

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Analisis Sistem

Visualisasi informasi penjualan *voucher* pulsa menggunakan sistem *dashboard* dirancang untuk membantu pihak Rajawali Reload dalam memonitoring dan memberikan gambaran pencapaian jumlah penjualan *voucher* pulsa elektrik. Sehingga informasi tersebut dapat digunakan untuk melihat hasil penjualan dan juga sebagai dasar dalam memberikan arahan.

##### 3.1.1 Sekilas Mengenai Rajawali Reload

Rajawali Reload Mojokerto merupakan salah satu pemasok pulsa *voucher* telpon elektrik di wilayah Mojokerto dan Lamongan. Rajawali Reload Berdiri sejak 3 juli 2014, hingga saat ini Rajawali Reload sudah memiliki 500 lebih *customer* atau *downline* pulsa di *clouster* Mojokerto dan Lamongan. Rajawali Reload juga menjual berbagai jenis pulsa *voucher all operator* atau *provider*.

Dalam proses bisnis ini, *customer* datang langsung ke kantor Rajawali Reload atau mendatangi sales-salesnya untuk melakukan pendaftaran sebagai *customer*. Setelah melakukan pendaftaran, *customer* akan diminta untuk melakukan *deposit* pulsa minimal Rp 50.000,-. Setelah itu *customer* akan mendapatkan *id* dan *pin* untuk melakukan transaksi jual beli *voucher* pulsa telpon elektrik.

### 3.1.2 Identifikasi Permasalahan

Dalam menjalankan proses bisnisnya sebagai salah satu pemasok *voucher* pulsa elektrik, Rajawali Reload membutuhkan suatu sistem yang mampu untuk membatu penjualan *voucher* pulsa elektrik. Sistem yang dibutuhkan adalah sistem yang dapat memberikan informasi mengenai penjualan *voucher* pulsa elektri yang biasanya menjadi masalah pada Rajawali Reload. Rajawali Reload mengalami kesulitan dalam memperoleh informasi penjualan yang mudah dipahami dan belum mampu memonitoring kinerja penjualan pulsa elektrik dalam mencapai target penjualan yang sudah ditentukan. Sistem yang ada belum bisa memonitoring penjualan pulsa *voucher* dan memberikan informasi yang lebih mudah dibaca dan dimengerti oleh pihak Rajawali Reload. Berikut adalah contoh kondisi penjualan *voucher* pulsa elektrik yang sering terjadi pada Rajawali Reload:

1. Tidak bisa mengetahui apakah transaksi penjualan sudah memenuhi target.
2. Tidak dapat melihat *deposit* harian, mingguan, bulanan apakah sudah memenuhi target penjualan.
3. Kesulitan dalam melihat *voucher provider* yang sering adanya transaksi oleh pelanggan.
4. Tidak dapat melihat *customer* yang melakukan transaksi terbanyak.
5. Tidak dapat melihat *cluster* yang melakukan transaksi terbanyak.

Hal-hal tersebut dapat membuat Rajawali Reload kehilangan potensi *income* dan sering kali tidak dapat memenuhi target penjualan yang telah ditentukan.

### 3.1.3 Analisis Permasalahan

Untuk mengatasi permasalahan tersebut, Rajawali Reload membutuhkan *dashboard* untuk visualisasi penjualan *voucher* pulsa elektrik yang dapat memberikan informasi kondisi penjualan yang sedang terjadi, sehingga dapat membantu dalam memonitor penjualan agar dapat mencapai tujuan yang telah ditetapkan. Berikut adalah contoh solusi penanganan kondisi kritis yang bisa dilakukan *dashboard* penjualan *voucher* pulsa dalam membantu Rajawali Reload menangani permasalahan yang ada:

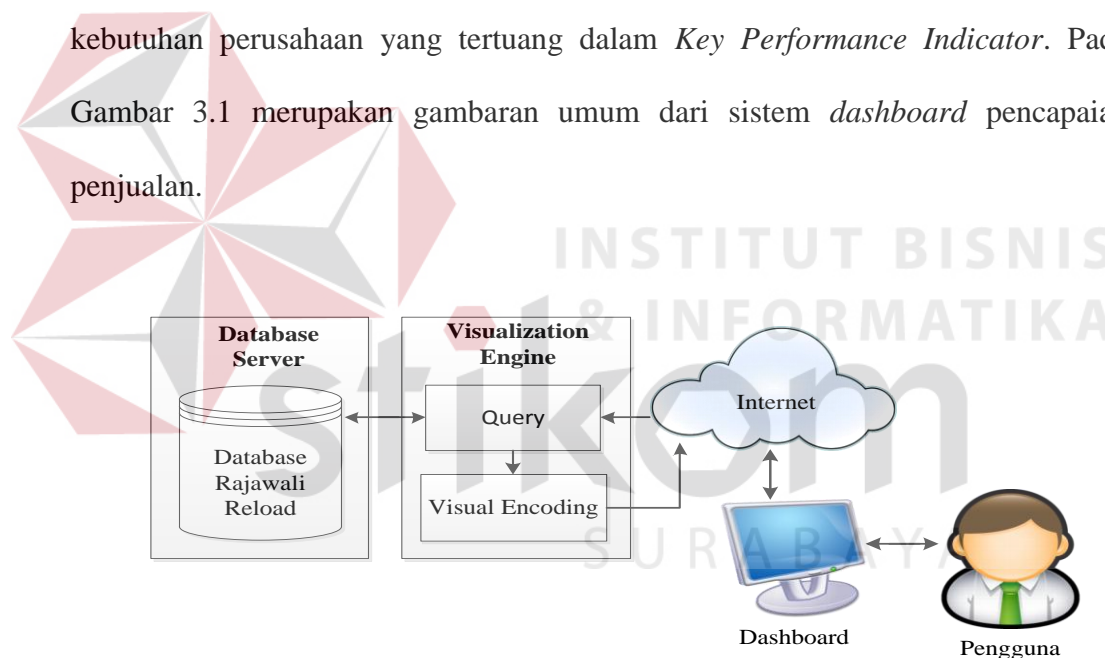
1. Untuk menangani kesulitan dalam melihat transaksi penjualan pulsa sudah terpenuhi apa belum, akan dibuat informasi yang menampilkan transaksi penjualan *voucher* pulsa elektrik.
2. Untuk menangani kesulitan dalam melihat *deposit* penjualan. akan dibuat informasi yang menampilkan *deposit* penjualan ke *customer*.
3. Untuk menangani kesulitan dalam melihat *voucher* per *provider* yang sering dilakukan transaksi oleh pelanggan, akan dibuat informasi yang menampilkan *voucher* per *provider* yang memberikan kontribusi penjualan terbanyak.
4. Untuk dapat melihat *customer* yang melakukan transaksi terbanyak, akan dibuat informasi yang menampilkan *customer* dengan jumlah pembelian terbanyak.
5. Untuk dapat melihat *cluster* yang melakukan transaksi terbanyak, akan dibuat informasi yang menampilkan kota dengan pembelian terbanyak .

Dengan demikian, penggunaan *dashboard* penjualan *voucher* pulsa elektrik diharapkan dapat membantu Rajawali Reload dalam bersaing dengan

pemasok pulsa lainnya dan dapat membantu dalam memonitoring penjualan agar dapat mencapai tujuan yang telah ditetapkan.

### 3.2 Gambaran Umum Sistem

Sistem *dashboard* yang dibangun akan memiliki dua *user* yaitu *Owner* dan *accounting*. Sistem *dashboard* pencapaian penjualan akan digunakan untuk memantau penjualan dalam periode waktu dan dapat digunakan sebagai alat bantu untuk mengambil keputusan. Data yang akan diolah untuk ditampilkan dalam sistem *dashboard* adalah data penjualan. Data penjualan diolah sesuai dengan kebutuhan perusahaan yang tertuang dalam *Key Performance Indicator*. Pada Gambar 3.1 merupakan gambaran umum dari sistem *dashboard* pencapaian penjualan.



Gambar 3.1 Gambaran Umum Sistem

Pada Gambar 3.1 Pengguna berinteraksi dengan sistem melalui tampilan *dashboard* dari layar personal komputer yang sedang terkoneksi dengan *internet*. Hal itu karena *database* penjualan terdapat pada *cpanelserver* yang berbeda tempat dengan pengguna. Jadi pengguna dapat mengaksesnya dimanapun dan

setiap saat asalkan menggunakan personal komputer dan terkoneksi dengan *internet*.

Ketika pengguna berinteraksi dengan *dashboard* penjualan *voucher* pulsa elektrik, sistem memberikan perintah/*task* yang dikirim dari personal computer *dashboard* melalui *internet* dan dieksekusi oleh *visual engine*. Di dalam *visual engine*, *query* berguna untuk mengolah data pada *database* Rajawali Reload. Setelah data diolah oleh *query*, data tersebut dikodekan menjadi gambar/*chart* oleh *visual encoding*. Kemudian data tersebut akan dikirim kembali kepada pengguna melalui *internet* dan ditampilkan pada layar *dashboard* pengguna.

Proses pembuatan sistem *dashboard* akan didasarkan pada beberapa tahapan, pembuatan *dashboard* memperhatikan semua tahapan dalam siklus hidup proyek perangkat lunak. Terdiri atas 4 tahap yaitu *Plan*, *Requirement Gathering*, *Design*, *Build and Validate*, *Deploy*. Pada tahap *Plan* akan ditentukan ruang lingkup dan KPI yang digunakan. Tahap *Requirement Gathering* akan dilakukan wawancara, studi literatur, menentukan stakeholder dan kebutuhannya. Tahap *Design* untuk menentukan sumber data, prototype tampilan, menentukan drill-down, dan menentukan *query*. Tahap *Build and Validate* untuk melakukan pengkodean menerapkan desain antar-muka dan implementasi *query*. Dapat dilihat pada tabel 3.1

Tabel 3.1 Tahapan Penelitian

Tahap 1 (Plan)	Tahap 2 (Requirement Gathering)	Tahap 3 (Design)	Tahap 4 (Build and Validate)
a. Menentukan ruang lingkup b. Menentukan KPI yang digunakan	a. Interview b. Studi literatur c. Menentukan stakeholder d. Menentukan kebutuhan stakeholder	a. Menentukan sumber data b. Prototype tampilan antar-muka c. Menentukan drill-down d. Menentukan query	a. Pengkodean menerapkan desain antar-muka b. Implementasi query

### 3.3 Perancangan Sistem

Dalam melakukan perancangan sistem ada beberapa tahap yang harus dilakukan. Pada tahap ini akan dibuat narasi sistem dan desain arsitektur. Berikut adalah penjelasan secara lengkap tahap-tahap yang dilakukan dalam perancangan sistem.

#### 3.3.1 Analisis Kebutuhan Sistem

Pada tahap ini peneliti melakukan identifikasi informasi mengenai kebutuhan pengguna, seperti informasi apa yang perlu disampaikan, kepada siapa informasi disampaikan, dan seberapa detail informasi tersebut perlu disampaikan.

Tabel 3.2 merupakan analisis kebutuhan pengguna sekaligus rancangan *output* pada sistem ini.

Tabel 3.2 Analisis Kebutuhan

No	Posisi	Kebutuhan	Periode
1	Owner	Melakukan monitoring penjualan a. Tampilan penjualan <i>voucher provider</i> per kategori, berkontribusi b. Tampilan penjualan per <i>customer</i> c. Tampilan penjualan per <i>cluster</i> d. Set target transaksi penjualan dan <i>deposit</i>	Bulanan/Tahun

No	Posisi	Kebutuhan	Periode
2	<i>Accounting</i>	Monitoring target penjualan a. Tampilan target pencapaian transaksi penjualan b. Tampilan target <i>deposit</i>	Harian/Mingguan /Bulanan

### 3.3.2 Identifikasi Parameter Indikator

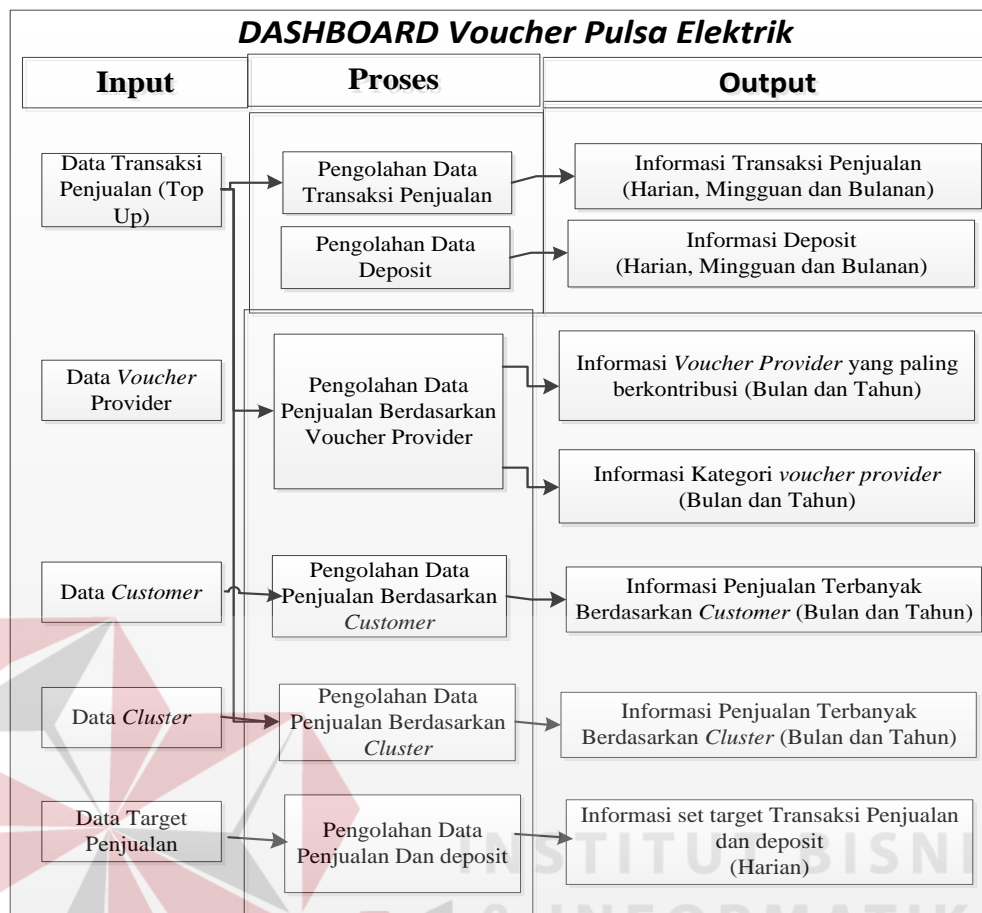
Parameter indikator dibuat agar pengguna dapat mengetahui performa penjualan *voucher* pulsa elektrik Rajawali Reload, apakah dalam keadaan buruk, normal, ataupun bagus. Berdasarkan wawancara dengan pihak Rajawali Reload didapatkan beberapa informasi memiliki parameter indikator tertentu. Informasi yang memiliki parameter indikator tersebut dapat dilihat pada Tabel 3.3.

Tabel 3.3 Nilai Parameter Indikator

No	Nama Parameter	Nilai Standar	Satuan
1.	Transaksi penjualan <i>voucher</i> harian	1.000	Transaksi
2.	Transaksi Penjualan <i>voucher</i> mingguan	7.000	Transaksi
3.	Transaksi penjualan <i>voucher</i> bulanan	30.000	Transaksi
4.	<i>Deposit</i> harian	10.000.000	Rupiah
5	<i>Deposit</i> mingguan	70.000.000	Rupiah
6	<i>Deposit</i> bulanan	300.000.000	Rupiah
7	Transaksi penjualan <i>voucher cluster</i> Mojokerto	250	Transaksi
8	Transaksi penjualan <i>voucher cluster</i> Lamongan	750	Transaksi
9	<i>Deposit cluster</i> mojokerto bulanan	75.000.000	Rupiah
10	<i>Deposit cluster</i> lamongan bulanan	225.000.000	Rupiah

### 3.3.3 Input, Proses, dan Output

*Input*, proses, dan *output* sistem *dashboard* penjualan *voucher* pulsa elektrik dapat dilihat pada Gambar 3.2.



Gambar 3.2 IPO Diagram Dashboard

Pada Gambar 3.2 di atas menjelaskan gambaran umum proses dashboard berupa I.P.O diagram yang menggambarkan *input*, proses dan *output* sebagai berikut:

### 1. *Input*

Pada bagian *input*, terdapat data-data yang digunakan sebagai masukan diantaranya :

- a. Data transaksi penjualan ( *Top Up* )

Tabel yang menyimpan seluruh data mengenai transaksi penjualan.

- b. Data *voucher provider*

Tabel yang menyimpan data *voucher provider*.



c. Data *customer*

Tabel yang menyimpan data *customer*.

d. Data *cluster*

Tabel yang menyimpan data *cluster*.

e. Data Target Penjualan

Tabel yang menyimpan data target transaksi penjualan dan *deposit*.

## 2. Proses

Data masukan di atas akan diolah dan diproses yang nantinya akan menghasilkan suatu keluaran atau *output*, adapun proses tersebut yaitu:

a. Proses pengolahan data transaksi penjualan

Mengolah tabel transaksi penjualan berdasarkan harian, mingguan dan bulanan.

b. Proses pengolahan data *deposit*

Mengolah tabel *deposit* penjualan berdasarkan harian, mingguan dan bulanan.

c. Proses pengolahan data *voucher provider*

Mengolah tabel *voucher* dan tabel transaksi penjualan berdasarkan bulanan dan tahunan.

d. Proses pengolahan data *customer*

Mengolah tabel *customer* dan tabel *deposit* berdasarkan bulanan dan tahunan.

e. Proses pengolahan data *cluster*

Mengolah tabel *cluster* dan tabel transaksi penjualan berdasarkan bulanan dan tahunan.

f. Proses pengolahan data target penjualan

Mengolah tabel target penjualan (transaksi dan *deposit*) untuk membandingkan dengan parameter yang sudah ada.

### 3. *Output*

Dari hasil proses yang telah dilakukan maka menghasilkan beberapa *output* sesuai level *dashboard* yang akan ditampilkan :

a. Informasi transaksi penjualan (harian, mingguan, dan bulanan)

Informasi transaksi penjualan dalam tujuh hari terakhir, lima minggu terakhir, dan tiga bulan terakhir.

b. Informasi *deposit* (harian, mingguan, dan bulanan)

Informasi *deposit* dalam tujuh hari terakhir, lima minggu terakhir, dan tiga bulan terakhir.

c. Informasi *voucher provider* yang paling berkontribusi (bulanan dan tahunan)

Informasi *voucher provider* yang memberikan kontribusi penjualan terbanyak pada bulan ini dan tahun ini.

d. Informasi kategori *voucher provider* (bulan dan tahun)

Informasi *voucher provider* (dilihat dari histori transaksi penjualan) pada bulan ini dan tahun ini.

e. Informasi penjualan terbanyak berdasarkan *customer* (bulan dan tahun)

Informasi penjualan terbanyak yang dikelompokkan berdasarkan *customer* (dilihat dari histori transaksi penjualan) pada bulan ini dan tahun ini.

f. Informasi penjualan terbanyak berdasarkan *cluster* (bulan dan tahun)

Informasi penjualan terbanyak yang dikelompokkan berdasarkan *cluster* (dilihat dari histori transaksi penjualan) pada bulan ini dan tahun ini.

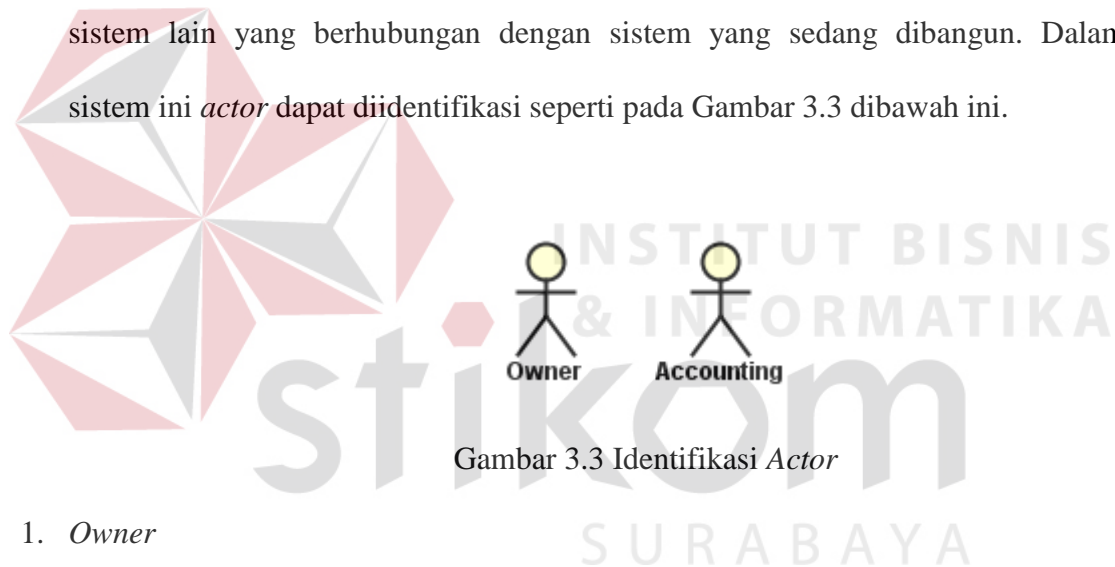
g. Informasi set target transaksi penjualan dan *deposit*

Informasi transaksi penjualan dan *deposit* untuk membandingkan dengan parameter yang ada untuk menampilkan informasi kinerja penjualan *voucher* pulsa.

### 3.3.4 UML

#### a. Mengidentifikasi *Actor*

*Actor* menggambarkan seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun. Ada dua tipe *actor* yaitu: pengguna sistem dan sistem lain yang berhubungan dengan sistem yang sedang dibangun. Dalam sistem ini *actor* dapat diidentifikasi seperti pada Gambar 3.3 dibawah ini.



Gambar 3.3 Identifikasi *Actor*

#### 1. *Owner*

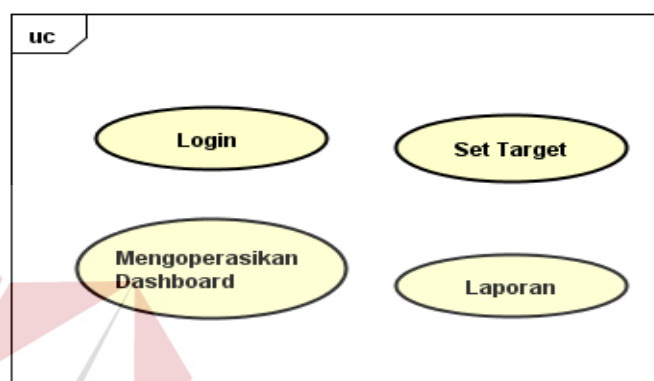
*Owner* merupakan pemilik Rajawali Reload yang menjalankan aplikasi *dashboard* untuk melihat tampilan penjualan *voucher provider* perkategori, penjualan yang *voucher* yang berkontribusi, penjualan berdasarkan *customer*, Tampilan penjualan berdasarkan *cluster*.

#### 2. *Accounting*

*Accounting* merupakan bagian keuangan yang menjalankan aplikasi *dashsboard* Rajawali Reload untuk melihat tampilan target pencapaian transaksi penjualan dan tampilan target *deposit*.

## b. Use Case

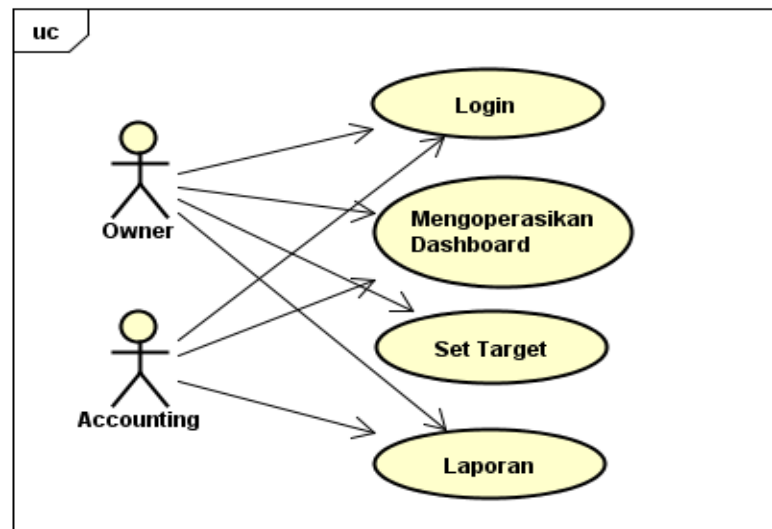
*Use case* adalah bagian tingkat tinggi dan fungsional sistem. Dengan kata lain, *use case* menggambarkan bagaimana seseorang menggunakan sistem (Sholiq, 2006). Berikut ini merupakan *use case* yang telah didapat berdasarkan kebutuhan sistem yang dapat dilihat pada Gambar 3.4



Gambar 3.4 Identifikasi *Use Case Dashboard*.

## c. Use Case Diagram

*Use Case Diagram* menyajikan interaksi antara *Use Case* dan *Actor* (Sholiq, 2006). *Use Case* dan *Actor* menggambarkan ruang lingkup sistem yang sedang dibangun. *Use Case* meliputi semua yang ada di dalam sistem, sedangkan *actor* meliputi semua yang ada diluar sistem. *Use Case Diagram* dapat dilihat pada Gambar 3.5

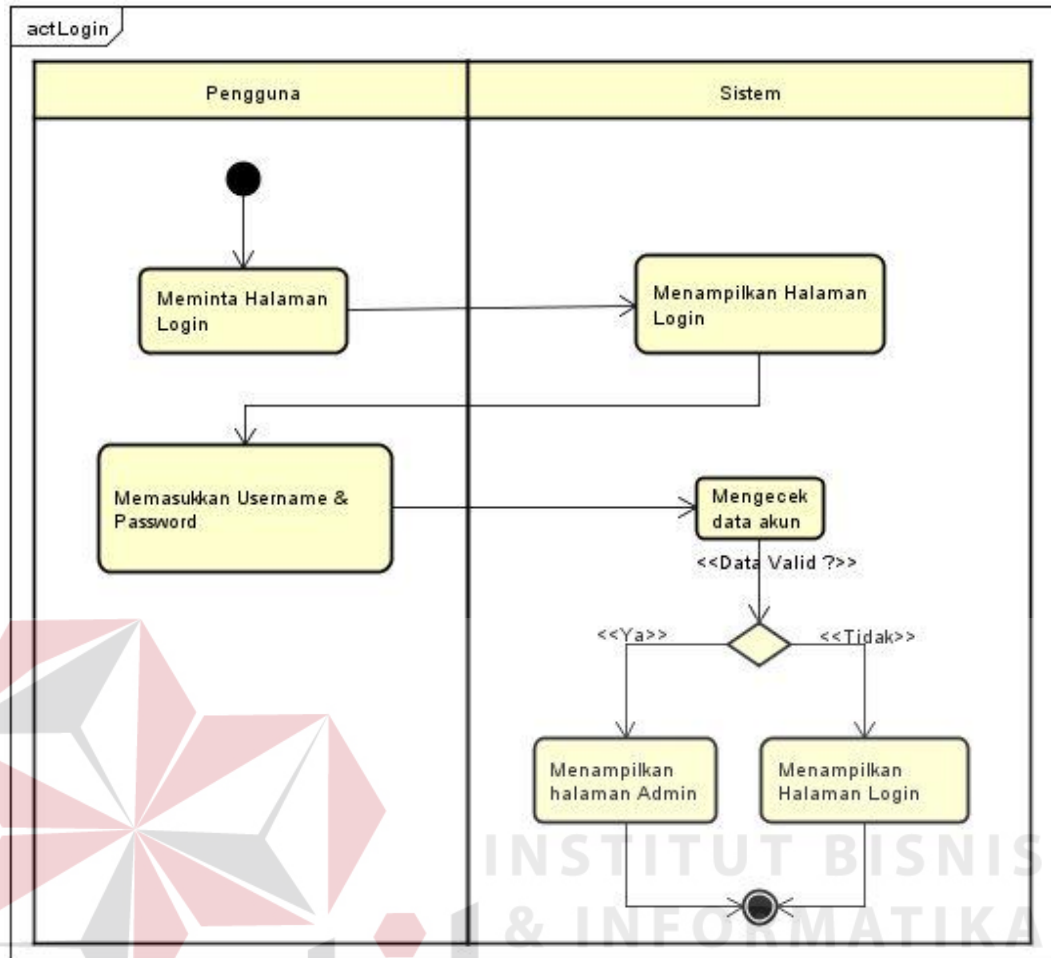


Gambar 3.5 Use Case Diagram Dashboard

Pada Gambar 3.5 diatas dapat dilihat bahwa *owner* dan *accounting* dapat melakukan login sesuai jabatannya dan mengoperasikan halaman *dashboard* sesuai dengan kebutuhan.

#### d. Diagram Aktivitas *Login*

Desain diagram aktivitas *login* beserta penjelasannya dapat dilihat pada Gambar 3.6.

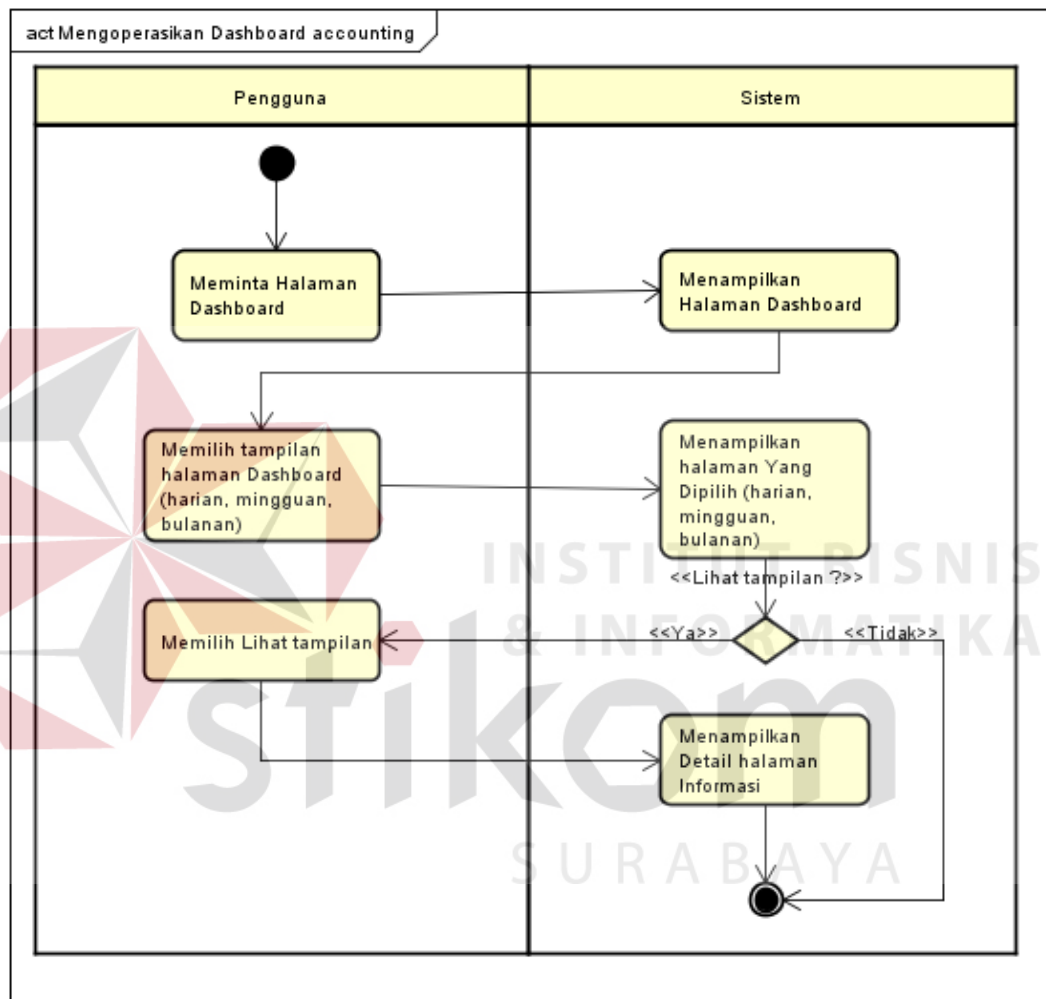


Gambar 3.6 Diagram Aktifitas *Login*

Diagram aktifitas *login* dilakukan oleh semua pengguna. Proses dimulai ketika pengguna mengakses halaman *login* dengan cara memasukkan alamat halaman *login*. Sistem akan menampilkan *form login* dan pengguna memasukkan *username* dan *password*. Kemudian sistem akan mengecek apakah *username* dan *password* tersebut sama dengan yang ada di dalam *database* akun. Apabila data sesuai dengan *database* akun maka sistem akan memberikan notifikasi bahwa *login* sukses. Setelah *login* berhasil maka sistem akan menampilkan halaman *dashboard*.

e. **Diagram Aktivitas Mengoperasikan *Dashboard Accounting***

Desain diagram aktifitas mengoperasikan tampilan *dashboard accounting* beserta penjelasannya dapat dilihat pada Gambar 3.7.

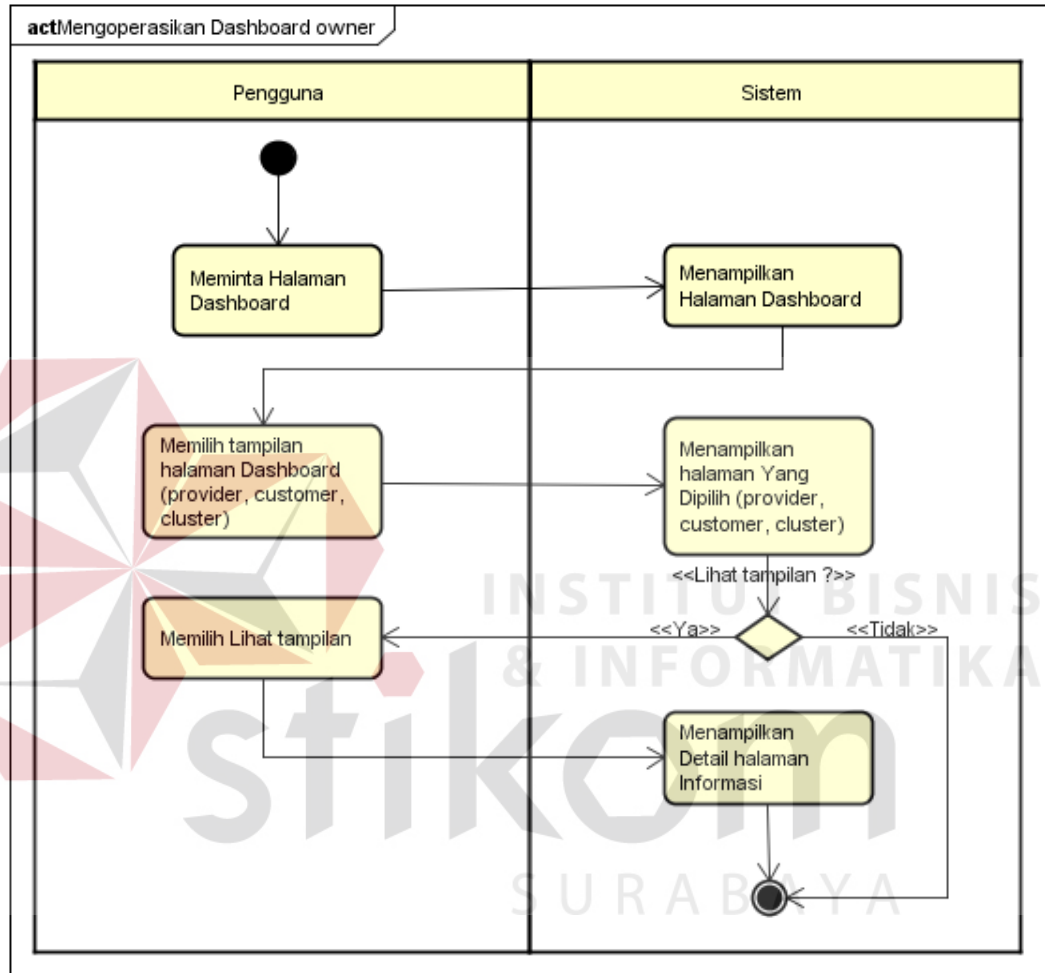


Gambar 3.7 Diagram Aktivitas Mengoperasikan *Dashboard accounting*

Pengguna meminta halaman dengan *login* terlebih dahulu, setelah itu pengguna dapat melihat tampilan *dashboard*. Kemudian pengguna dapat memilih informasi yang ingin ditampilkan pada *dashboard* maka sistem akan menampilkan informasi tersebut.

#### f. Diagram Aktivitas Mengoperasikan *Dashboard Owner*

Desain diagram aktifitas mengoperasikan tampilan *dashboard* owner beserta penjelasannya dapat dilihat pada Gambar 3.8.



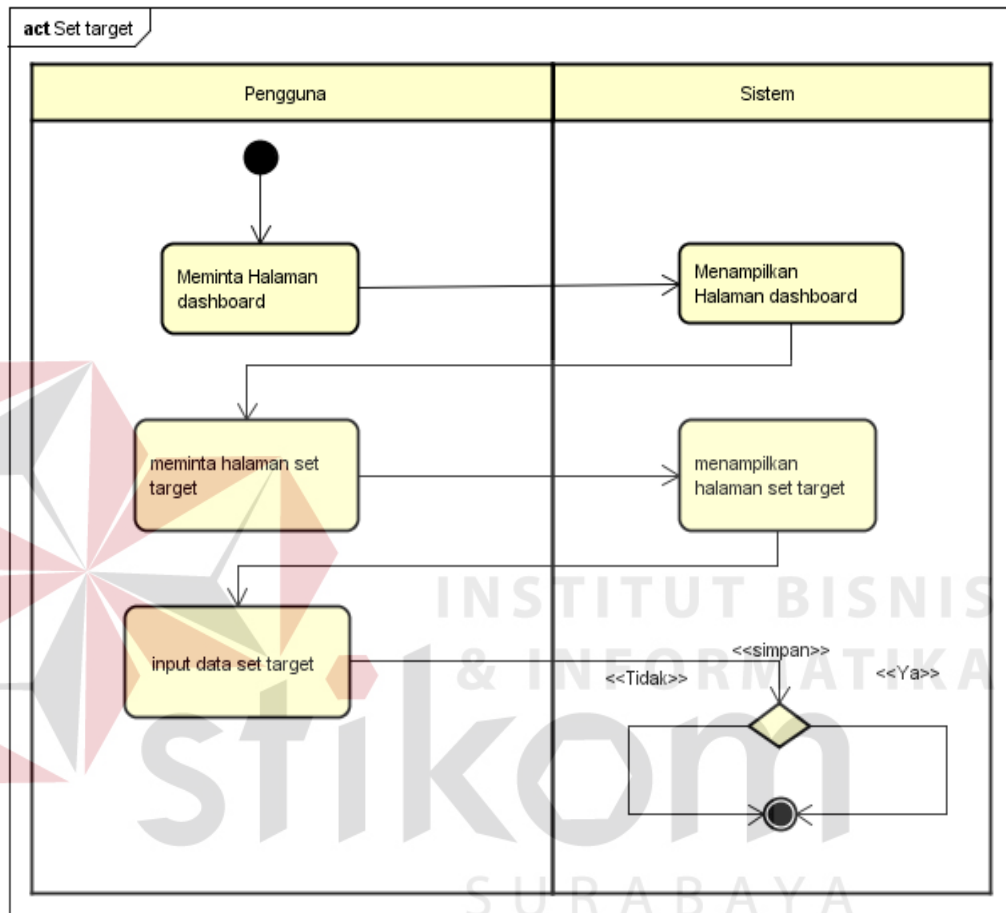
Gambar 3.8 Diagram Aktivitas Mengoperasikan *Dashboard Owner*.

Pengguna meminta halaman dengan *login* terlebih dahulu, setelah itu pengguna dapat melihat tampilan *dashboard*. Kemudian pengguna dapat memilih informasi yang ingin ditampilkan pada *dashboard* maka sistem akan menampilkan informasi tersebut.



### g. Diagram Aktifitas Set Target

Desain diagram aktifitas set target beserta penjelasannya dapat dilihat pada Gambar 3.9.

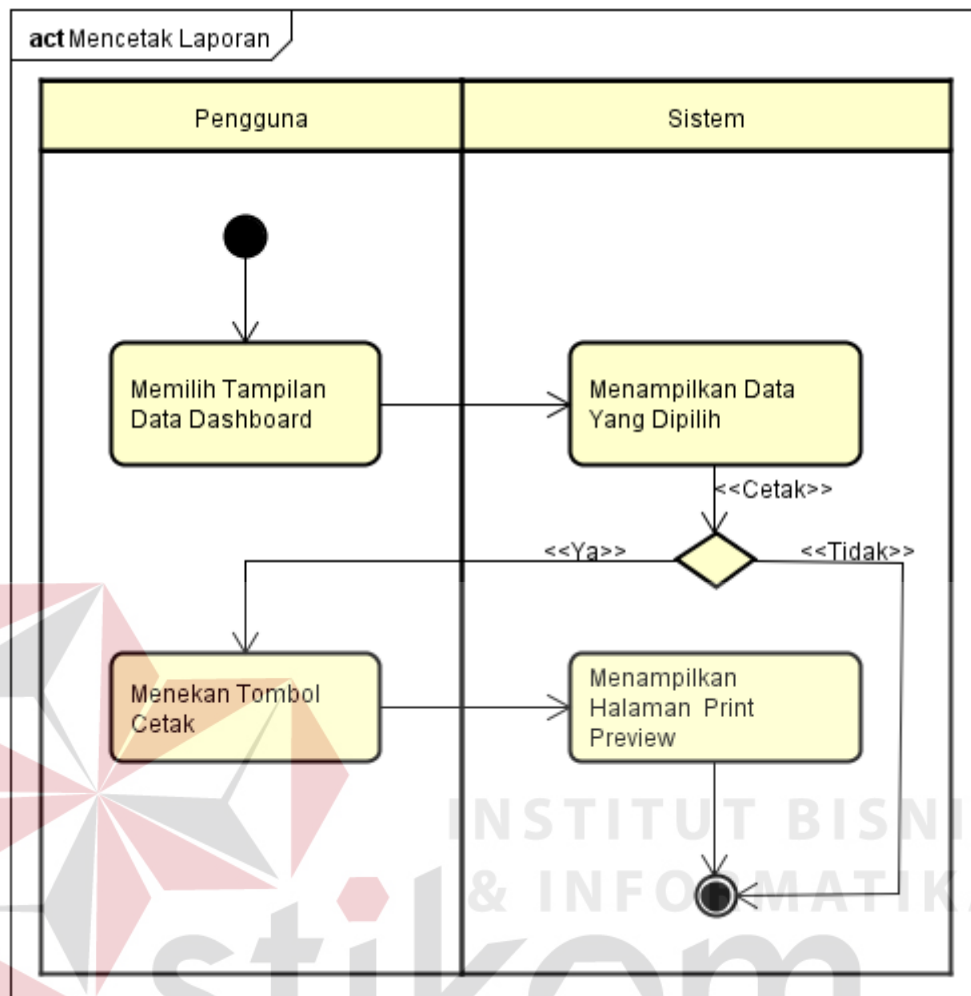


Gambar 3.9 Diagram Aktifitas Set Target

Diagram aktifitas set target dilakukan oleh pengguna, pengguna disini adalah *owner*. *Owner* dapat mengisi data set target transaksi penjualan dan *deposit* harian setelah melakukan *login* terlebih dahulu.

### h. Diagram Aktifitas Mencetak Laporan

Desain diagram aktifitas mencetak laporan beserta penjelasannya dapat dilihat pada Gambar 3.10.

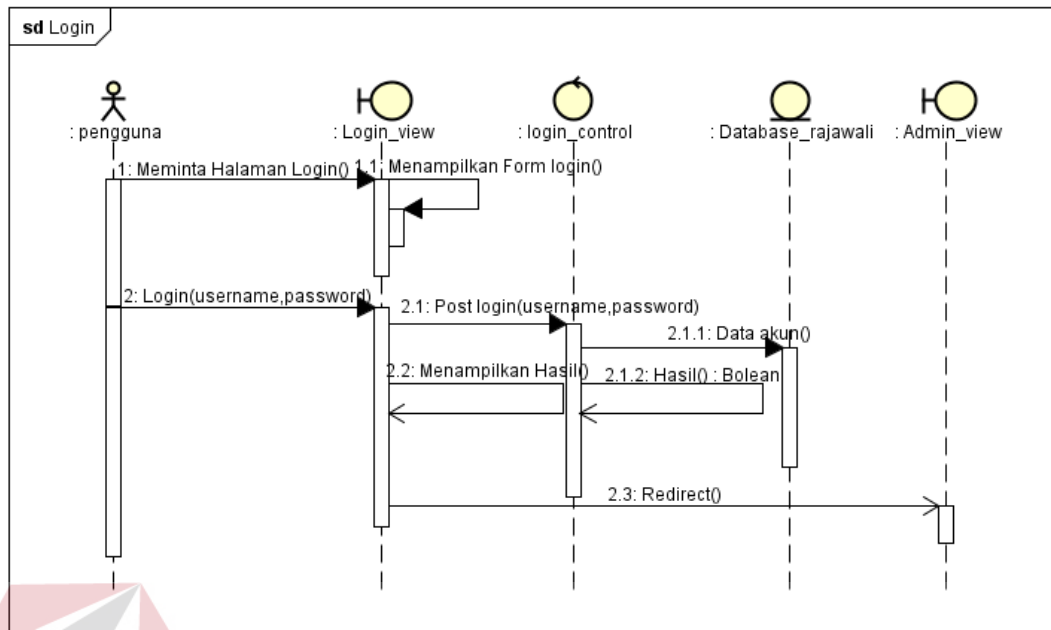


Gambar 3.10 Diagram Aktifitas Mencetak Laporan

Diagram aktifitas mencetak laporan dilakukan oleh pengguna. Pengguna memilih terlebih dahulu informasi mana yang ingin dicetak pada layar *dashboard* dan sistem akan menampilkan informasi yang dipilih tersebut. Kemudian pengguna dapat menekan tombol cetak lalu sistem akan menampilkan halaman *print preview* yang dapat mencetak informasi tersebut.

#### i. Diagram *Sequence Login*

Desain diagram *sequence login* beserta penjelasannya dapat dilihat pada Gambar 3.11.

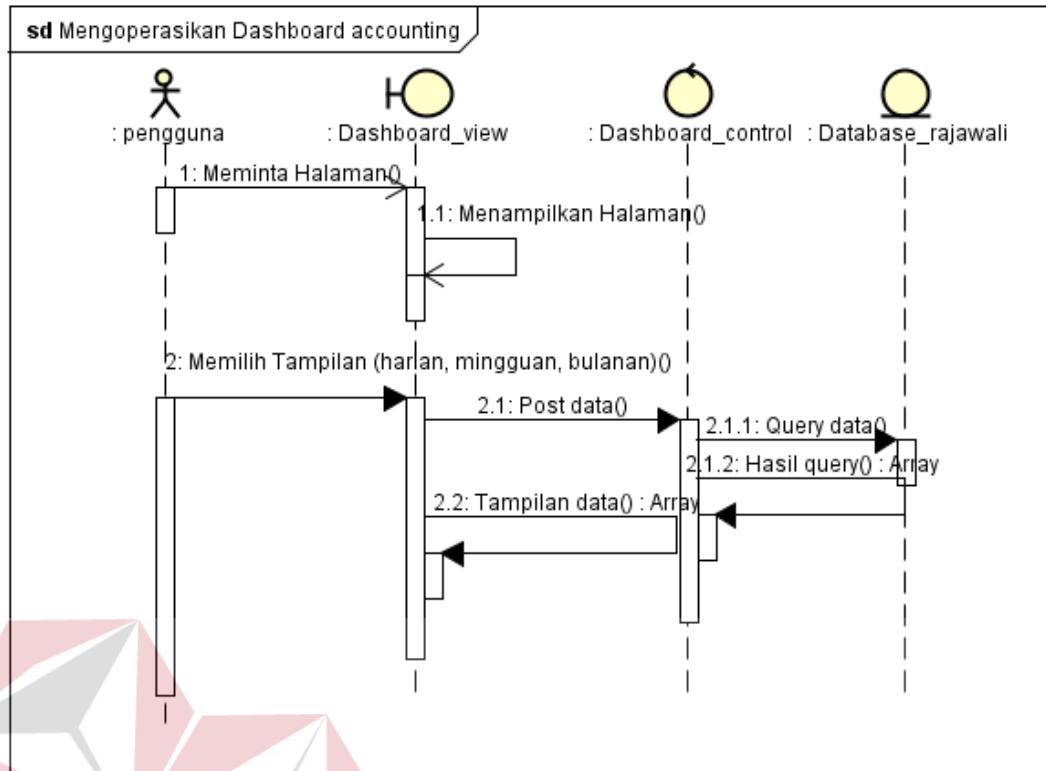


Gambar 3.11 Diagram *Sequence Login*

Diagram *sequence login* dilakukan oleh semua pengguna, pertama pengguna meminta mengakses halaman *login*. Sistem akan menampilkan halaman tersebut. Kemudian pengguna memasukkan data *login* (*username* dan *password*) dan sistem akan mengecek data tersebut. Apabila data sesuai dengan *database* akun maka sistem akan memberikan notifikasi bahwa *login* sukses. Setelah *login* berhasil maka sistem akan menampilkan halaman *dashboard*.

#### j. Diagram *Sequence Mengoperasikan Dashboard Accounting*

Desain diagram *sequence* mengoperasikan tampilan *dashboard accounting* beserta penjelasannya dapat dilihat pada Gambar 3.12.

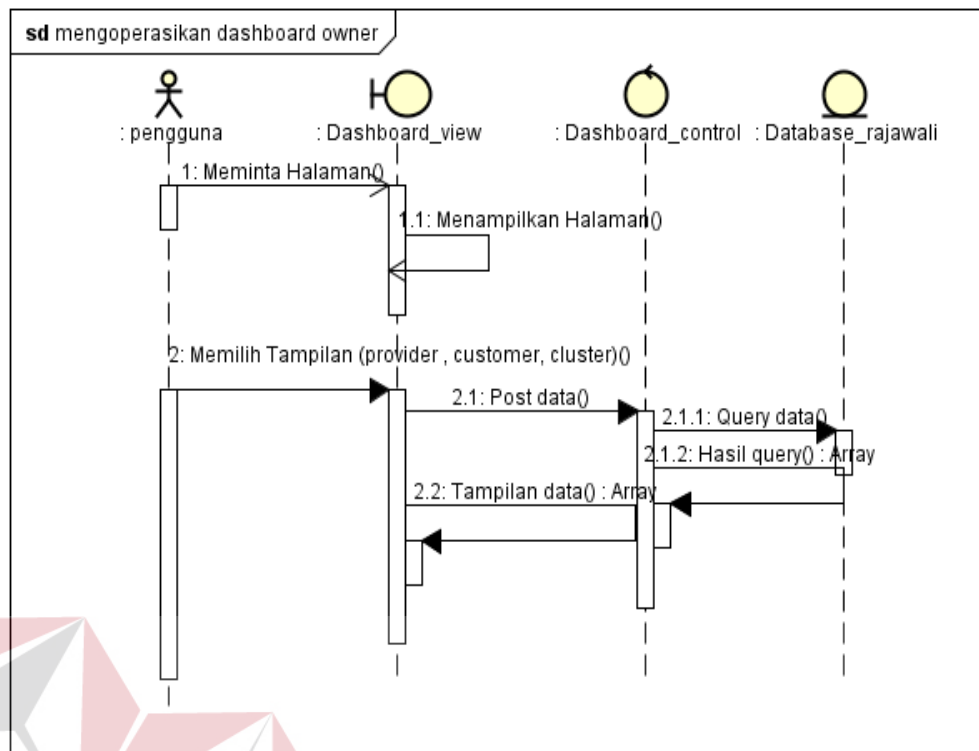


Gambar 3.12 Diagram *Sequence* Mengoperasikan *Dashboard Accounting*

Diagram *sequence* mengoperasikan tampilan *dashboard accounting* berisikan tampilan menu harian, mingguan dan bulanan. *Accounting* dapat memilih menu setelah melakukan *login* terlebih dahulu. Kemudian pengguna dapat memilih informasi yang ingin ditampilkan pada *dashboard*, maka sistem akan menampilkan informasi tersebut.

#### k. Diagram *Sequence* Mengoperasikan *Dashboard Owner*

Desain diagram *sequence* mengoperasikan tampilan *dashboard owner* beserta penjelasannya dapat dilihat pada Gambar 3.13.

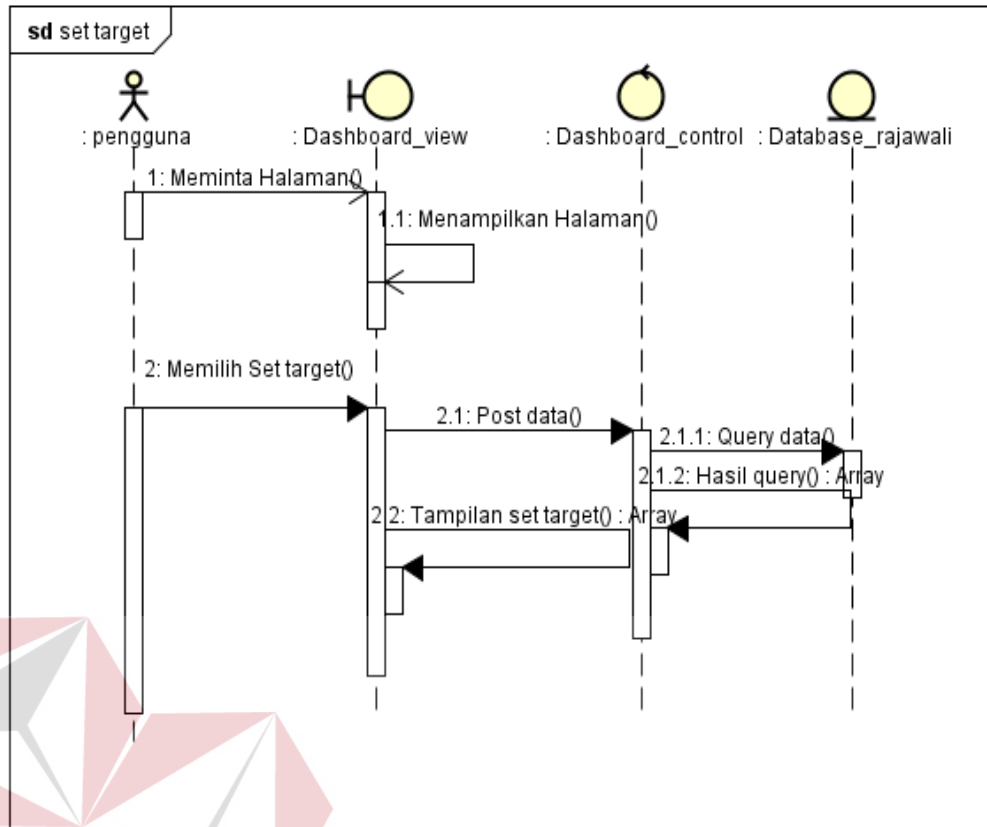


Gambar 3.13 Diagram *Sequence* mengoperasikan *Dashboard Owner*

Diagram *sequence* mengoperasikan tampilan *dashboard owner* berisikan tampilan menu *provider*, *customer* dan *cluster*. *Owner* dapat memilih menu setelah melakukan *login* terlebih dahulu. Kemudian pengguna dapat memilih informasi yang ingin ditampilkan pada *dashboard*, maka sistem akan menampilkan informasi tersebut.

#### 1. Diagram *Sequence* Set Target

Desain diagram *sequence* mencetak laporan beserta penjelasannya dapat dilihat pada Gambar 3.14.

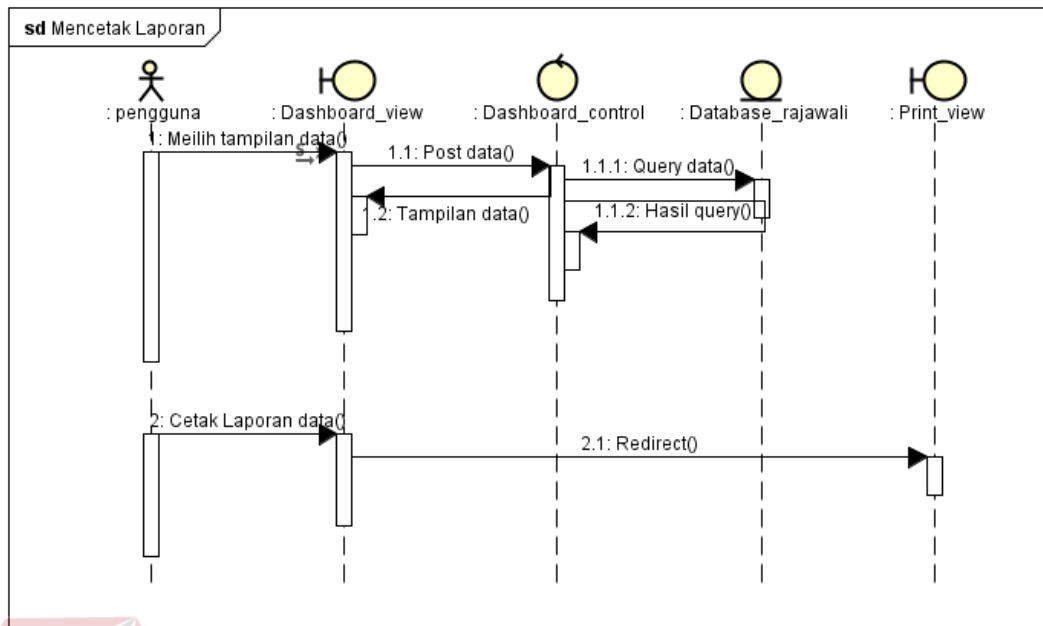


Gambar 3.14 Diagram *Sequence* Set Target

Diagram *sequence* set target digunakan untuk mengubah target transaksi penjualan dan *deposit*. Pengguna *login* terlebih dahulu untuk meminta halaman *dashboard*, setelah itu pengguna meminta halaman set target. Kemudian pengguna dapat mengubah set target transaksi penjualan dan *deposit*.

#### m. Diagram *Sequence* Mencetak Laporan

Desain diagram *sequence* mencetak laporan beserta penjelasannya dapat dilihat pada Gambar 3.15.

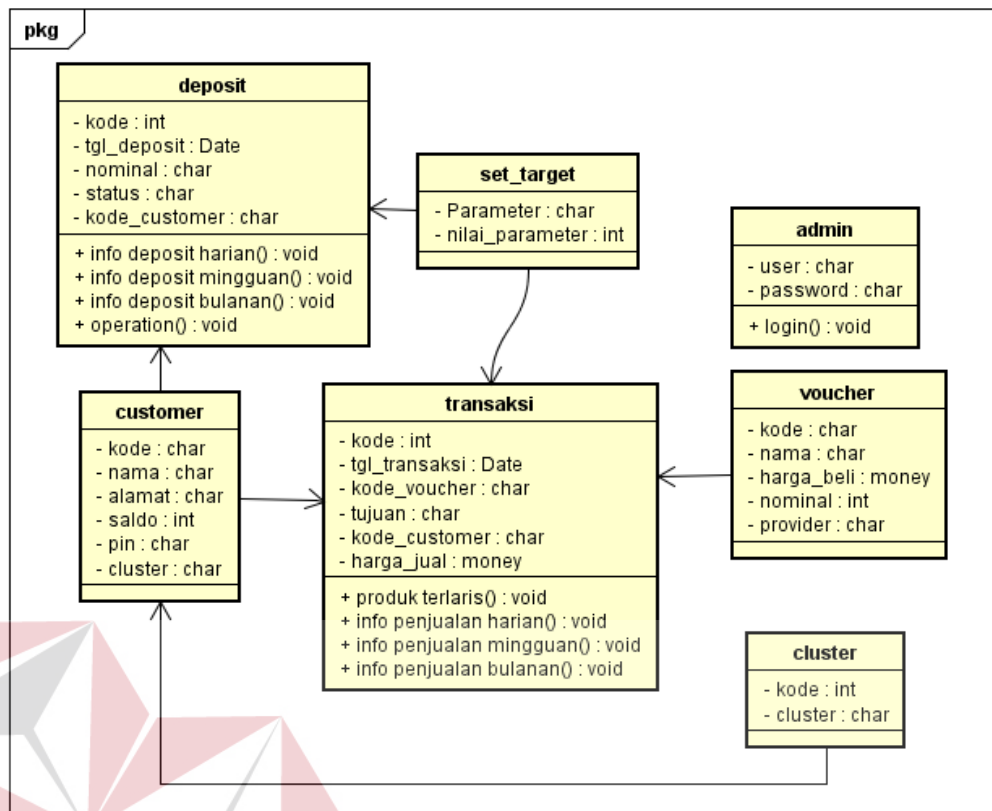


Gambar 3.15 Diagram *Sequence* Mencetak Laporan

Diagram *sequence* mencetak laporan dilakukan oleh pengguna. Pengguna memilih terlebih dahulu tampilan informasi yang ingin dicetak pada layar *dashboard* dan sistem akan menampilkan informasi yang dipilih tersebut. Kemudian pengguna dapat menekan tombol cetak lalu sistem akan menampilkan halaman *print preview* yang dapat mencetak informasi tersebut.

#### n. *Class Diagram*

*Class diagram* digunakan untuk menunjukkan interaksi antar kelas dalam sistem (Sholiq, 2006). *Class diagram* memberikan gambaran sistem secara statis dan relasi antar mereka. Berikut adalah gambar *class diagram dashboard* penjualan *voucher* pulsa elektrik Rajawali Reload 3.16.



Gambar 3.16 *Class diagram* Rajawali Reload.

*Class* yang ada memiliki atribut dan operasinya masing-masing, yang akan digunakan untuk menjelaskan bagaimana setiap *class* berperan dalam sistem, *class-class* yang ada dijelaskan sebagai berikut:

#### 1. *Class* Admin

*Class* admin adalah kelas yang digunakan untuk mengakses halaman *dashboard* penjualan *voucher* pulsa elektrik Rajawali Reload. *Class* admin memiliki atribut *user* dan *password*, pengguna harus memiliki *user* dan *password* untuk mengakses halaman *dashboard* penjualan.



## 2. *Class Deposit*

*Class deposit* merupakan kelas yang memiliki atribut kode, *tgl\_deposit*, nominal, status, *kode\_customer*. *Class* ini digunakan untuk menyimpan data *deposit customer* yang akan ditampilkan kedalam *dashboard* penjualan.

## 3. *Class Voucher*

*Class voucher* memiliki atribut kode, nama, harga\_beli, nominal, *provider*. *Class* ini digunakan untuk menyimpan data *voucher* berbagai jenis *provider* yang akan ditampilkan kedalam *dashboard*.

## 4. *Class Customer*

*Class customer* memiliki atribut kode, nama, alamat, saldo, pin, *cluster*. *Class* ini digunakan untuk menyimpan data *customer* yang mana data tersebut akan ditampilkan kedalam visual *dashboard*.

## 5. *Class Transaksi*

*Class transaksi* memiliki atribut kode, *tgl\_transaksi*, *kode\_voucher*, tujuan, *kode\_customer*, harga\_jual. *Class* ini digunakan untuk menyimpan data transaksi yang mana data tersebut akan diubah kedalam bentuk visual *dashboard*.

## 6. *Class Cluster*

*Class cluster* memiliki atribut kode dan *cluster*. *Class* ini digunakan untuk menyimpan data *cluster* yang mana data tersebut akan diubah kedalam bentuk visual *dashboard*.

## 7. *Class Set\_target*

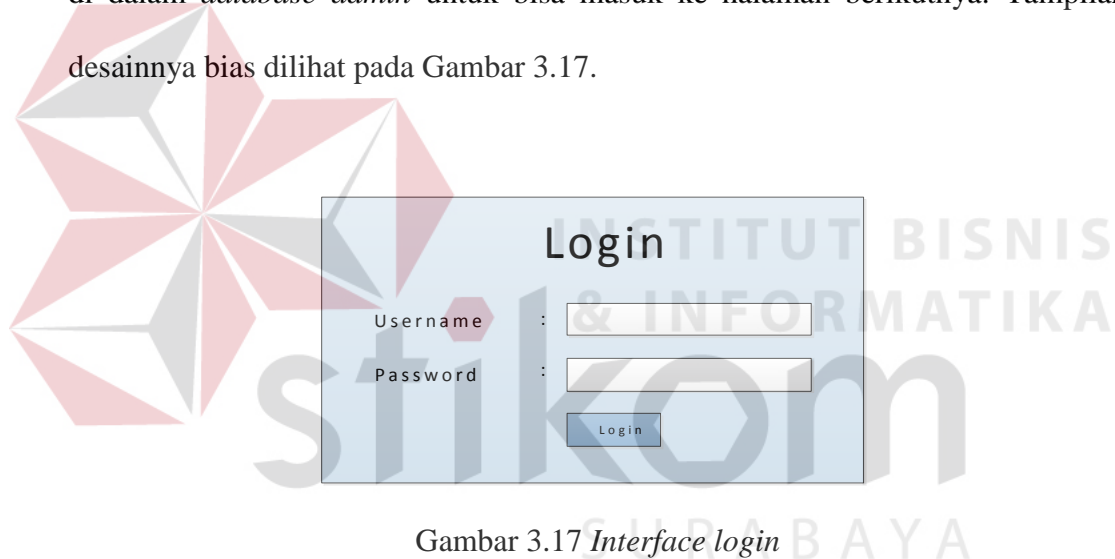
*Class set\_target* memiliki atribut nama\_parameter, nilai\_parameter. *Class* ini digunakan untuk menyimpan data perubahan target transaksi penjualan dan *deposit* harian.

### 3.3.5 Desain *Interface*

Pada tahap ini akan dibahas tentang desain *Interface* dari *dashboard* penjualan *voucher* pulsa elektrik di Rajawali Reload. Penjelasan lebih lengkapnya adalah sebagai berikut:

#### A. Desain *Interface Login*

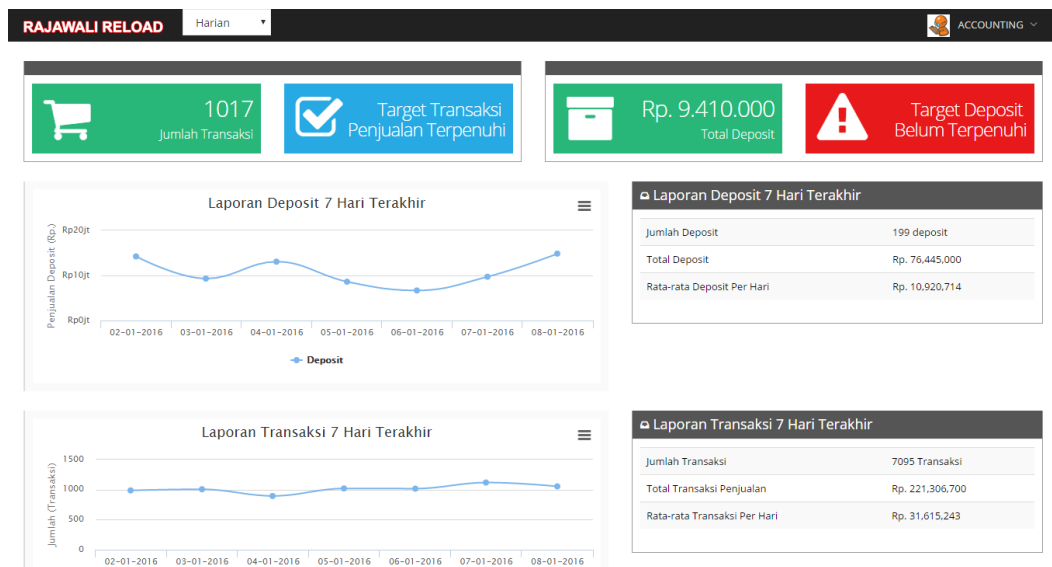
Desain tampilan halaman *login* ini dibuat sama untuk semua pengguna yaitu *owner* dan *accounting*, dan akan tampil saat pengguna mengakses *dashboard*. Pengguna harus memasukkan *username* dan *password* yang telah ada di dalam *database admin* untuk bisa masuk ke halaman berikutnya. Tampilan desainnya bias dilihat pada Gambar 3.17.



Gambar 3.17 *Interface login*

#### B. Desain *Interface Dashboard Penjualan dan Deposit Harian*

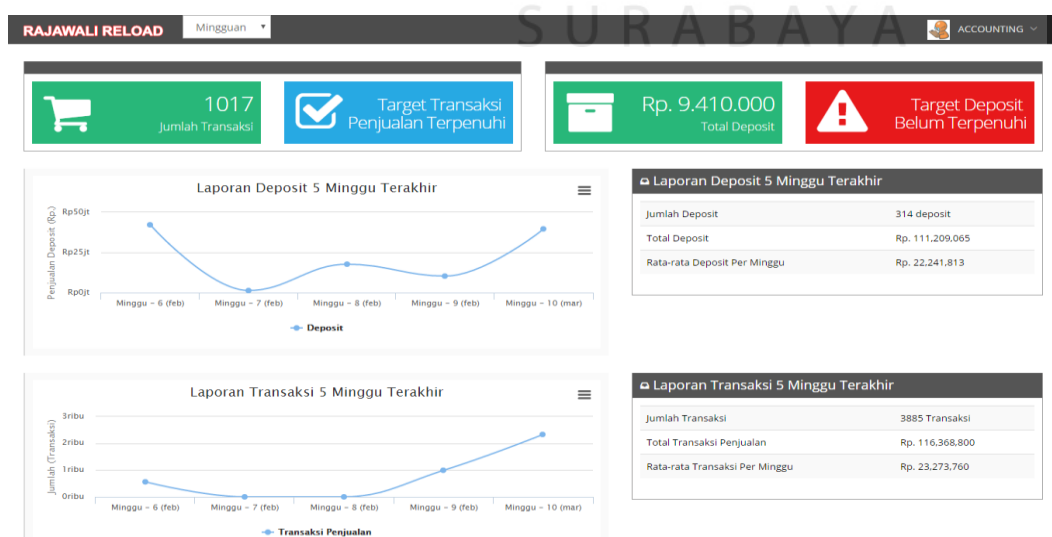
Desain tampilan *Interface dashboard* penjualan dan *deposit* harian ini akan digunakan oleh pihak *accounting*. *Interface dashboard* untuk *accounting* ini digunakan untuk menampilkan visual penjualan dan *deposit* harian sesuai kebutuhan *accounting*. Tampilan desainnya bisa dilihat pada gambar 3.18.



Gambar. 3.18 *Interface dashboard* penjualan dan *deposit* harian

### C. Desain *Interface Dashboard* Penjualan dan *Deposit* Mingguan

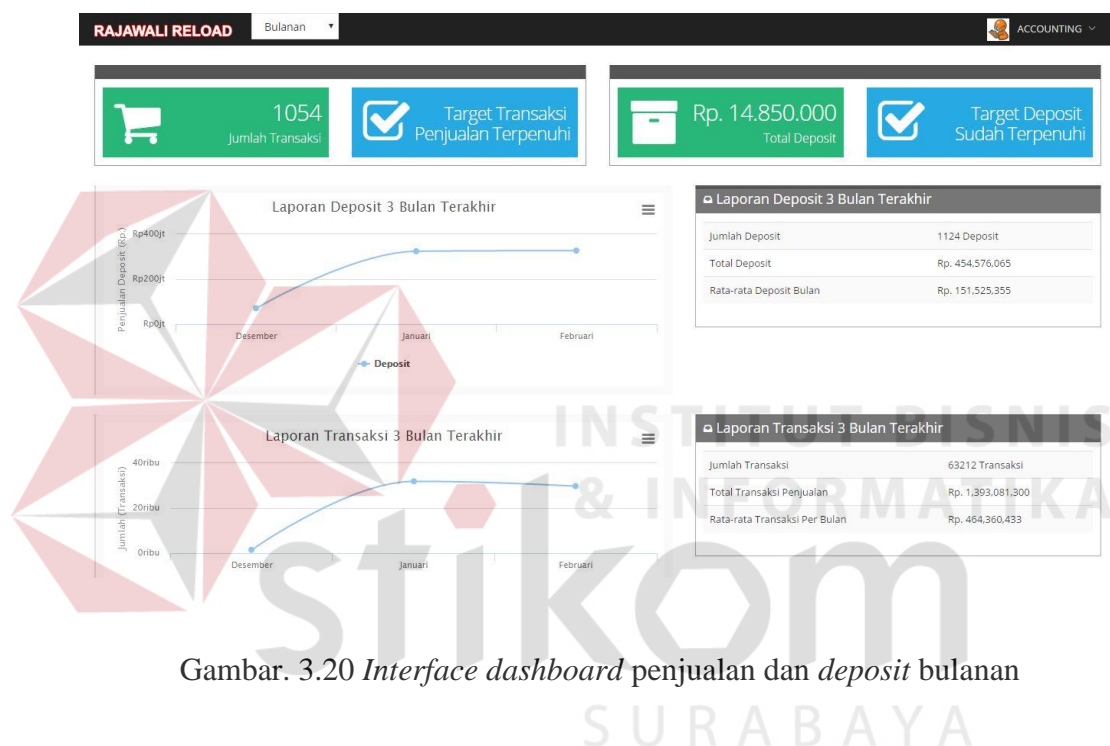
Desain tampilan *Interface dashboard* penjualan dan *deposit* mingguan ini akan digunakan oleh pihak *accounting*. *Interface dashboard* untuk *accounting* ini digunakan untuk menampilkan visual penjualan dan *deposit* mingguan sesuai kebutuhan *accounting*. Tampilan desainnya bisa dilihat pada gambar 3.19.



Gambar. 3.19 *Interface dashboard* penjualan dan *deposit* mingguan

#### D. Desain *Interface Dashboard Penjualan dan Deposit Bulanan*

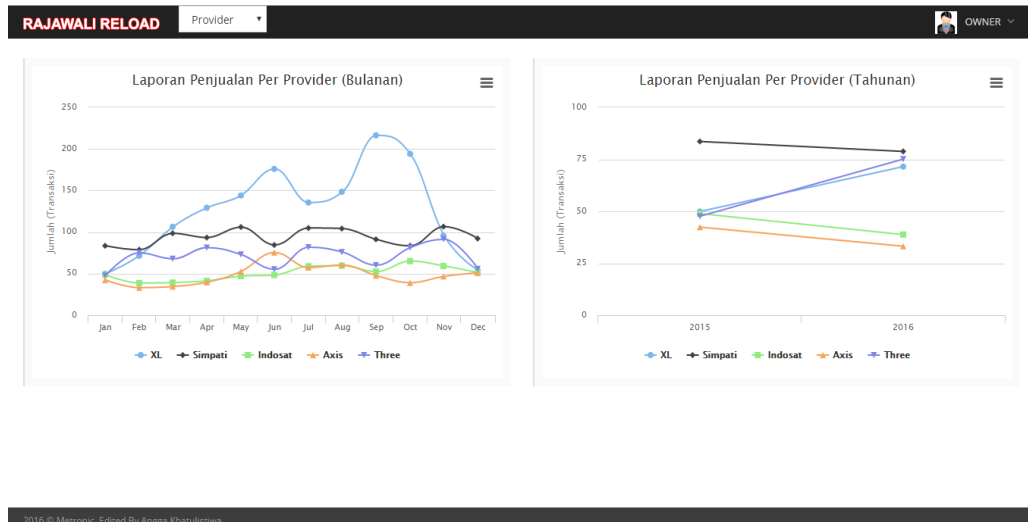
Desain tampilan *Interface dashboard* penjualan dan *deposit* bulanan ini akan digunakan oleh pihak *accounting*. *Interface dashboard* untuk *accounting* ini digunakan untuk menampilkan visual penjualan dan *deposit* bulanan sesuai kebutuhan *accounting*. Tampilan desainnya bisa dilihat pada gambar 3.20.



Gambar. 3.20 *Interface dashboard* penjualan dan *deposit* bulanan

#### E. Desain *Interface Dashboard untuk Voucher Provider*

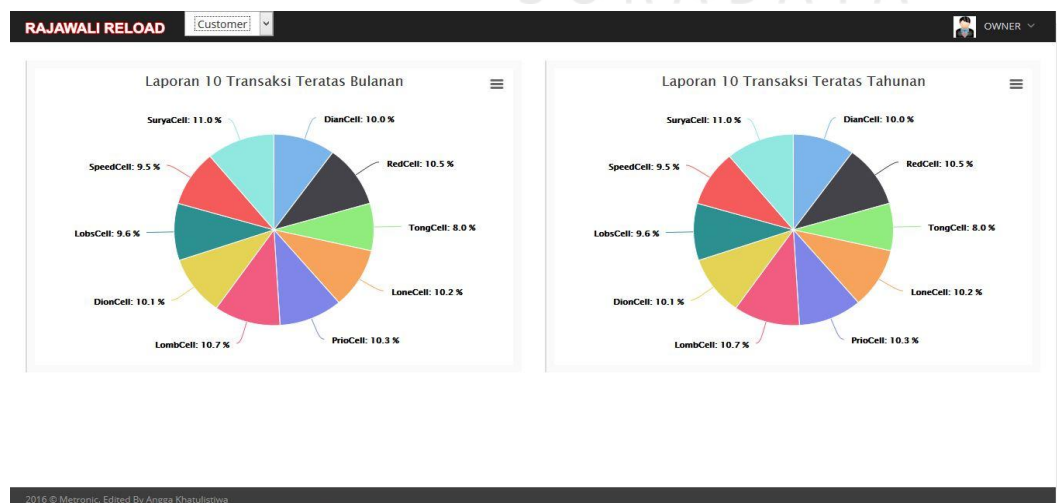
Desain tampilan *Interface dashboard* untuk *provider Voucher* dan berkontribusi ini akan digunakan oleh *owner*, Halaman *dashboard* ini menampilkan grafik informasi penjualan *voucher provider* perkategori bulanan dan tahunan yang dibutuhkan oleh *owner*. Tampilan desainnya bisa dilihat pada gambar 3.21.



Gambar 3.21 *Interface dashboard voucher provider (owner)*

## F. Desain *Interface Dashboard* untuk Penjualan *Customer*

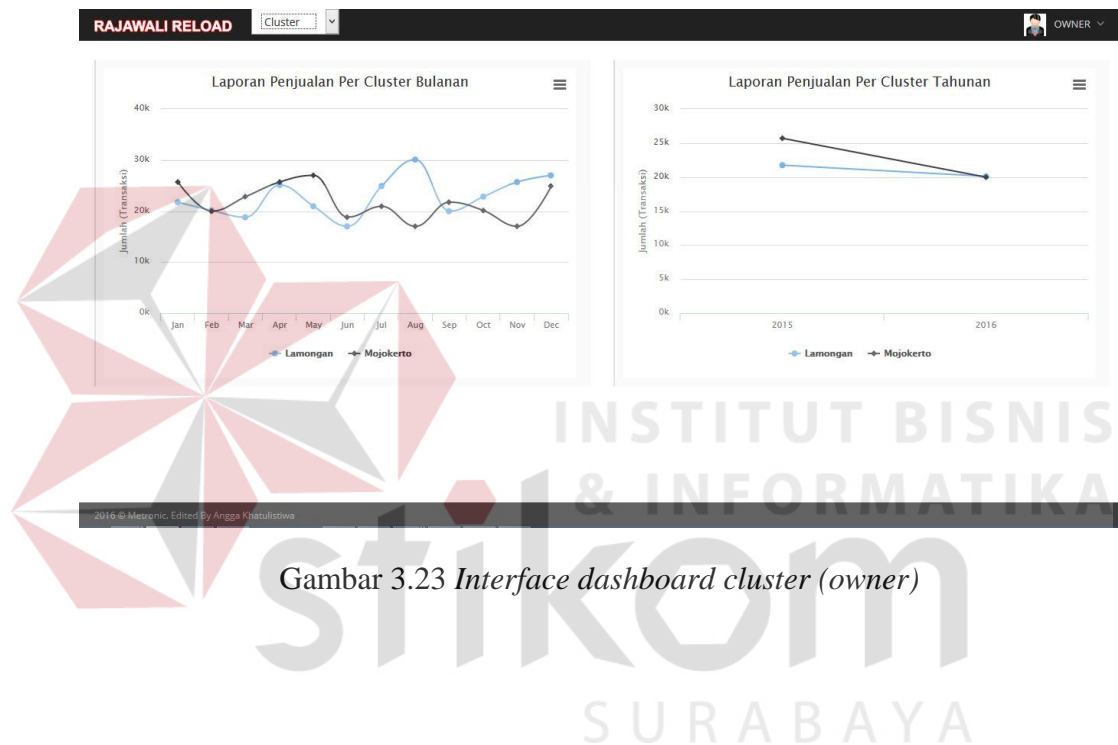
Desain tampilan *Interface dashboard* untuk penjualan berdasarkan *customer* ini akan digunakan oleh *owner*, Halaman *dashboard* ini menampilkan grafik informasi penjualan 10 *customer* terbanyak dalam melakukan penjualan *voucher* bulanan dan tahunan yang dibutuhkan oleh *owner*. Tampilan desainnya bisa dilihat pada gambar 3.22.



Gambar 3.22 *Interface dashboard customer (owner)*

### G. Desain *Interface Dashboard* untuk *Cluster*

Desain tampilan *Interface dashboard* penjualan per *cluster* ini akan digunakan oleh *owner*, Halaman *dashboard* ini menampilkan grafik informasi penjualan per *cluster* bulanan dan tahunan yang dibutuhkan oleh *owner*. Tampilan desainnya bisa dilihat pada gambar 3.23.



Gambar 3.23 *Interface dashboard cluster (owner)*