3.1 Realisasi kerja magang

Minggu	Pekerjaan yang Dilakukan
1	<ul style="list-style-type: none"> ● <i>Briefing jobdesc</i> dan pengenalan ● Belajar <i>framework</i> Laravel 4.2 ● Instalasi <i>software</i> yang akan digunakan ● Instalasi dan mempelajari Whitelabel lama
2	<ul style="list-style-type: none"> ● Belajar <i>framework</i> Laravel 4.2 ● Pembuatan fitur <i>bluetooth printer</i> untuk Whitelabel Sijek Pulsa
3	<ul style="list-style-type: none"> ● Belajar <i>framework</i> Laravel 5.8 ● Instalasi dan mempelajari Whitelabel baru
4	<ul style="list-style-type: none"> ● Pembuatan Whitelabel Nuritronik ● Halaman laporan users: Growth User Report ● Implementasi <i>plugin</i> DataTables.
5	<ul style="list-style-type: none"> ● Fitur memindahkan kolom tabel ● Fitur ekspor PDF dan Excel ● Halaman laporan pembayaran: laporan transfer dan laporan penambahan deposit
6	<ul style="list-style-type: none"> ● Fitur sembunyikan kolom tabel laporan ● Halaman laporan prabayar: laporan penjualan per agen
7	<ul style="list-style-type: none"> ● Halaman laporan prabayar: laporan penjualan per produk dan laporan keuntungan ● Pembetulan <i>bug</i> dari fitur tabel.
8	<ul style="list-style-type: none"> ● Halaman laporan pascabayar: laporan keuntungan, laporan penjualan per produk, laporan penjualan per agen ● Pembuatan <i>template</i> halaman laporan
9	<ul style="list-style-type: none"> ● Implementasi <i>bugsnag</i> ● Tabel transaksi pascabayar
10	<ul style="list-style-type: none"> ● Proses <i>inquiry</i> pascabayar
11	<ul style="list-style-type: none"> ● Update Whitelabel multipay ● Update tabel transaksi pascabayar pada panel admin
12	<ul style="list-style-type: none"> ● Tampilan <i>inquiry</i> pascabayar

Framework Laravel digunakan karena banyaknya fitur yang dapat mempermudah dan mempercepat proses pembuatan halaman *website* pada umumnya. Laravel membantu *developer* untuk menuliskan *code* yang lebih baik dengan menghindari potongan *code* berulang-ulang untuk menjaga struktur aplikasi yang rapi (Bean, 2015). *Database* MongoDB dipilih sebagai *data store* kedua disamping MySQL. MongoDB menawarkan kemudahan dalam menyimpan data dalam bentuk *document* dan performa yang tinggi (Chodorow, 2013).

3.3 Uraian Pelaksanaan Kerja Magang

Uraian Pelaksanaan kerja magang dibagi menjadi tiga bagian yaitu, proses pelaksanaan, kendala yang ditemukan, dan solusi atas kendala yang ditemukan.

3.3.1 Struktur Aplikasi

Website utama Whitelabel merupakan aplikasi tunggal yang mampu membuat website milik *customer*. Aplikasi Whitelabel memiliki database pusat yang berisi informasi mengenai daftar website Whitelabel yang terdaftar. Daftar Whitelabel yang sudah terdaftar dibedakan dengan sebuah *UID*.

Aplikasi Whitelabel ini akan dapat diakses oleh banyak orang dari satu *server*. Setiap Whitelabel akan memiliki satu buah *database* miliknya masing-masing. Nama *database* yang dimiliki oleh sebuah Whitelabel didapatkan dari *UID* dari Whitelabel tersebut. Sehingga, aplikasi Whitelabel dapat menentukan database yang dapat diakses oleh masing-masing Whitelabel. Hal ini yang digunakan untuk membedakan data yang ada pada satu Whitelabel dengan Whitelabel yang lainnya.

3.3.2 Proses Pelaksanaan

Pembuatan modul laporan untuk panel admin dibuat menggunakan *software* dan *hardware*. *Software* yang digunakan adalah sebagai berikut.

1. *Framework* Laravel 5.8
2. Element UI v2.7.0
3. WampServer v3.1.9 (64-bit), Apache v2.4.39, PHP v7.3.5, MySQL v5.7.26, dan MariaDB v10.3.14.
4. Robo 3T v1.3.1, MongoDB v3.6.3
5. IDE Visual Studio Code v1.37.1
6. Fork v1.38.0.0
7. Google Chrome v76.0.3809.132 (64-bit)
8. Sistem Operasi Windows 10 Home (64-bit)

Perangkat keras yang digunakan untuk membuat fitur laporan ini adalah ASUS ROG GL553VD dengan spesifikasi sebagai berikut.

1. CPU Intel i7-7700HQ
2. RAM 8 GB
3. NVIDIA Geforce GTX 1050 4GB
4. Harddisk 1 TB

A. Perancangan Sistem

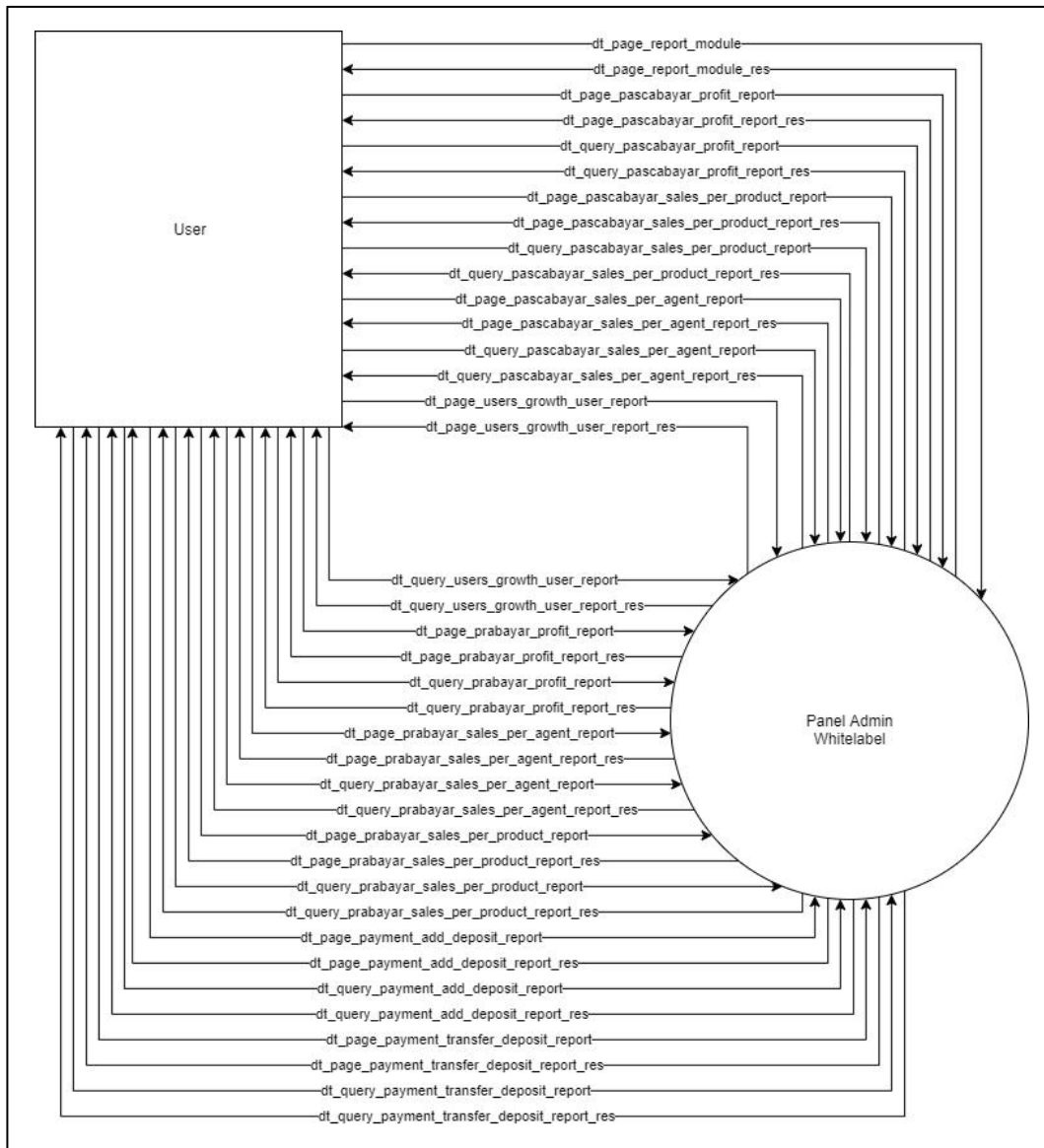
Perancangan sistem dilakukan pembuatan *Data Flow Diagram* (DFD), *Flowchart*, dan perancangan *User Interface* (UI).

A.1 Data Flow Diagram

DFD adalah alat analisis sistem yang terstruktur dan tervisualisasi sehingga memungkinkan untuk penganalisis sistem dan subsistem dapat memahami aliran dan rangkaian data yang saling berkaitan satu dengan lainnya (Kendall & Kendall, 2010).

A.1.1 DFD Context Diagram

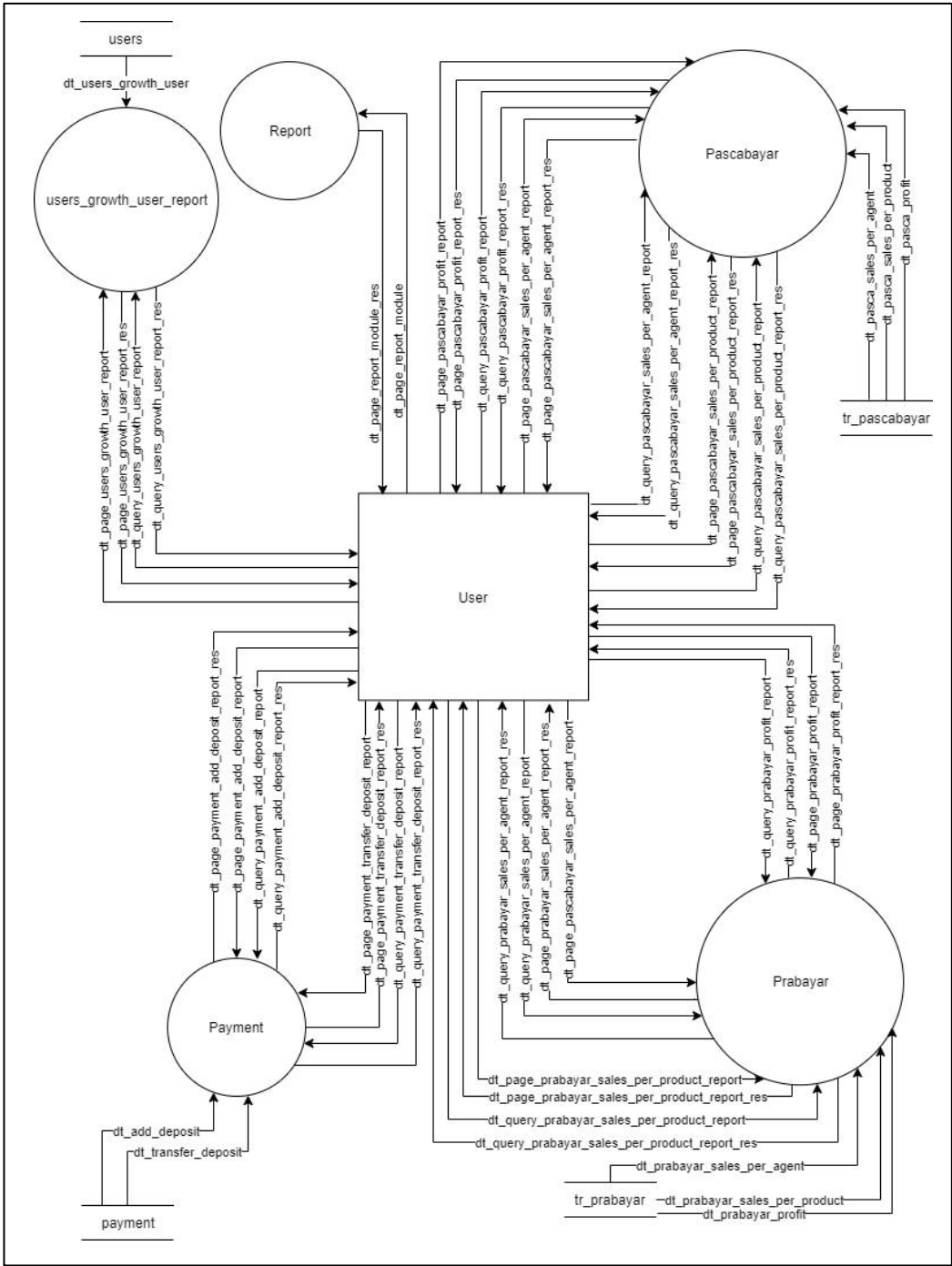
Modul laporan ini hanya akan digunakan oleh satu entitas yaitu, user. User disini adalah admin dari masing-masing *client* Whitelabel dari Mobilepulsa. Pertukaran data ini dilakukan oleh beberapa submenu yang akan diberikan. Submenu tersebut adalah pascabayar, *payment*, prabayar, dan *user*. Gambar 3.1 adalah gambar dari DFD Context Diagram dari modul laporan pada panel admin



Gambar 3.1 Context diagram

A.1.2 DFD Level 1

DFD Level 1 dibagi menjadi lima proses. Lima proses yang ada pada DFD Level 1 adalah pascabayar, *payment*, prabayar, users, dan *report*. Proses *report* berguna untuk menampilkan laporan apa saja yang sudah tersedia dalam panel admin untuk setiap modul. Masing-masing proses, selain proses *report* dan *users_growth_user_report*, akan memproses submenu sesuai dengan nama prosesnya. Proses *users_growth_user_report* bertugas untuk mengelola laporan Growth User. Proses ini menerima permintaan halaman berupa *dt_page_growth_user_report* sebagai permintaannya dan mengembalikan *dt_page_growth_user_report_res* sebagai halamannya. Selain itu, proses ini juga menerima permintaan *query* berupa *dt_query_growth_user_report* dan mengembalikan hasilnya *dt_query_growth_users_report_res*. Submenu user terhubung kepada tabel users sebagai *data store*. Ke-empat proses lainnya akan diperjelas kembali pada diagram DFD Level 2.



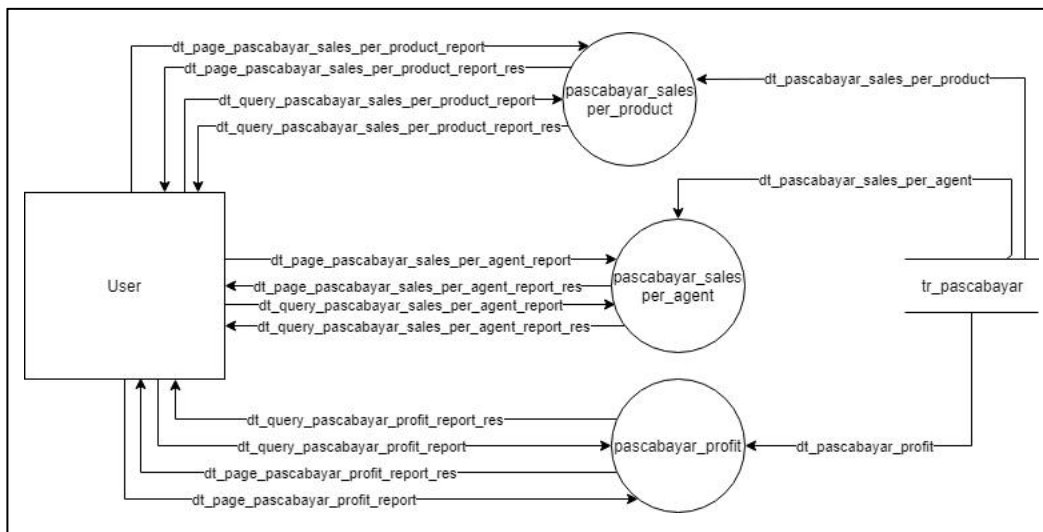
Gambar 3.2 DFD Level 1

A.1.3 DFD Level 2

1. Proses pada Submenu Pascabayar

Submenu Pascabayar tiga buah proses, yaitu `pascabayar_profit`, `pascabayar_sales_per_agent`, dan `pascabayar_sales_per_product`. Tiga proses yang ada pada submenu pascabayar memiliki tugasnya masing-masing. Proses pertama yaitu `pascabayar_profit`. Proses `pascabayar_profit` bertugas untuk mengelola Laporan Profit Pascabayar. Laporan Profit Pascabayar akan menampilkan penghasilan, jumlah transaksi dan omset yang didapatkan per satuan hari dari transaksi pascabayar yang berhasil. Proses berikutnya adalah `pascabayar_sales_per_agent`. Proses `pascabayar_sales_per_agent` bertugas untuk mengelola laporan Pascabayar Sales per Agent. Laporan Pascabayar Sales per Agent menunjukkan penghasilan berdasarkan agen yang terdaftar pada masing-masing Whitelabel. Laporan ini dikelompokan berdasarkan agen-agen yang terdaftar dan telah melakukan transaksi pascabayar yang berhasil. Proses ketiga adalah `pascabayar_sales_per_product`. Proses ini bertugas untuk mengelola laporan Pascabayar Sales per Product. Laporan Pascabayar per Product memberikan rincian transaksi berdasarkan produk pascabayar yang berhasil. Masing-masing proses memiliki pola arus perpindahan data yang mirip. Masing-masing proses akan mengelola permintaan halaman berupa data `dt_page_pascabayar_sales_per_product_report` untuk halaman laporan Pascabayar Sales per Product berupa `dt_page_pascabayar_sales_per_product_report_res`, `dt_page_pascabayar_sales_per_agent_report` untuk halaman laporan Pascabayar Sales per Agent berupa `dt_page_pascabayar_sales_per_agent_report_res` dan `dt_page_pascabayar_profit_report` untuk halaman laporan Pascabayar Profit

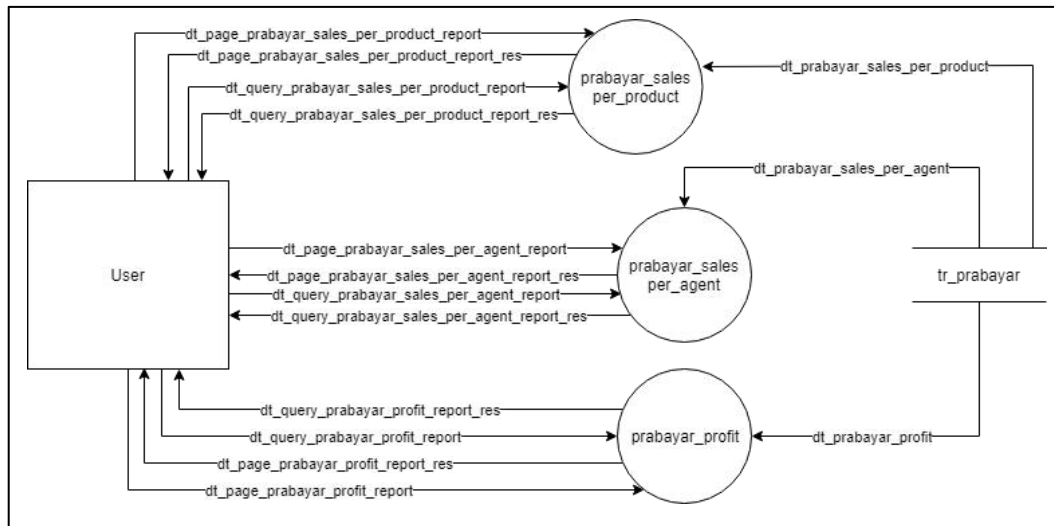
berupa `dt_page_pascabayar_profit_report_res`. Masing-masing proses juga akan mengelola permintaan data transaksi berupa *query* berdasarkan jarak tanggal yang dimasukan oleh admin. Data-data tersebut adalah `dt_query_pascabayar_sales_per_product_report` untuk *query* data laporan Pascabayar Sales per Product berupa `dt_query_pascabayar_sales_per_product_report_res`, `dt_query_pascabayar_sales_per_agent_report` untuk *query* laporan Pascabayar Sales per Agent berupa `dt_query_pascabayar_sales_per_agent_report_res` dan `dt_query_pascabayar_profit_report` untuk *query* laporan Pascabayar Profit berupa `dt_query_pascabayar_profit_report_res`. Seluruh aliran data yang terjadi pada submenu pascabayar terhubung kedalam tabel `tr_pascabayar`. Gambar 3.3 menunjukkan gambar dari DFD Level 2 proses Pascabayar.



Gambar 3.3 DFD Level 2 Proses Pascabayar

2. Proses pada Submenu Prabayar

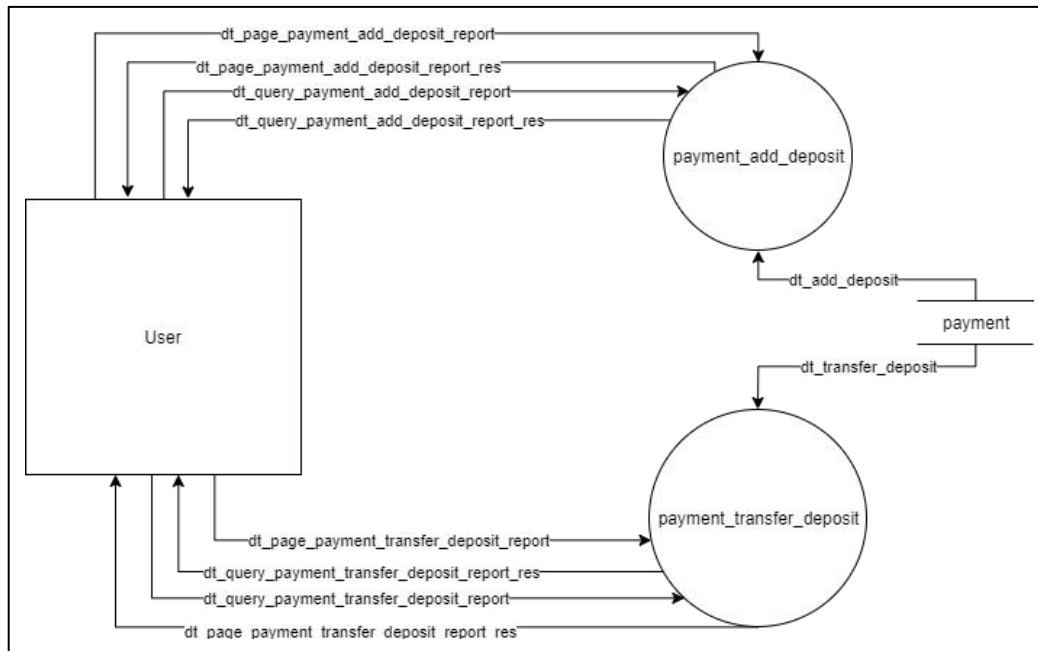
Submenu prabayar mempunyai tiga proses yaitu `prabayar_profit`, `prabayar_sales_per_agent`, dan `prabayar_sales_per_product`. Proses pada submenu prabayar memiliki proses yang memiliki kesamaan pada submenu Pascabayar pada sebelumnya. Perbedaan yang dimiliki oleh submenu pascabayar dan prabayar adalah table pada *database* yang berbeda. Proses `prabayar_sales_per_product` mengelola laporan Prabayar Sales per Product. Proses ini menerima `dt_query_prabayar_sales_per_product_report` untuk permintaan laporan Prabayar Sales per Product dan dikembalikan berupa `dt_query_prabayar_sales_per_product_report_res`. Proses `prabayar_sales_per_agent` mengelola laporan Prabayar Sales per Agent. Proses ini menerima `dt_query_prabayar_sales_per_agent_report` untuk permintaan laporan Prabayar Sales per Product dan dikembalikan berupa `dt_query_prabayar_sales_per_agent_report_res`. Proses yang ketiga adalah `prabayar_profit`. Proses ini bertugas untuk mengelola laporan Prabayar Profit. Proses ini menerima `dt_query_prabayar_profit_report` untuk permintaan laporan Prabayar Sales per Product dan dikembalikan berupa `dt_query_prabayar_profit_report_res`. Aliran data yang ada pada submenu prabayar terhubung kepada tabel `tr_prabayar`. Gambar 3.4 menunjukkan DFD Level 2 Proses Prabayar.



Gambar 3.4 DFD Level 2 Proses Prabayar

3. Proses pada submenu Payment

Submenu *payment* memiliki dua proses yaitu *add_deposit* dan *transfer_deposit*. Proses *payment_add_deposit* mengelola laporan *Payment Add Deposit*. Proses ini menerima permintaan halaman berupa *dt_page_payment_add_deposit_report* dan mengembalikan hasilnya berupa *dt_page_payment_add_deposit_report_res*. *Query* yang diberikan pada proses ini adalah *dt_query_payment_add_deposit_report* dan mengembalikan laporannya berupa *dt_query_payment_add_deposit_report_res*. Proses *payment_transfer_deposit* mengelola laporan *Payment Transfer Deposit*. Proses ini menerima permintaan halaman berupa *dt_page_payment_transfer_deposit_report* dan mengembalikan hasilnya berupa *dt_page_payment_transfer_deposit_report_res*. *Query* yang diberikan pada proses ini adalah *dt_query_payment_transfer_deposit_report* dan mengembalikan laporannya berupa *dt_query_payment_transfer_deposit_report_res*. Aliran data terhubung pada tabel *payment* sebagai *data store*. Gambar 3.5 menunjukkan DFD Level 2 Submenu *Payment*.



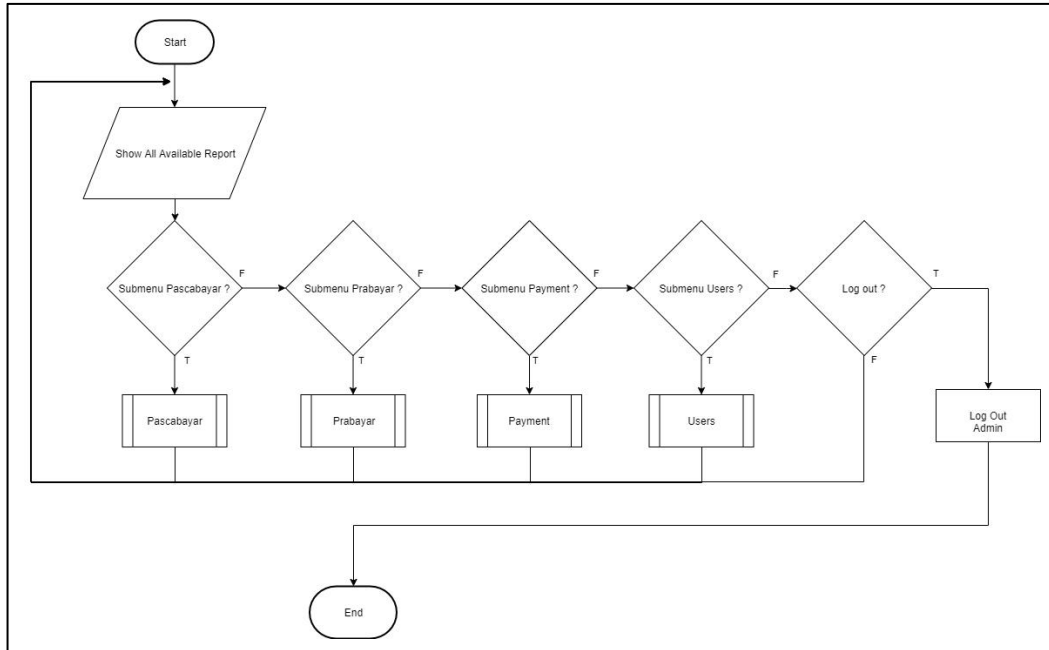
Gambar 3.5 DFD Level 2 Proses Payment

A.2 Flowchart

Flowchart adalah penggambaran melalui grafik dari suatu langkah-langkah berupa prosedur guna menyelesaikan sebuah masalah yang memvisualisasikan proses pengolahan data yang terjadi (Robertson, 2006).

A2.1 Flowchart Halaman Modul Laporan

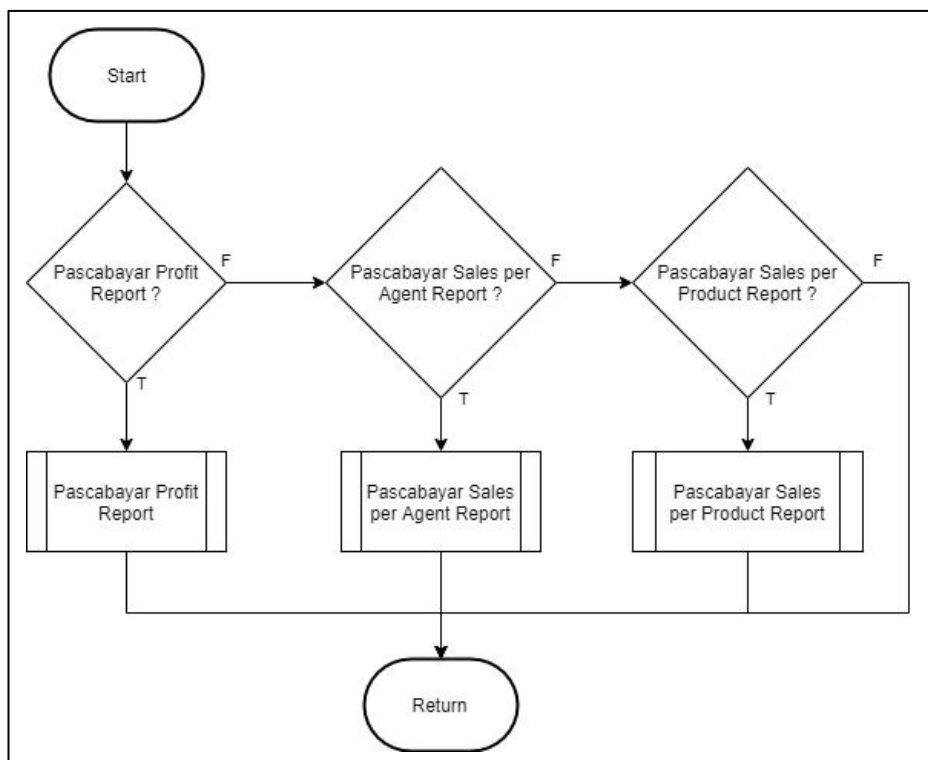
Seluruh laporan yang tersedia akan ditampilkan terlebih dahulu yang kemudian akan dapat dijadikan panduan untuk bernavigasi untuk admin. Selanjutnya, admin akan memilih submenu yang kemudian akan dilanjutkan untuk penentuan laporan apa yang dipilih didalam submenunya. Submenu yang dibangun adalah pascabayar, prabayar, *payment*, dan *user*. Modul akan berhenti ketika admin melakukan *log out*. Proses Log Out User sudah dikembangkan sebelumnya. Gambar 3.6 menunjukan *flowchart* halaman modul laporan.



Gambar 3.6 Flowchart Halaman Modul Laporan

A.2.2 Flowchart Submenu Pascabayar

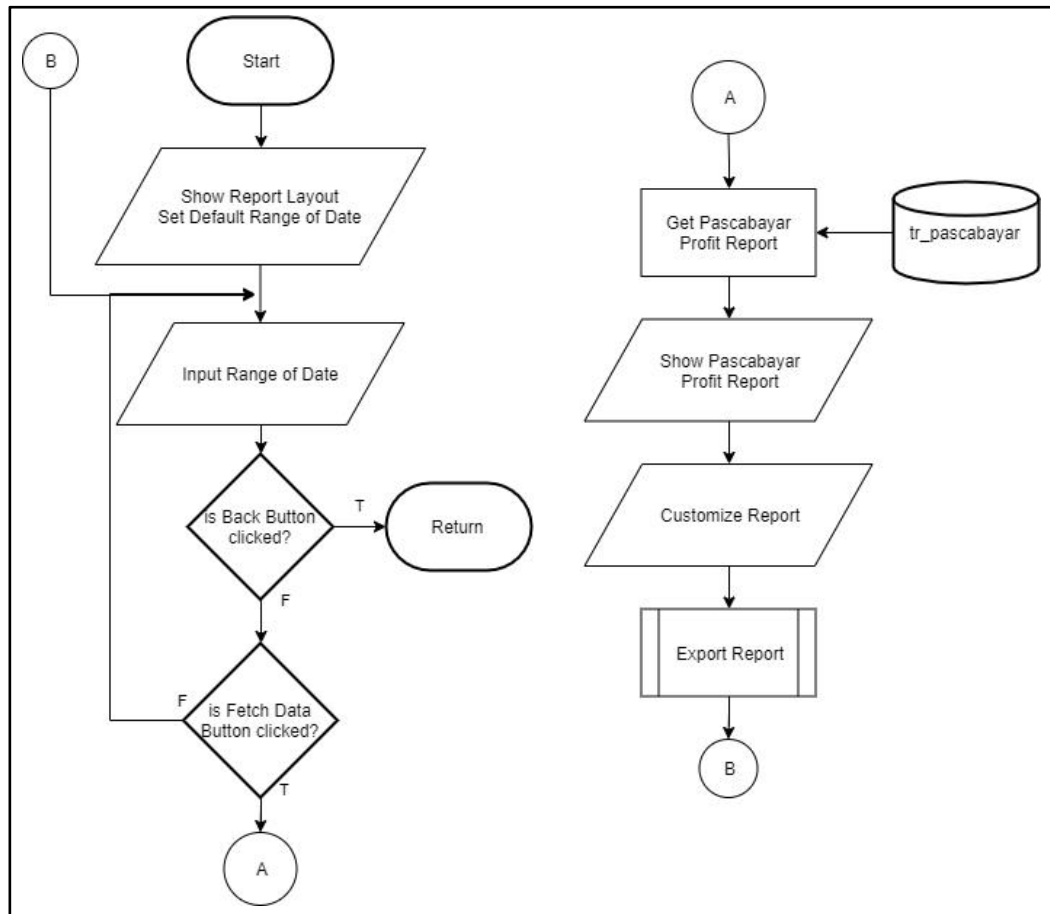
Submenu pascabayar akan mengantarkan admin kepada pilihan laporan yang tersedia untuk transaksi pascabayar. Pemilihan yang tersedia adalah laporan Pascabayar Profit, laporan Pascabayar Sales per Agent dan laporan Pascabayar Sales per Product. Selanjutnya akan diteruskan kedalam modul laporan yang sesuai dengan apa yang telah diinginkan oleh admin. Gambar 3.7 menunjukkan *Flowchart* submenu pascabayar.



Gambar 3.7 Flowchart Submenu Pascabayar

1. Flowchart Modul Pascabayar Profit Report

Modul Pascabayar Profit Report akan menampilkan *layout* dari halaman laporan dan menetapkan *Range of Date* secara *default*. Admin dapat memasukan jarak tanggal yang diinginkannya. Jika admin menekan tombol Fetch Data, maka aplikasi akan menjalankan Get Pascabayar Profit Report dari `tr_pascabayar` yang kemudian akan ditampilkan dalam bentuk tabel. Tabel yang ditampilkan akan dapat dikostumisasi kolomnya. Kolom yang ada pada tabel dapat dihilangkan dan atau dipindahkan. Laporan diteruskan kepada modul Export Report untuk kemudian dapat diunduh. Jika sudah selesai admin dapat keluar dengan menekan tombol Back untuk kembali. Gambar 3.8 menunjukan *flowchart* modul Pascabayar Profit Report.

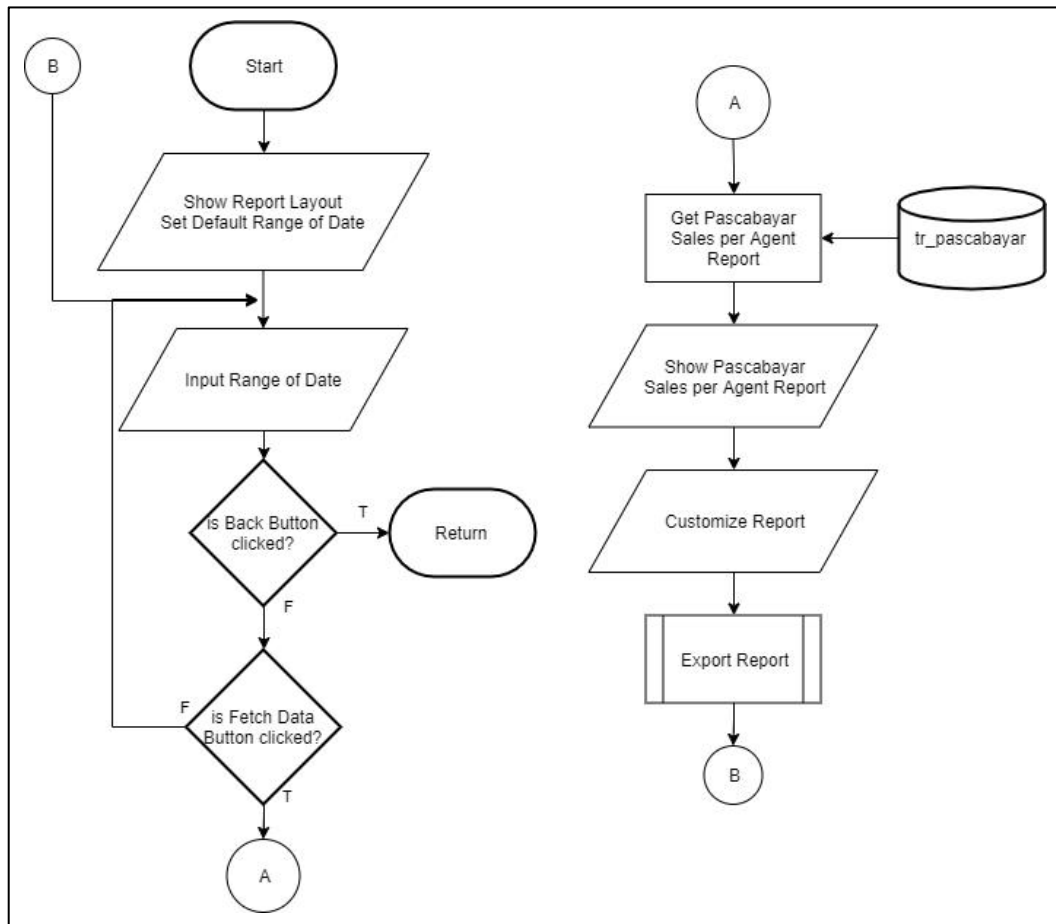


Gambar 3.8 Flowchart Modul Pascabayar Profit Report

2. Flowchart Modul Pascabayar Sales per Agent Report

Modul Pascabayar Sales Per Agent Report pertama kali akan menampilkan tampilan awal dari halaman laporan dan mengeset *Range of Date* sesuai dengan tanggal hari pada saat halaman tersebut diakses. Admin dapat melanjutkan untuk menentukan jarak tanggal yang diinginkan. Jika admin menekan tombol Fetch Data, maka aplikasi akan mengambil laporan pendapatan berdasarkan agen yang dimilikinya dari *tr_pascabayar*. Jika admin menekan tombol Back, maka admin akan diarahkan kembali ke halaman utama dari modul Laporan. Laporan yang telah diambil akan direpresentasikan melalui tabel. Tabel ini kemudian dapat dikostumisasi

mengenai posisi kolom dan atau menghilangkan kolom yang diinginkan. Laporan kemudian akan ditampilkan dan akan dapat diunduh melalui modul Export Report. Gambar 3.9 menunjukkan *flowchart* modul Pascabayar Sales per Agent Report.

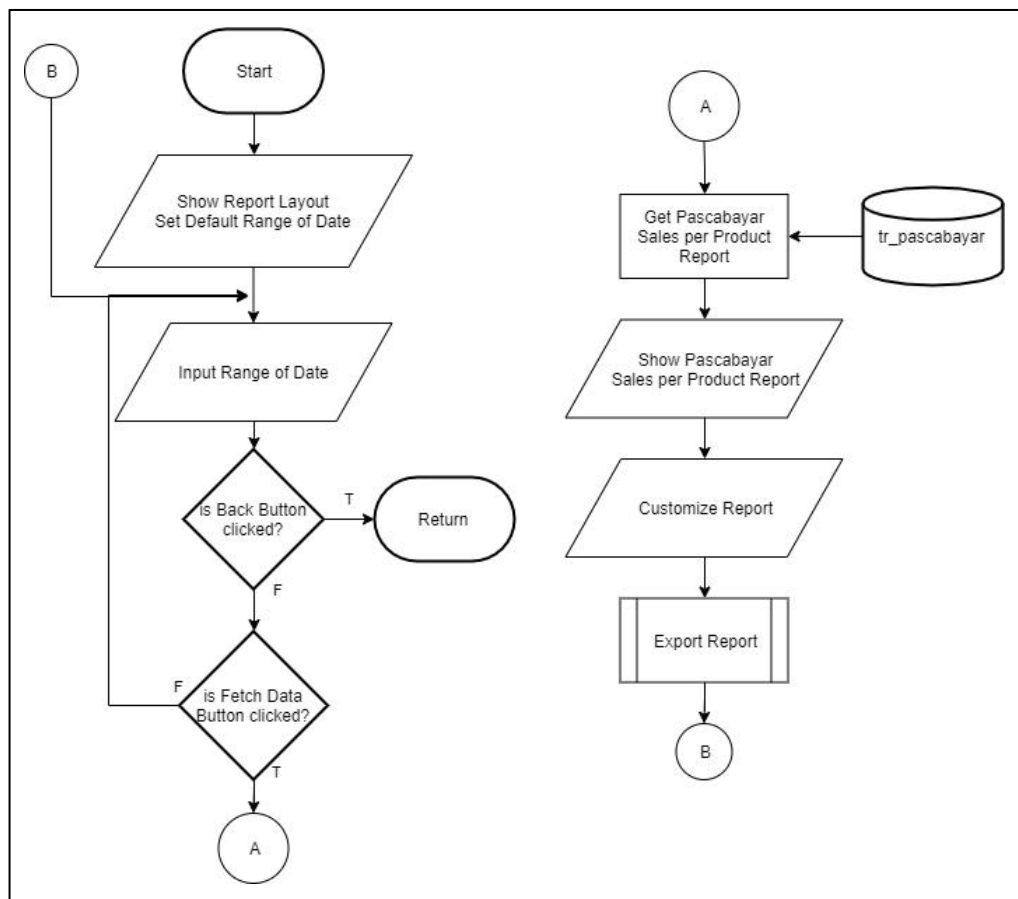


Gambar 3.9 Flowchart Modul Pascabayar Sales per Agent Report

3. Flowchart Modul Pascabayar Sales per Product Report

Modul Pascabayar Sales per Product Report akan pertama kali menampilkan tampilan *layout* laporan dan menetapkan hari pada saat halaman laporan diakes. Admin dapat langsung menekan tombol Fetch Data atau menentukan jarak tanggal pilihannya terlebih dahulu. Proses Get Pascabayar per

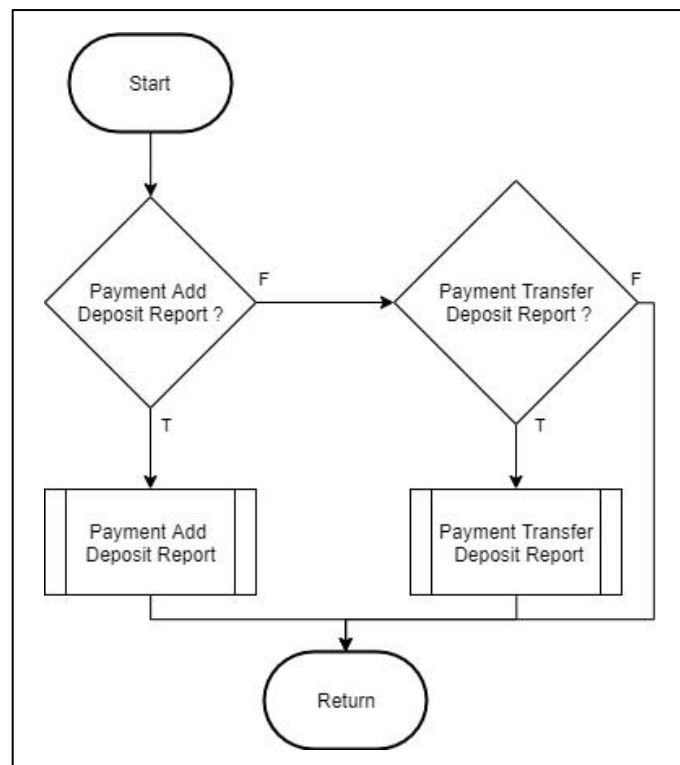
Product akan selanjutnya dilakukan yakni mengambil data transaksi yang ada pada tr_pascabayar. Data yang telah diambil akan ditampilkan melalui tabel. Tabel yang ditampilkan dapat dikostumisasi oleh admin. Kolom dapat dihilangkan sementara dan atau memindahkan kolom. Laporan yang ditampilkan kemudian akan dapat diunduh melalui modul Export Report. Jika admin sudah selesai, maka admin dapat menekan tombol Bank untuk kembali ke halaman utama dari modul Laporan. Gambar 3.10 menunjukkan *flowchart* modul Pascabayar Sales per Product Report.



Gambar 3.10 Flowchart Modul Pascabayar Sales per Product Report

A.2.3 Flowchart Submenu Payment

Submenu *payment* akan mengarahkan admin kepada laporan dalam submenu *payment* yang telah dipilihnya. Pilihan yang tersedia dalam submenu *payment* adalah laporan Add Deposit, dan laporan Transfer Deposit. Gambar 3.11 menunjukkan *flowchart* submenu *payment*.

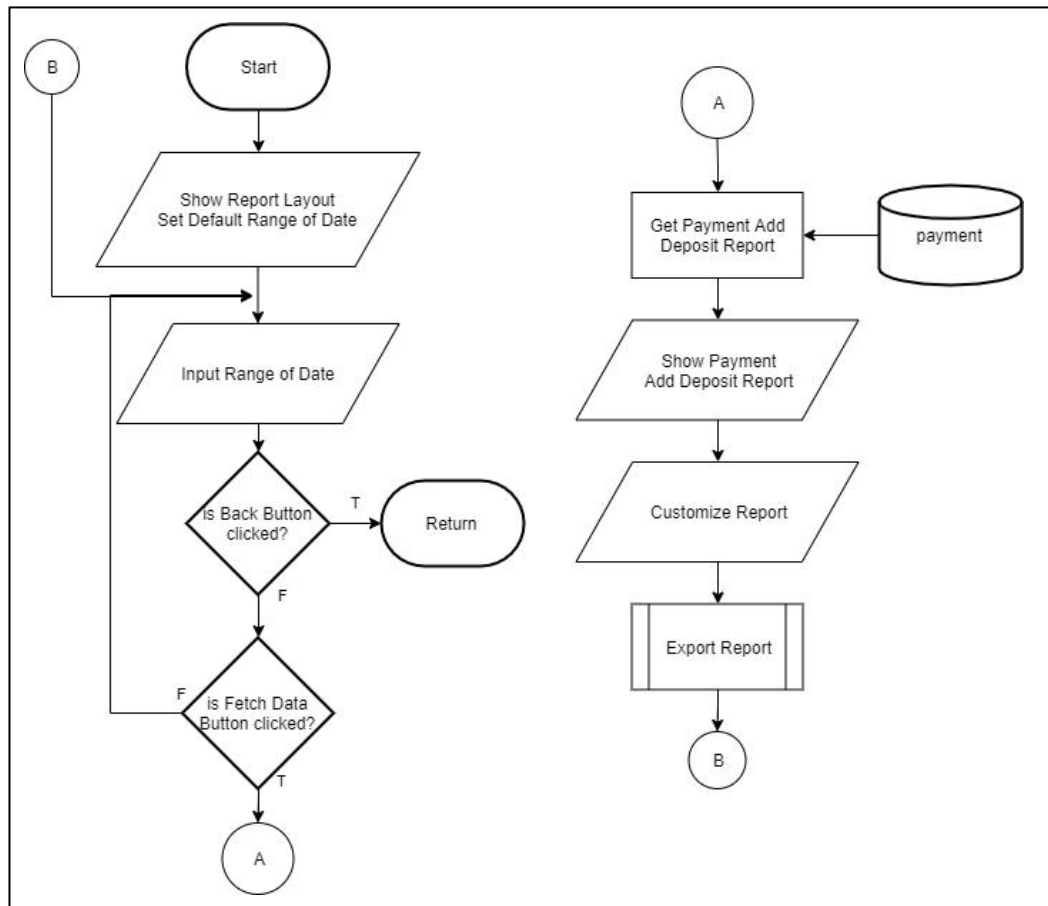


Gambar 3.11 Flowchart Submenu Payment

1. Flowchart Modul Payment Add Deposit Report

Modul Payment Add Deposit bertugas untuk menangani alur kerja dari Payment Add Deposit Report. *Layout* dari halaman laporan akan ditampilkan terlebih dahulu dan mengeset jarak tanggal menjadi tanggal pada hari halaman tersebut dibuka. Kemudian, admin dapat mengeset jarak tanggal yang diinginkan. Jika admin menekan tombol Fetch Data, maka aplikasi akan mengambil data dari

payment. Data yang telah didapatkan akan dapat dikostumisasi urutan kolom atau menghilangkan kolom yang diinginkan. Laporan tersebut juga dapat diunduh melalui Export Report. Gambar 3.12 menunjukan *flowchart* modul Payment Add Deposit Report Report.

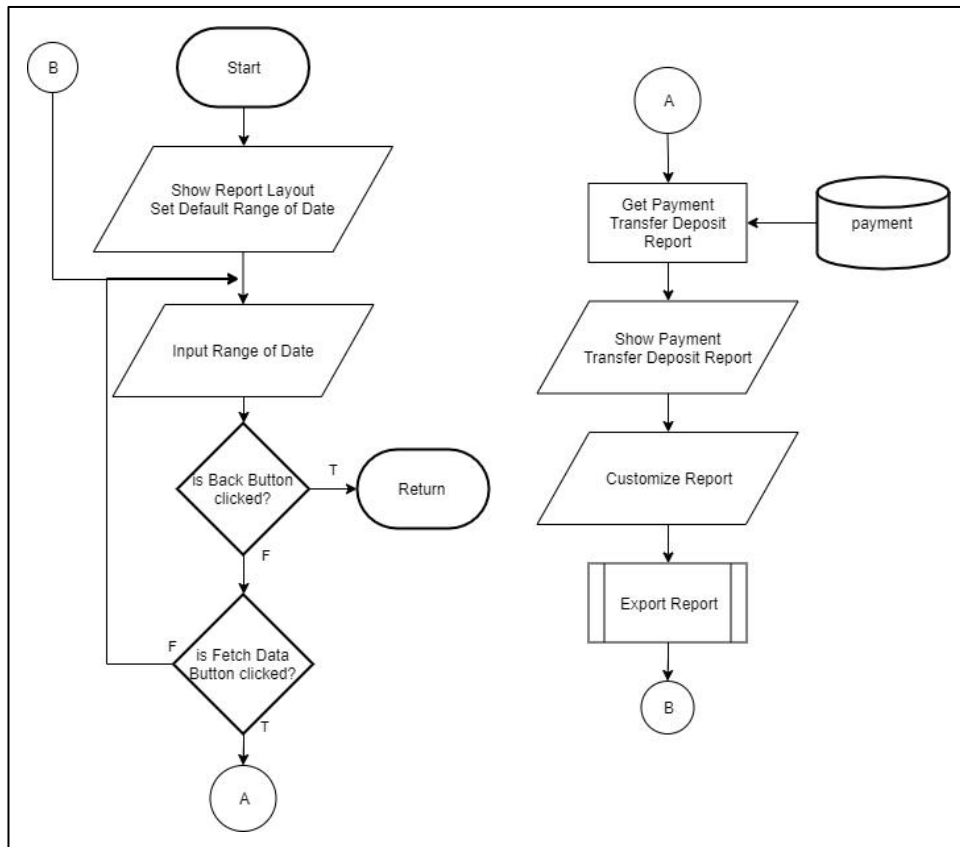


Gambar 3.12 Flowchart Modul Payment Add Deposit Report

2. Flowchart Modul Payment Transfer Deposit Report

Modul Payment Transfer Deposit Report berperan untuk menggambarkan aliran kerja aplikasi. Langkah pertama yang dikerjakan adalah menampilkan *layout* dari halaman laporan beserta mengeset jarak tanggal menjadi tanggal pada saat halaman tersebut diakses. Admin dapat melakukan pergantian

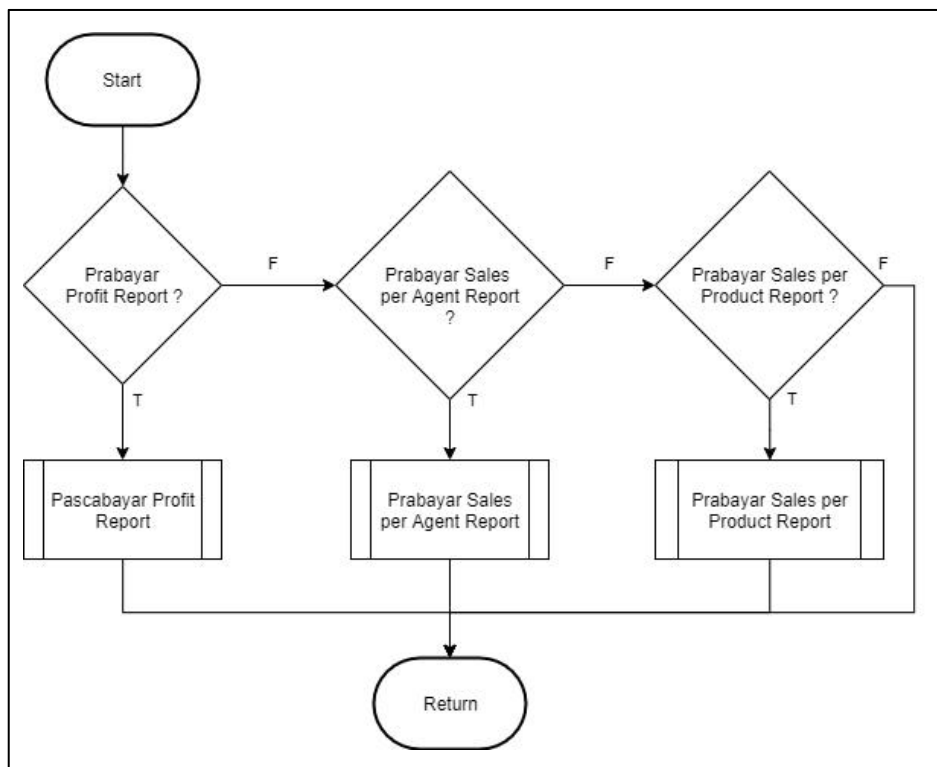
tanggal untuk memilih jarak tanggal yang diinginkan untuk melihat laporan transaksi transfer deposit yang telah dilakukan. Aplikasi akan mengambil data dari payment jika tombol Fetch Data ditekan. Data transaksi ditampilkan melalui tabel yang kemudian dapat dikostumisasi urutan kolom ataupun menghilangkan kolom sesuai yang diinginkan oleh admin. Laporan yang telah dikostumisasi kemudian akan dapat diunduh melalui modul Export Report. Admin dapat kembali ke halaman utama modul laporan. Gambar 3.13 menunjukkan *flowchart* modul Payment Transfer Deposit Report Report.



Gambar 3.13 Flowchart Modul Payment Transfer Deposit Report

A.2.4 Flowchart Submenu Prabayar

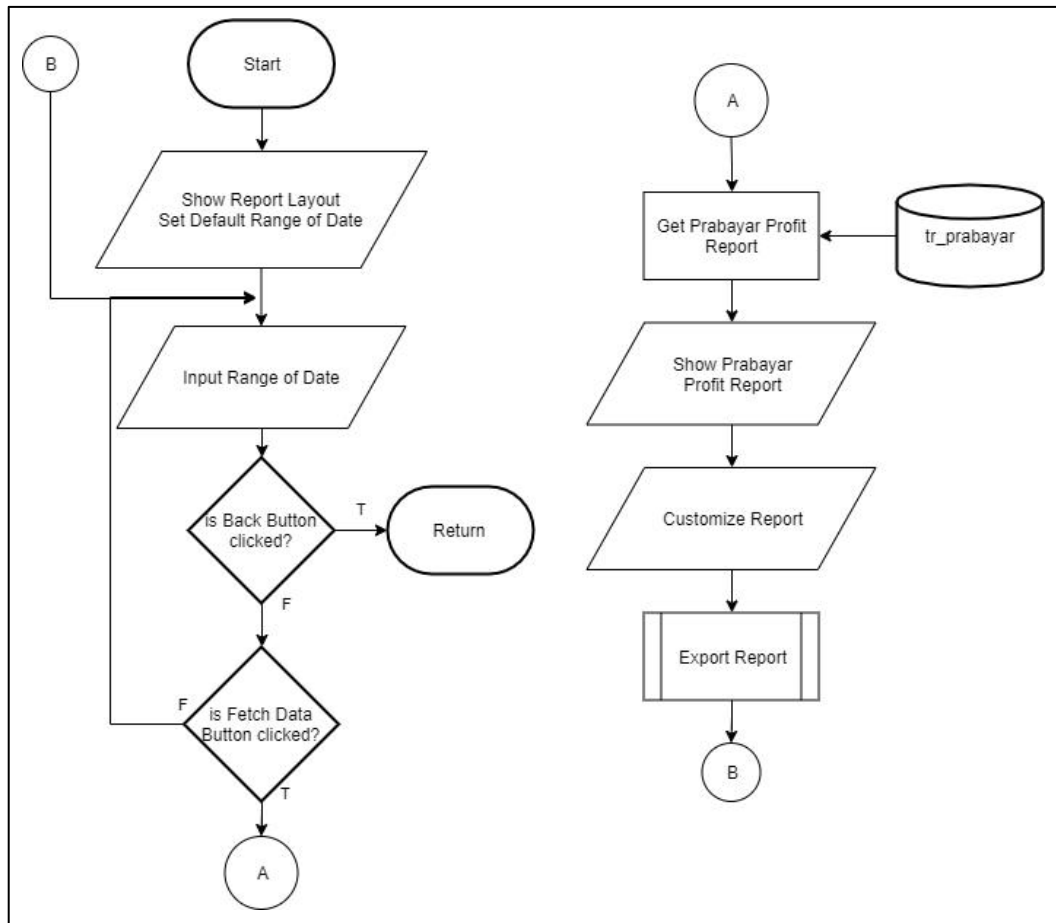
Submenu prabayar bertugas untuk mengarahkan admin menuju halaman laporan yang tersedia pada modul prabayar. Halaman yang tersedia dalam submenu Prabayar adalah laporan Prabayar Profit, laporan Prabayar Sales per Agent, dan laporan Prabayar Sales per Product. Laporan yang ada pada submenu Prabayar identik dengan laporan yang ada pada submenu Pasacabayar. Perbedaan yang ada pada kedua submenu ini adalah lokasi *data store* yang berbeda. Gambar 3.14 menunjukkan *flowchart* submenu Prabayar.



Gambar 3.14 Flowchart Submenu Prabayar

1. Flowchart Modul Prabayar Profit Report

Modul Prabayar Profit Report berfungsi untuk mengatur langkah-langkah yang dibutuhkan untuk mengambil laporan Prabayar Profit. Aplikasi akan menampilkan *layout* utama dari halaman laporan beserta mengatur jarak tanggal menjadi tanggal pada hari dimana halaman ini diakses. Kemudian, admin dapat mengganti jarak tanggal ini dengan cara memberikan masukan yang diinginkan. Jika tombol Fetch Data ditekan, maka aplikasi akan melanjutkannya dengan mengambil data yang ada pada `tr_prabayar`. Transaksi yang didapatkan akan kemudian ditampilkan dalam bentuk tabel. Tabel ini akan dapat dikostumisasi letak kolomnya sesuai dengan yang admin inginkan. Selain letak kolom yang dapat diatur, tabel ini juga memungkinkan untuk admin menghilangkan kolom yang tidak diinginkan. Setelah selesai, admin dapat mengunduh laporan tersebut di modul Export Report. Gambar 3.15 menunjukkan *flowchart* modul Prabayar Profit Report.

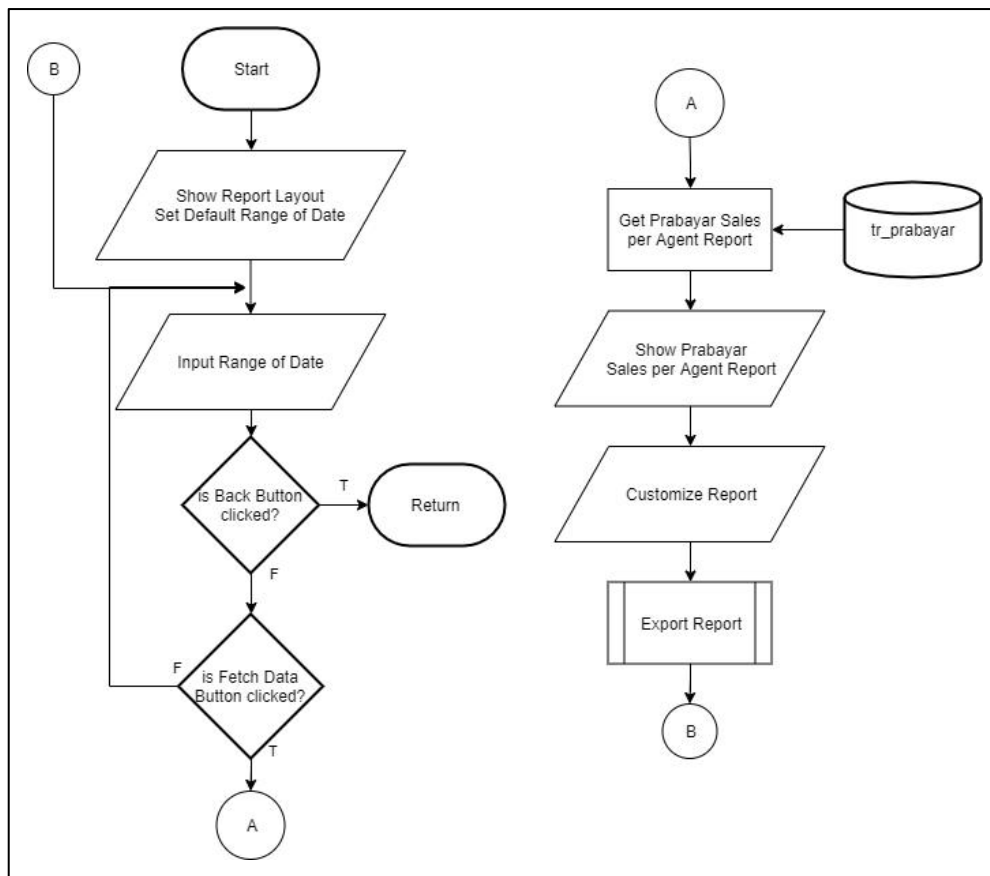


Gambar 3.15 Flowchart Modul Prabayar Profit Report

2. Flowchart Modul Prabayar Sales per Agent Report

Modul Prabayar Sales per Agent Report menunjukkan cara bagaimana laporan Prabayar Sales per Agent didapatkan. Tampilan *layout* laporan dan mengeset jarak tanggal menjadi tanggal pada saat halaman tersebut diakses adalah langkah awal untuk memulai pengambilan data. Selanjutnya, admin dapat mengganti jarak tanggal yang sudah ditetapkan menjadi jarak tanggal yang diinginkannya. Jika admin menekan tombol Fetch Data, maka aplikasi akan melakukan Get Prabayar Sales per Agent Report melalui `tr_prabayar`. Laporan yang didapatkan akan direpresentasikan melalui tabel yang kemudian dapat dikostumisasi. Bagian tabel yang dapat dikostumisasi adalah bagian kolom tabel

dimana admin dapat menghilangkan sebagian dan atau memindahkan kolom untuk menggantikan urutan kolom yang diinginkannya. Laporan juga dapat diunduh selanjutnya pada modul Export Report. Jika sudah selesai, admin dapat menekan tombol Back untuk kembali ke halaman utama modul laporan. Gambar 3.16 menunjukkan *flowchart* modul Prabayar Sales per Agent Report.

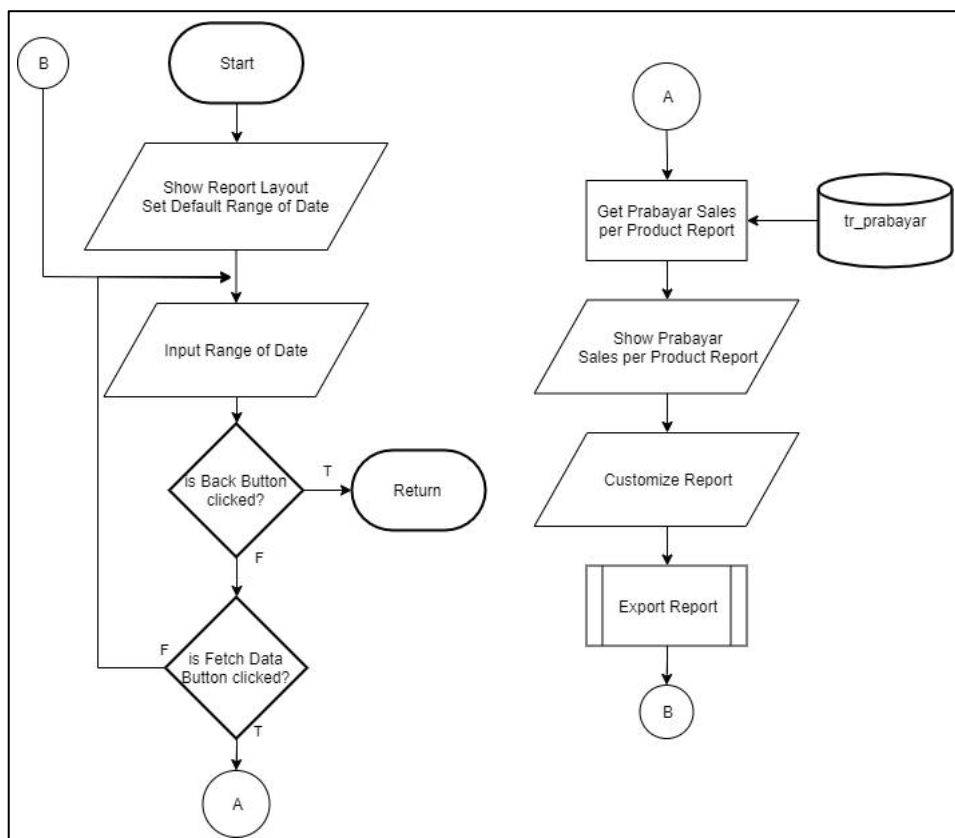


Gambar 3.16 Flowchart Modul Prabayar Sales per Agent Report

3. Flowchart Modul Prabayar Sales per Product Report

Modul Prabayar Sales per Product merupakan bertugas untuk menunjukkan cara Prabayar Sales per Product Report diperoleh. Aplikasi akan

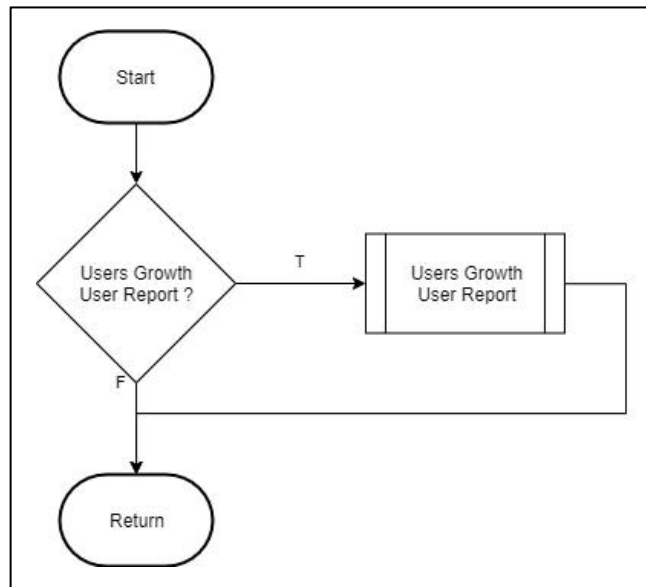
menampilkan *layout* dari modul laporan beserta mengeset jarak tanggal menjadi tanggal pada saat halaman tersebut diakses. Admin dapat mengganti tanggal yang diinginkan. Jika admin menekan tombol Fetch Data, maka aplikasi akan meresponnya dengan melakukan Get Prabayar Sales per Product Report yang ada pada `tr_prabayar`. Laporan akan kemudian ditampilkan melalui tabel. Kolom tabel tersebut akan dapat dikostumisasi dengan cara disembunyikan ataupun diatur urutannya. Laporan ini kemudian akan dapat diunduh melalui modul Export Report dilangkah selanjutnya. Admin dapat kembali ke halaman utama modul laporan dengan cara menekan tombol Back. Gambar 3.17 menunjukkan *flowchart* modul Prabayar Sales per Product Report.



Gambar 3.17 Flowchart Modul Prabayar Sales per Product Report

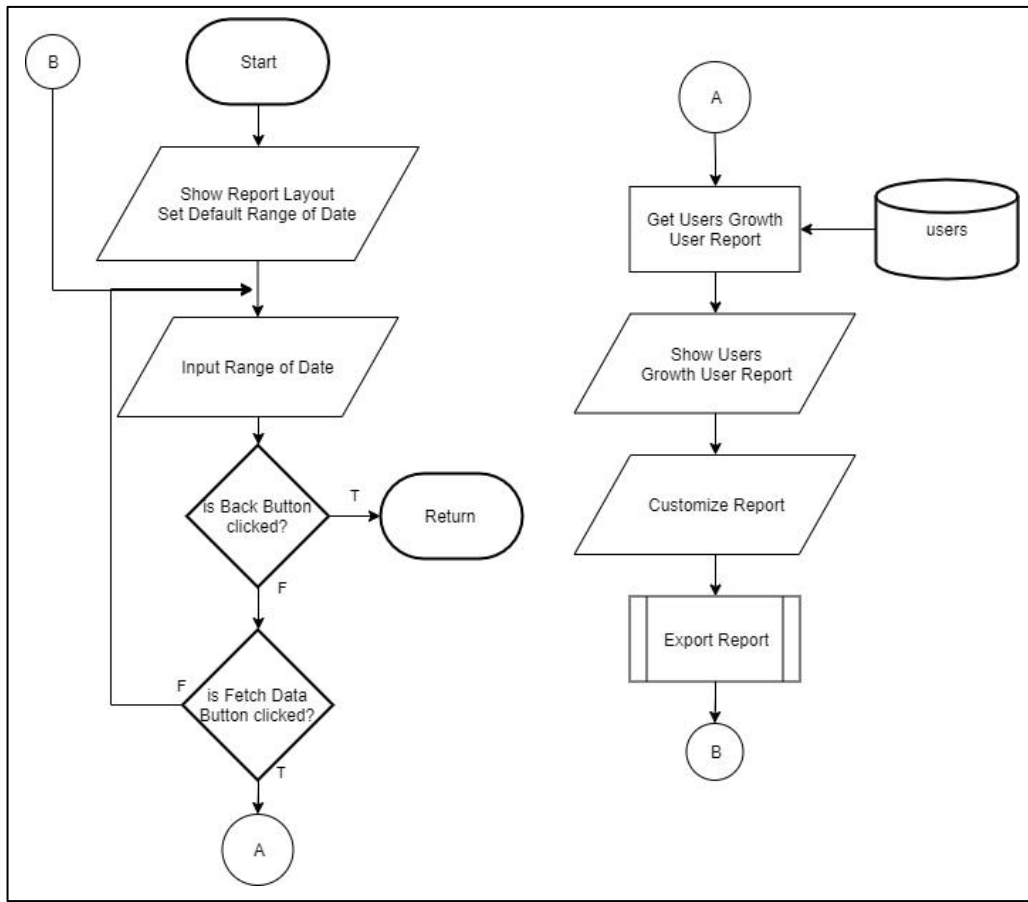
A.2.5 Flowchart Submenu Users

Submenu *users* hanya memiliki satu jenis laporan yakni laporan Growth Users. Oleh karena itu, submodul *users* hanya memiliki satu buah alur sebagaimana gambar 3.18 tunjukan dengan *flowchart* submenu *users*.



Gambar 3.18 Flowchart Submenu Users

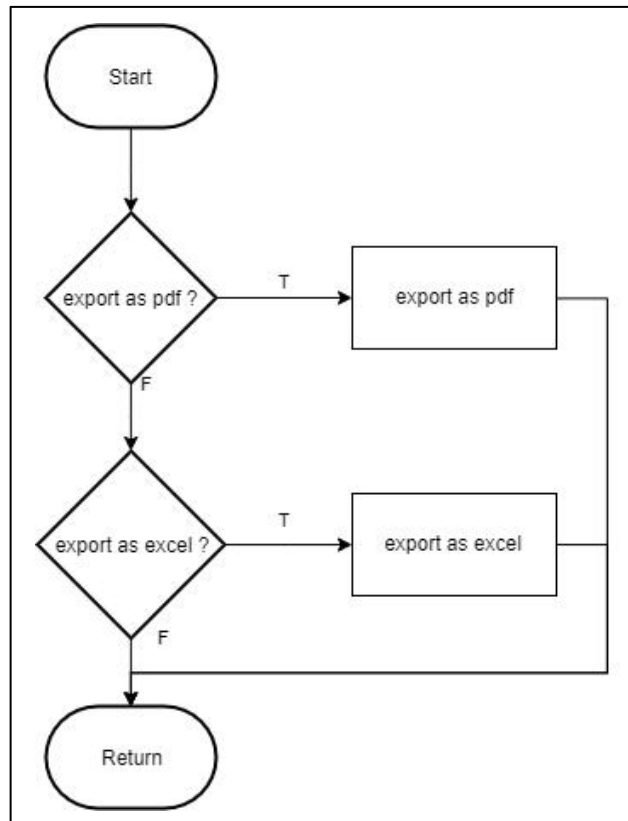
Modul Users Growth User Report akan pertama kali menampilkan tampilan *layout* laporan bersamaan dengan mengeset jarak tanggal secara *default* yaitu dengan hari disaat halaman tersebut diakses. Admin dapat memasukan jarak tanggal yang diinginkan dan dilanjutkan dengan menekan tombol Fetch Data. Aplikasi akan menjalankan proses Get Users Growth User Report yang ada pada users. Laporan akan kemudian ditampilkan melalui tabel. Tabel laporan tersebut akan dapat dikostumisasi sehingga dapat mengganti urutan kolom atau menyembunyikan kolom sesuai yang diinginkan. Laporan dapat diunduh dengan menggunakan modul Export Report yang ada pada langkah selanjutnya. Gambar 3.19 menunjukan *flowchart* modul Users Growth User Report.



Gambar 3.19 Flowchart Modul Users Growth User Report

A.2.6 Flowchart Modul Export Report

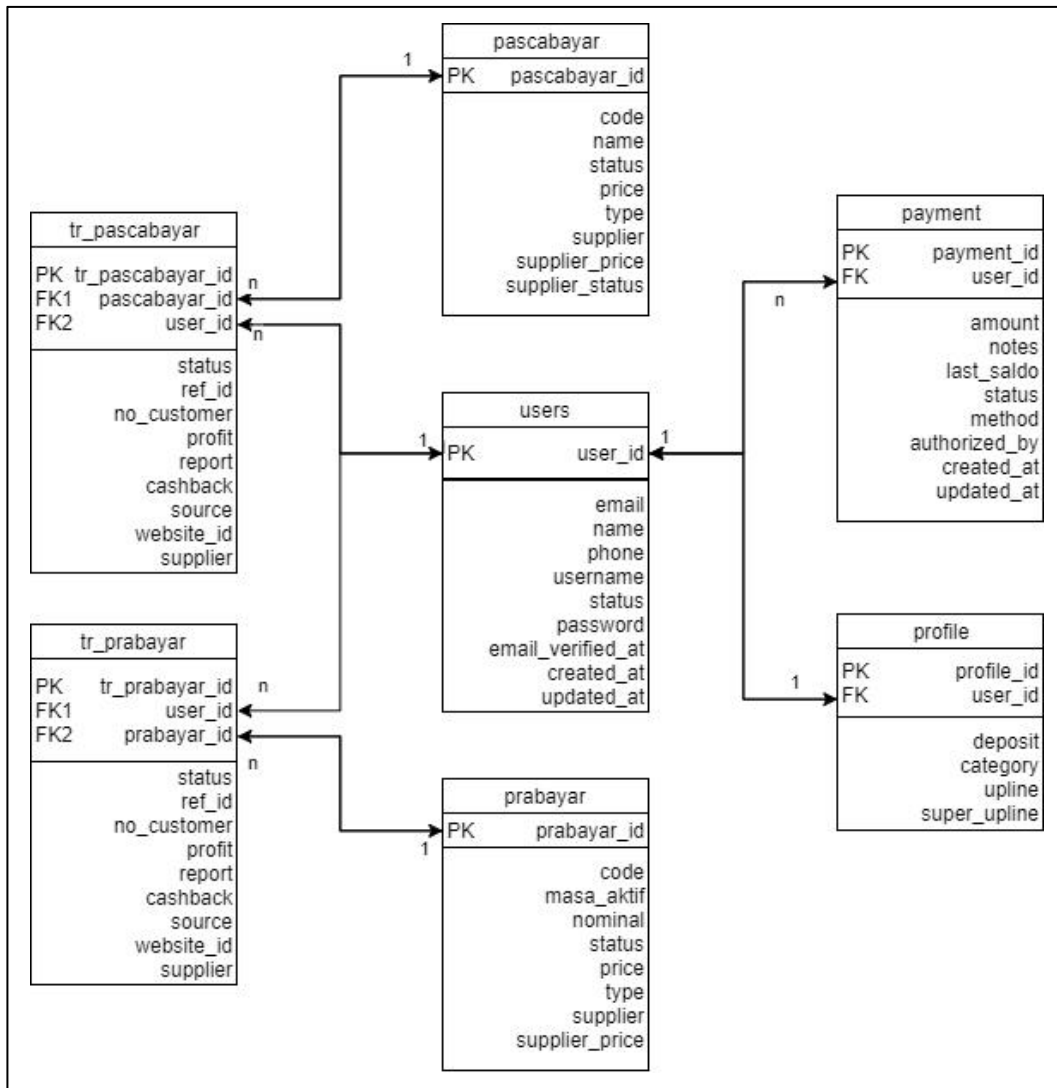
Modul Export Report berfungsi untuk mengunduh laporan yang sedang ditampilkan pada pengguna. Modul ini akan akan mengunduh laporan sebagai pdf jika pengguna memilih untuk mengunduh sebagai pdf dan modul ini juga akan mengecek apakah pengguna beringin untuk mengunduh laporan dalam bentuk *excel*. Gambar 3.20 menunjukkan *flowchart* dari modul Export Report.



Gambar 3.20 Flowchart Modul Export Report

A.3 Database Schema

Berikut adalah visualisasi tabel yang digunakan untuk mengembangkan modul laporan. Tabel yang digunakan adalah tabel tr_pascabayar, tr_prabayar, pascabayar, prabayar, users, dan payment. Tabel-tabel tersebut telah ditentukan oleh perusahaan sehingga untuk mengembangkan aplikasi ini tidak dirancang kembali. Atribut-atribut yang dimiliki disetiap tabel disesuaikan dengan yang dibutuhkan Mobilepulsa. Gambar 3.21 menunjukkan *database schema* dari tabel yang digunakan pada modul laporan



Gambar 3.21 Database Schema dari sebagian Tabel Whitelabel

Entitas *users* dapat mempunyai banyak transaksi pascabayar (*tr_pascabayar*) dan prabayar (*tr_prabayar*). Sehingga dalam sebuah transaksi harus memuat *id* dari satu *users* yang melakukan banyak transaksi, dan satu *id* dari produk yang terkait dari satu transaksi. Produk transaksi disimpan pada tabel prabayar maupun pascabayar. Tabel pascabayar dan prabayar memiliki atribut yang memuat informasi mengenai produk yang ada pada Whitelabel. Entitas *payment* adalah tabel untuk memuat informasi mengenai penambahan ataupun penggunaan deposit *users*. Relasi pada entitas *payment* dengan *users* adalah satu

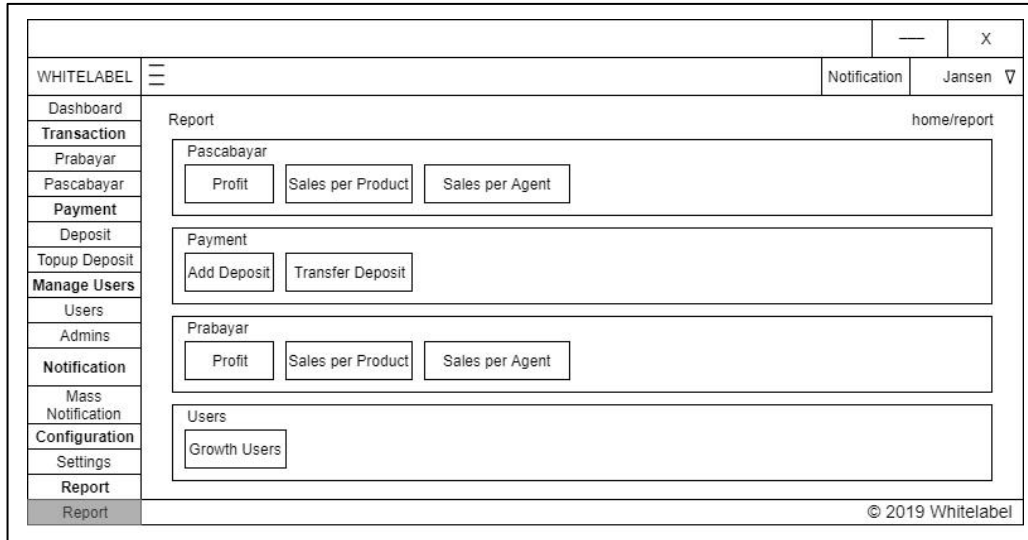
users dapat melakukan banyak transaksi deposit. Entitas *profile* berisikan informasi mengenai *users* seperti jumlah deposit saat ini, kategori *users* seperti agen reguler, *master*, ataupun *super master* beserta menunjukkan atasan dari agen tersebut melalui *upline* dan *super_upline*.

A.4 Rancangan User Interface

A.4.1 Rancangan Halaman Utama

Tampilan akan didasarkan pada rancangan tata letak yang disiapkan. Semua konten yang ada pada semua submenu akan diletakkan dalam *container* konten. *Container* konten terletak dibawah *title* dari halaman yang sedang diakses beserta *breadcrumbs*. Selain itu, terdapat *side bar* yang terdapat seluruh modul yang ada pada sistem Whitelabel. *Navigation bar* memiliki nama Whitelabel beserta admin yang sedang *Log In*. Modul yang sedang aktif akan ditandai dengan warna *background* yang lebih tua. Hal-hal ini semua sudah dikembangkan sebelumnya, sehingga tidak perlu dibuat ulang kembali.

Tata letak yang dibuat untuk modul laporan akan menggunakan tata letak yang sudah disiapkan sebelumnya. Konten yang dimasukkan kedalam *container* konten tersebut akan berupa pilihan laporan yang tersedia. Laporan-laporan tersebut akan dibagi terlebih dahulu berdasarkan asal submenu laporan. Sehingga submenu-submenu yang terbentuk adalah submenu Pascabayar, Payment, Prabayar, dan Users. Laporan yang tersedia dari masing-masing submenu akan ada pada *container* masing-masing submenu. Gambar 3.22 menunjukkan hasil dari tata letak halaman utama modul laporan.



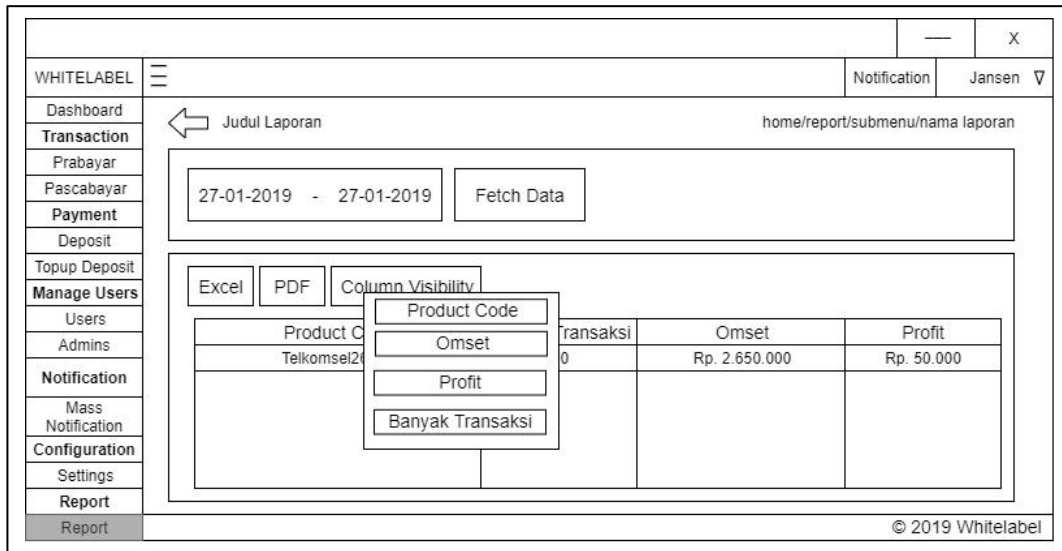
Gambar 3.22 Rancangan Halaman Utama

A4.2 Rancangan Halaman Laporan

Halaman laporan merupakan bagian inti dari modul laporan yang dikembangkan. Setiap laporan memiliki pola yang sama. Sehingga rancangan tata letak halaman setiap laporan adalah sama. Faktor pembeda yang ada diantara laporan adalah judul laporan, *breadcrumbs* halaman laporan, dan kolom yang ditampilkan dalam tabel.

Admin dapat mengganti jarak tanggal yang diinginkan pada tempat yang disediakan. Tombol Fetch Data dapat ditekan untuk memulai pengambilan data sesuai dengan laporan yang dipilih oleh admin sebelumnya. Tabel laporan akan baru ditampilkan ketika data sudah didapatkan. Tombol pdf dan excel juga disediakan untuk mengunduh laporan yang ada. Tombol *column visibility* berfungsi untuk mengatur kolom yang disembunyikan dan menampilkannya kembali. Admin juga dapat mengganti urutan kolom pada tabel dengan cara *drag*

kolom tersebut ketempat yang diinginkan. Gambar 3.23 menunjukkan tata letak halaman laporan.

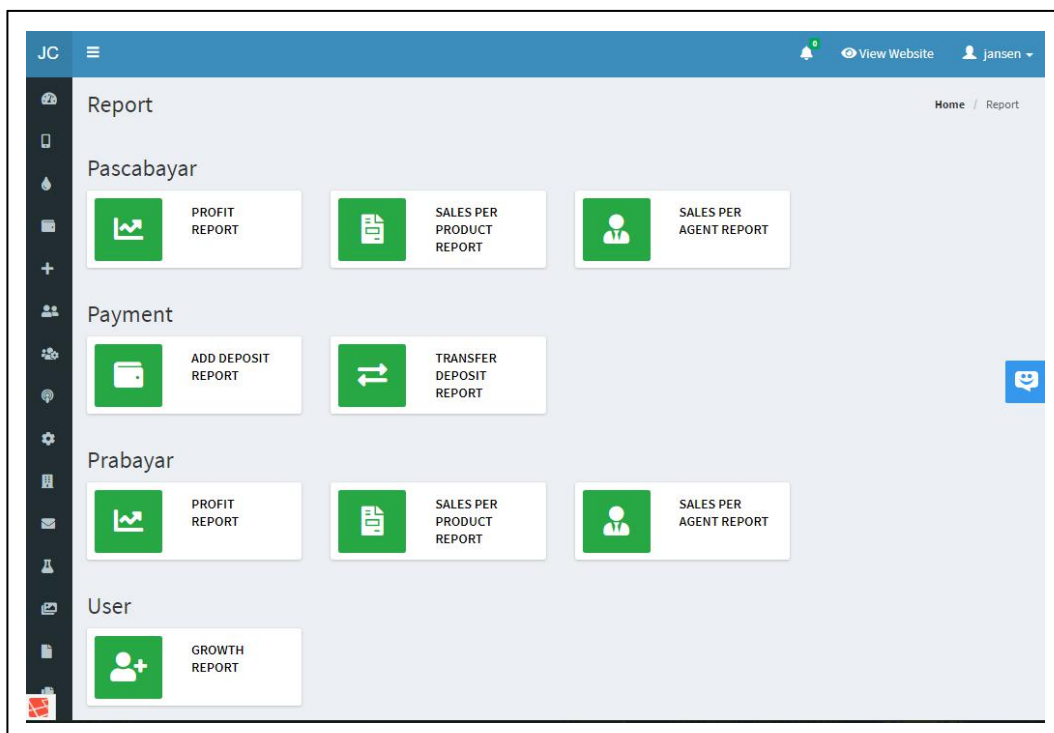


Gambar 3.23 Rancangan Halaman Laporan

B. Implementasi

B.1 Halaman Utama

Halaman utama modul laporan berisikan laporan-laporan yang sudah disediakan disetiap submodulnya. Submodul-submodul yang mempunyai laporan yaitu Pascabayar, Payment, Prabayar, dan User. Admin dapat bernavigasi untuk memilih laporan yang diinginkan. *Sidebar* sudah disediakan sebelumnya sehingga admin dapat bernavigasi kedalam modul lainnya. *Toolbar* yang terdapat diatas halaman mempunyai logo dari masing-masing Whitelabel dan profil dari admin yang sedang *log in* untuk melakukan *log out*. Gambar 3.24 menunjukkan implementasi dari halaman utama modul laporan.

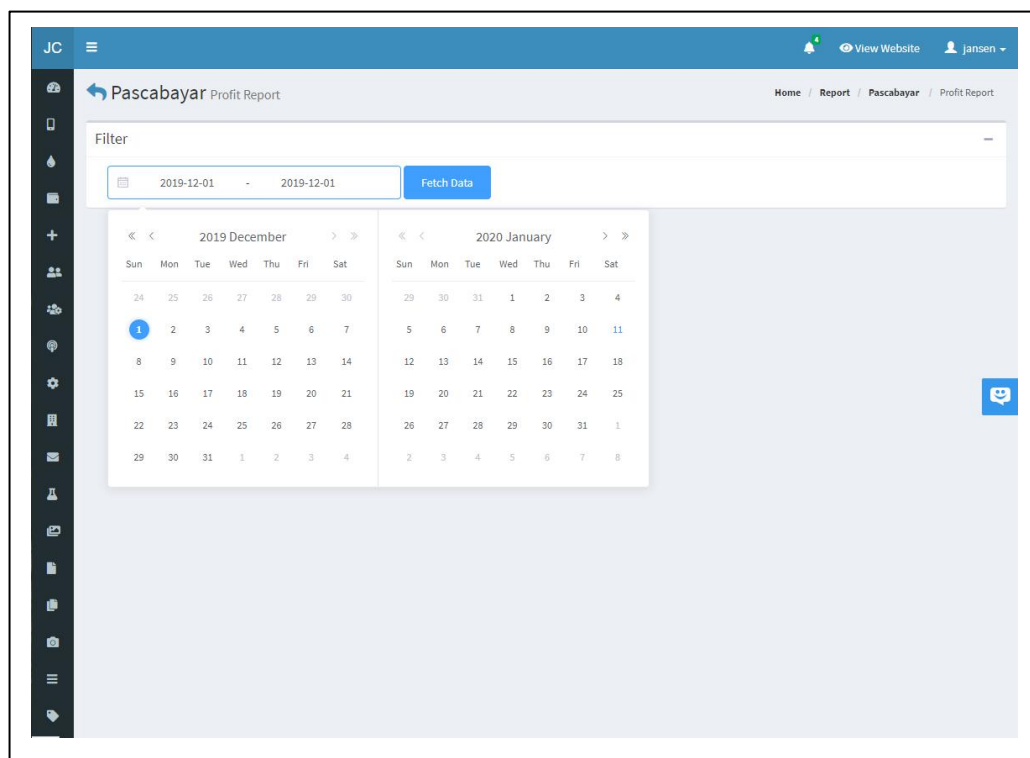


Gambar 3.24 Implementasi Halaman Utama

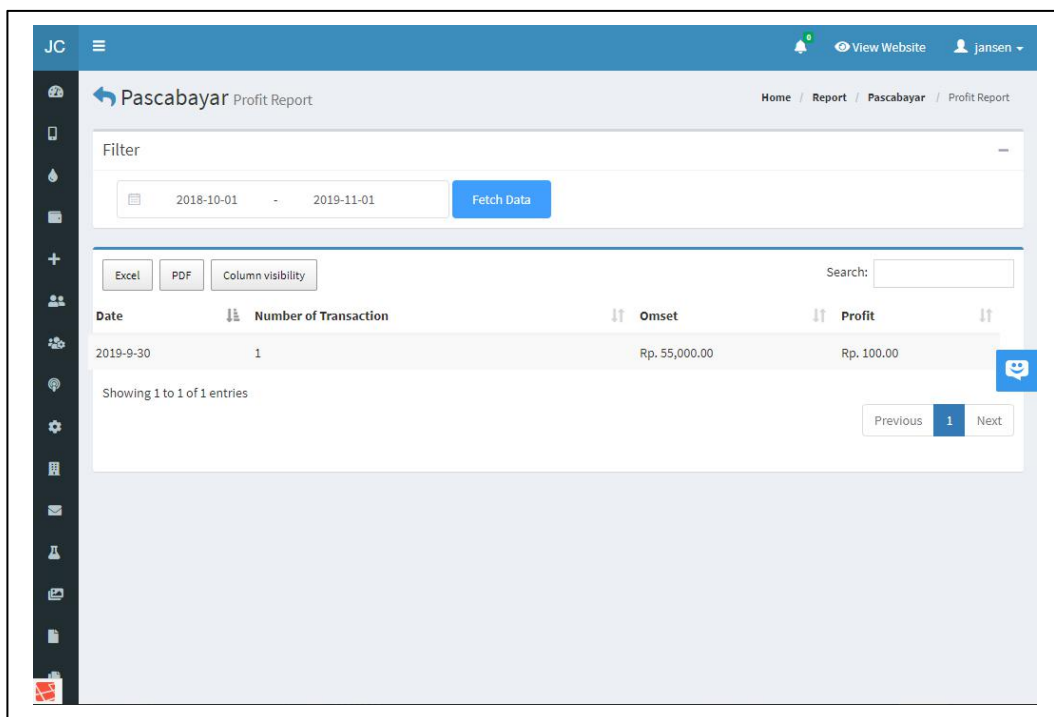
B.2 Halaman Laporan

Halaman laporan untuk masing-masing jenis laporan mempunyai tata letak yang sama. Perbedaan yang ada pada satu laporan dengan laporan yang lain adalah pada tabel yang ditampilkan sebagai hasil dari *query* untuk masing-masing laporan. Implementasi UI *Component* dengan menggunakan Element UI untuk tombol *range of date* dan tombol *fetch data* seperti yang ada pada Gambar 3.25. Tombol *range of date* berfungsi untuk memasukan jarak tanggal yang diinginkan oleh admin dan tombol *fetch data* digunakan untuk memulai pengambilan data berdasarkan jarak tanggal yang sudah di-*input*-kan. Gambar 3.26 menunjukkan halaman laporan pada Pascabayar Profit Report. Sebagai contoh implementasi dari halaman laporan, Pascabayar Profit Report dikelompokkan berdasarkan tanggal transaksi pascabayar untuk menunjukkan hasil penjualan. Laporan yang

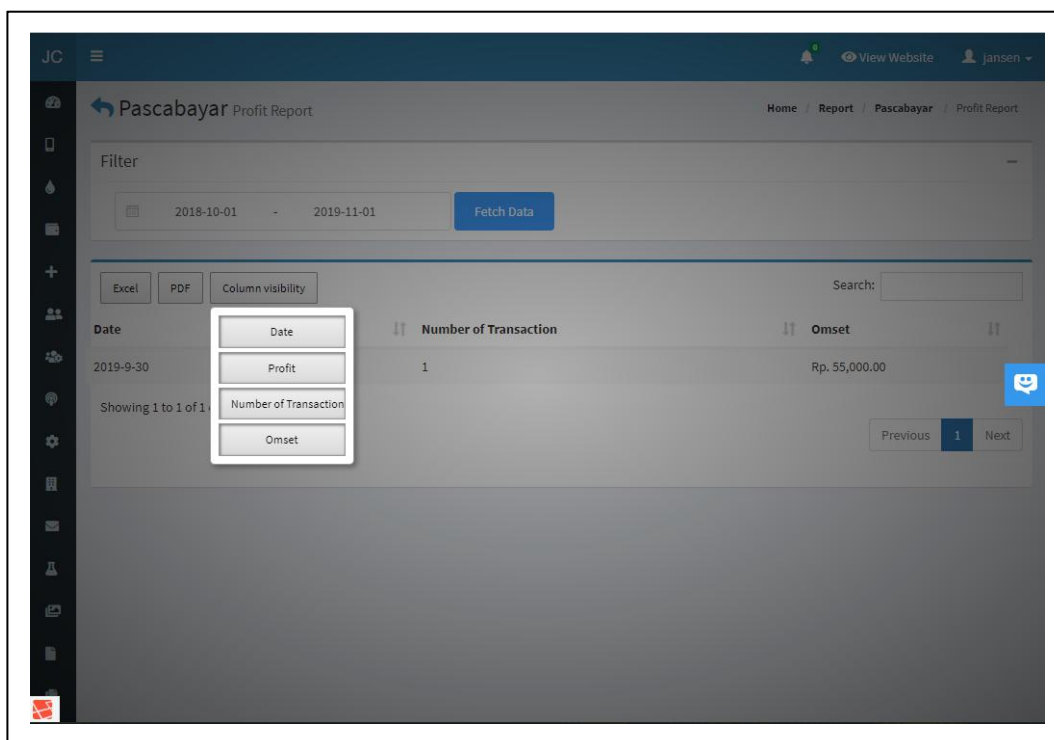
sudah ditampilkan kemudian dapat dilanjutkan untuk dikostumisasi tata letak dari kolom yang ada. Kolom-kolom ini dapat dihilangkan ataupun dipindahkan dengan cara digeser. Gambar 3.27 menunjukkan tombol-tombol untuk menghilangkan kolom. Setelah admin sudah selesai mengkostumisasi urutan kolom dan menyembunyikan kolom yang tidak diinginkan, admin dapat langsung melakukan *export report* menjadi Excel ataupun PDF dengan menekan tombol yang disediakan pada DataTables. Hasil *export report* akan memiliki data yang sudah terkostumisasi kolomnya. Gambar 3.28 dan Gambar 3.29 menunjukkan hasil *export* PDF dan Excel.



Gambar 3.25 Implementasi Element UI



Gambar 3.26 Implementasi halaman laporan pada Pascabayar Profit Report



Gambar 3.27 Tombol-tombol untuk menghilangkan kolom

CMS | jansenc Console
Pascabayar Profit Report

Date	Number of Transaction	Profit
2019-9-30	1	Rp. 100.00
2019-11-25	2	Rp. 3,700.00

Gambar 3.28 Hasil PDF Pascabayar Profit Report

CMS jansenc Console			
Number of Transaction	Omset	Profit	Date
1	Rp. 55,000.00	Rp. 100.00	2019-9-30
2	Rp. 358,750.00	Rp. 3,700.00	2019-11-25

Gambar 3.29 Hasil Excel Pascabayar Profit Report

3.3.3 Kendala yang Ditemukan

Kendala yang didapatkan selama proses kerja magang ada tiga. Kendala yang pertama adalah belum familiar dengan menggunakan *framework* Laravel sehingga memperlambat kerja magang untuk membiasakan diri. Kendala yang kedua adalah penggunaan database MongoDB yang belum pernah dipelajari sebelumnya. Kendala yang ketiga adalah penggunaan *plugin* DataTables dengan tingkat kostumisasi lebih yang belum pernah dipelajari sebelumnya.

3.3.4 Solusi atas Kendala yang Ditemukan

Solusi yang ditemukan untuk mengatasi kendala selama kerja magang adalah dengan membaca dokumentasi yang beredar secara daring melalui *internet*. Dokumentasi tersebut berupa halaman *web* Laravel, MongoDB dan *plugin* DataTables. Solusi yang kedua adalah dengan bertanya kepada rekan kerja pada saat yang tepat dan tidak mengganggu saat kerja. Solusi yang ketiga adalah dengan menonton *video tutorial* yang beredar untuk lebih dapat memahami secara visual.

3.3.5 Pengalaman Penting saat Kerja Magang

Pengalaman yang penting pada saat kerja magang adalah mendapatkan pengalaman secara *real* bekerja disuatu perusahaan. Bekerja disuatu perusahaan berarti harus mengikuti aturan-aturan yang ada dan dapat menyelesaikan tugas yang diberikan tepat waktu sesuai dengan *management* perusahaan sudah tentukan. Hal ini berkaitan dengan metode pengembangan yang dilakukan dalam perusahaan dan manajemen proyek piranti lunak.