



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, 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

METODE DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang digambarkan pada Tabel 3.1.

Tabel 3.1 Rencana Tahap Penelitian

Kegiatan	Minggu ke-													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Telaah literatur	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Pengumpulan Data			■	■										
Analisis dan Perancangan					■	■								
Implementasi							■	■	■	■				
Pengujian											■	■		
Evaluasi													■	■
Konsultasi dan penulisan	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Adapun dari Tabel 3.1, dapat dijelaskan tahap-tahap yang dilakukan selama melakukan penelitian adalah sebagai berikut.

1. Telaah Literatur

Pada tahap ini, dilakukan pencarian referensi dan pembelajaran mengenai bahan topik dan materi yang diperlukan dalam pengembangan aplikasi sistem pendukung keputusan *market basket analysis* ini. Beberapa hal yang penting adalah pemahaman mengenai *market basket analysis*, algoritma Apriori dan pengembangannya, serta penerapan dalam pembuatan aplikasinya. Hal lain yang perlu dipahami, yaitu cara penggunaan Laravel sebagai *framework* PHP.

Telaah literatur menggunakan referensi buku, jurnal, dan artikel terkait.

2. Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari data transaksi yang tercatat secara komputerisasi oleh sistem pada Minimarket Dua Putri. Data yang diperoleh merupakan transaksi nyata dengan jangka waktu 3 bulan dimulai dari tanggal 13 Januari 2016 sampai dengan tanggal 15 Maret 2016. Dalam periode tersebut terdapat 9562 transaksi.

Tabel 3.2 Contoh Format Awal Data Transaksi

no_struk	01ERL/AA001160113001
tgl_struk	13-01-2016
kd_reg	1
Customer_pay	CASH
Kd_ksr	ERL
lks	AA001
Item_barang	8993007000239 - 000247500000001 - UHT INDOMILK KIDS CHOCO 115ML
sat	pcs
Lks_brg	G0001
Hrg_sat	2300
Sub_total_disc	2300

3. Analisis dan Perancangan

Pada tahap ini dilakukan perancangan aplikasi Market Basket Analysis mulai dari diagram umum aplikasi, yaitu gambaran umum mengenai proses *market basket analysis*, dan *flowchart* dari algoritma *Improved Apriori* Shankar Bargadiya. Selain itu, dilakukan juga perancangan umum User Interface web yang digunakan untuk menampilkan hasil rekomendasi.

4. Implementasi

Sesuai dengan teori yang dipaparkan sebelumnya, pada dasarnya algoritma *Improved Apriori* Shankar Bargadiya memiliki tujuan akhir yang sama dengan penggunaan algoritma Apriori, yaitu untuk menemukan *frequent itemset* menggunakan teknik asosiasi. Dengan teknik asosiasi, barang-

barang yang sering muncul bersama-sama pada sebuah transaksi akan ditemukan. *Frequent itemset* akan menentukan hasil kombinasi yang akan terbentuk.

Tabel 3.3 Format *File Excel* Data Transaksi

id_header	01ERL/AA001160113001
tgl_transaksi	13-01-2016
Kd_kasir	ERL
id_barang	000247500000001
qty	2
Harga_satuan	2300
subtotal	4600
Dsc_subtotal	4600
Total	4600

Tabel 3.4 Format *File Excel* Data Inventories

Id_barang	000186900000001
Nama_barang	PARAMEX
Barcode	8998667100206
Stok	31
Harga_modal	1630
Harg_satuan	2000

Tabel 3.3 menunjukkan format kolom *file excel* untuk *import data transaksi* ke dalam sistem. Tabel 3.4 menunjukkan format kolom *file excel* untuk *import data inventory* ke dalam sistem.

Pembuatan aplikasi *Market Basket Analysis* dibutuhkan masukkan berupa daftar transaksi yang pernah terjadi. Masukkan ini diperlukan untuk melihat *item* apa saja yang pernah dibeli oleh konsumen, kemudian diolah untuk mendapatkan pola pembelian dengan melakukan perhitungan menggunakan algoritma *Improved Apriori Shankar Bargadiya*. Selain itu, dibutuhkan masukan *minimum support* untuk melakukan *filter* terhadap setiap *item* kandidat *k-itemset*. *Minimum support* dibuat secara dinamis sehingga harus di-*input* oleh *user*.

Algoritma *Improved Apriori* Shankar Bargadiya berbeda dengan algoritma Apriori yang melakukan *query database* berulang-ulang sesuai dengan jumlah *k-itemset* yang terbentuk. Algoritma *Improved Apriori* Shankar Bargadiya hanya melakukan dua kali *query database* (Lydia, 2014).

Langkah awal adalah menemukan *frequent 1-itemset*, yaitu mengetahui frekuensi munculnya sebuah *item* dalam daftar transaksi yang digunakan sebagai *input* aplikasi. Kemudian, di-filter menggunakan *minimum support*. *Item* dengan frekuensi munculnya kurang dari *minimum support* akan dihapus, sedangkan *item* dengan frekuensi munculnya minimal sama dengan *minimum support* tetap dipertahankan sebagai kandidat pembentukan *itemset* selanjutnya. *Item* yang tersisa tiap transaksi akan dibuat menjadi *Local Power Set*.

Langkah selanjutnya adalah membuat *Global Power Set* dengan inialisasi *count* = 0. Pada tahap ini, frekuensi masing-masing kandidat *k-itemset* dihitung. Perhitungan ini dilakukan dengan membandingkan *Local Power Set* dan *Global Power Set*. Jika kandidat *k-itemset* pada *Global Power Set* ditemukan pada *Local Power Set*, maka *count* dari *k-itemset* tersebut ditambah satu. Setelah perbandingan selesai dilakukan, kemudian dilakukan pemangkasan dengan *minimum support* sebagai nilainya. Jika jumlah *k-itemset* di bawah *minimum support*, maka dihapus dari *Global Power Set*. *Item* yang tersisa adalah *frequent itemset* yang disebut juga sebagai *strong rule*. Dari hasil tersebut, ditemukan *item* yang dapat merekomendasikan sebagai kombinasi barang.

5. Pengujian

Pada tahap ini dilakukan uji coba terhadap aplikasi dengan menggunakan laptop ASUS K-43TK. Tujuan dari pengujian adalah untuk mengetahui *error* atau *bug*, kemudian dilakukan perbaikan terhadap aplikasi jika ditemukan.

6. Evaluasi

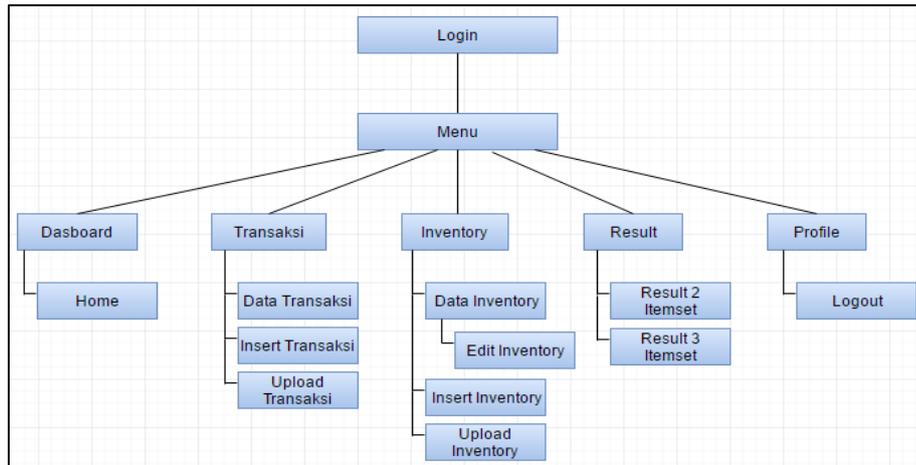
Penggunaan *lift ratio* merupakan teknik evaluasi yang dianggap tepat pada penelitian ini dikarenakan hasil pada pencarian *rules* tersebut akan menghasilkan suatu nilai *confidence* dan *support*. Dalam penelitian Yusuf dan Kuncoro (2014) dan Rafsanjani, dkk. (2014), menyatakan bahwa nilai *lift ratio* yang besar menandakan kemungkinan pasangan barang sangat besar terjadi. Oleh karena itu, *lift ratio* digunakan sebagai metode evaluasi pada penelitian ini.

3.2 Perancangan

Adapun untuk melakukan pemrograman aplikasi dibutuhkan sebuah perancangan yang tepat. Perancangan tersebut dilakukan dengan membuat diagram-diagram pendukung yang lazim digunakan dalam penelitian. *Flowchart* diagram digunakan untuk melakukan perancangan pada penelitian ini.

3.2.1 Sitemap

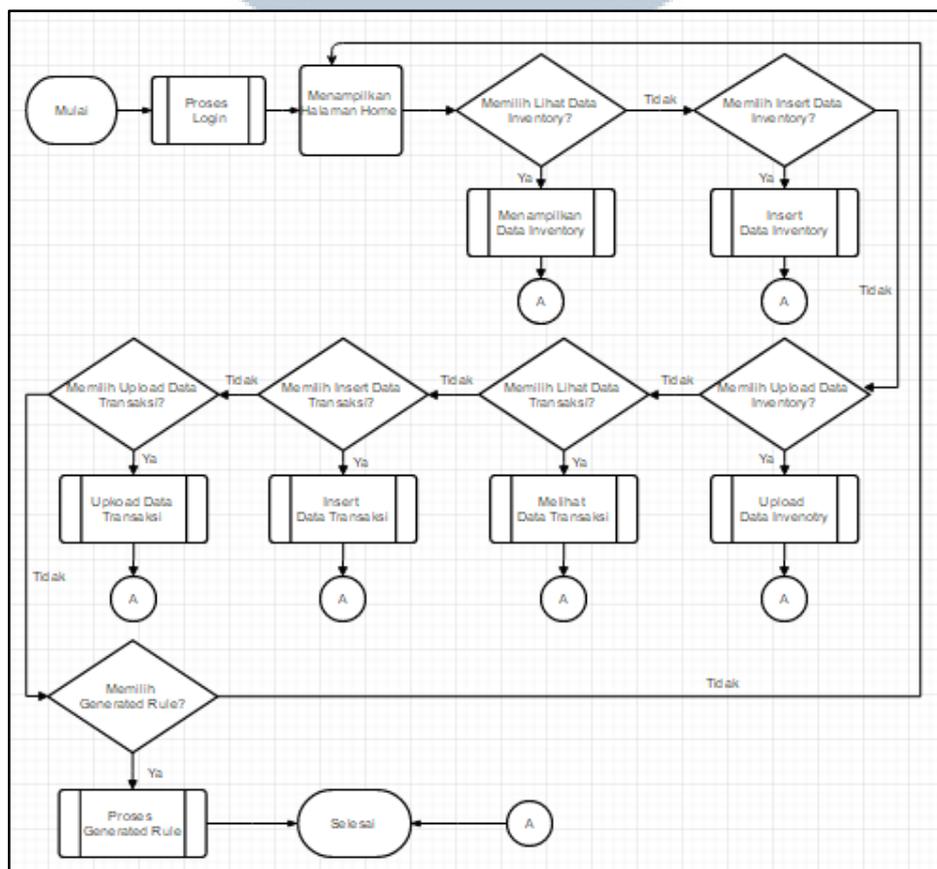
Gambar 3.1 merupakan *sitemap* yang digunakan untuk menjelaskan halaman dalam sistem. Halaman pertama adalah halaman *login* yang berfungsi untuk proses otentikasi. Selanjutnya terdapat lima menu untuk menuju halaman Home, Data Transaksi, Insert Transaksi, Upload Transaksi, Data Inventory, Edit Inventory, Insert Inventory, Upload Inventory, Result 2 Itemset, dan Result 3 Itemset.



Gambar 3.1 Sitemap

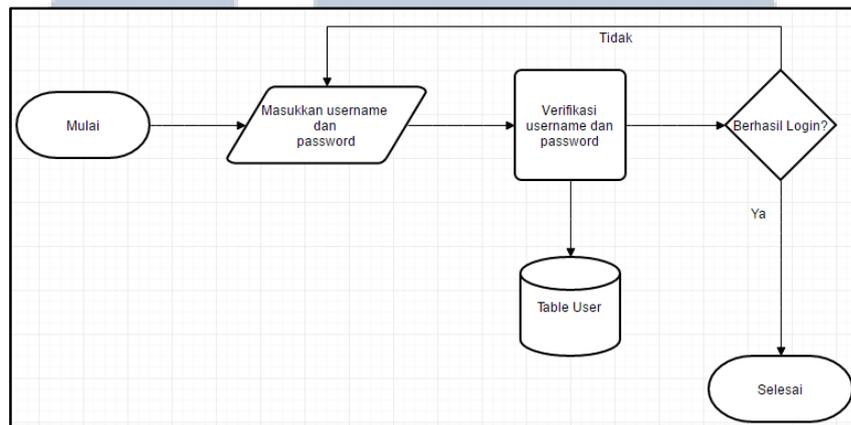
3.2.2 Flowchart Diagram

Berikut adalah *flowchart* utama aplikasi yang memiliki tujuh pilihan menu yang dapat dipilih oleh pengguna yang ditunjukkan oleh Gambar 3.1.



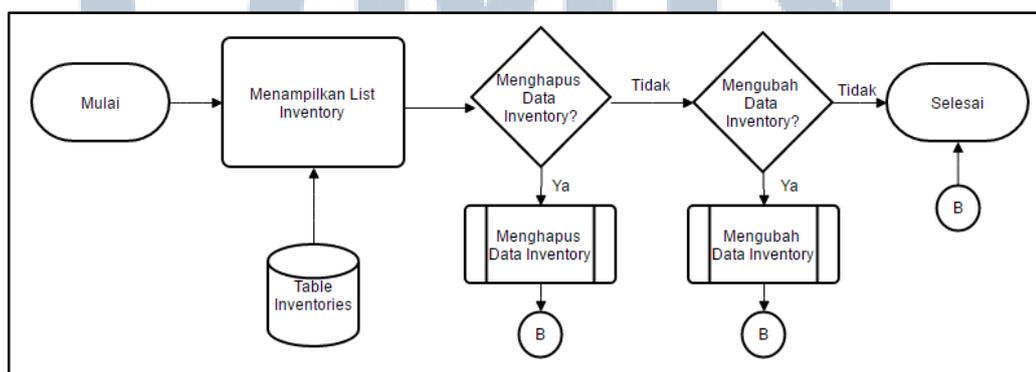
Gambar 3.2 Flowchart Utama Aplikasi

Gambar 3.2 menjelaskan setelah pengguna melakukan *login*. Pengguna dapat memilih tujuh menu, yaitu Menampilkan Data Inventory, Insert Data Inventory, Upload Data Inventory, Melihat Data Transaksi, Insert Data Transaksi, Upload Data Transaksi, Proses Generated Rule.



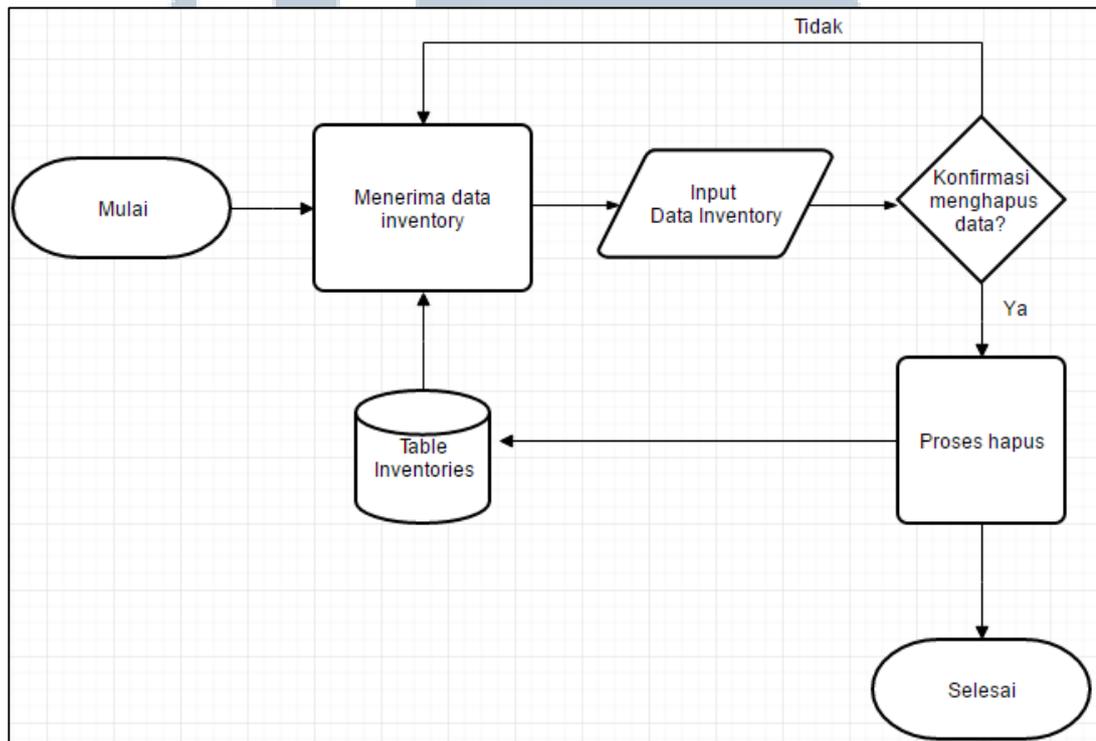
Gambar 3.3 Flowchart Proses Login

Gambar 3.3 menunjukkan *flowchart* proses *login* untuk memasuki sistem. Pengguna memasukkan *username* dan *password*. Kemudian, sistem akan melakukan verifikasi *username* dan *password*. Jika berhasil, maka proses *login* selesai. Jika tidak, maka pengguna tetap diarahkan kepada halaman *login*.



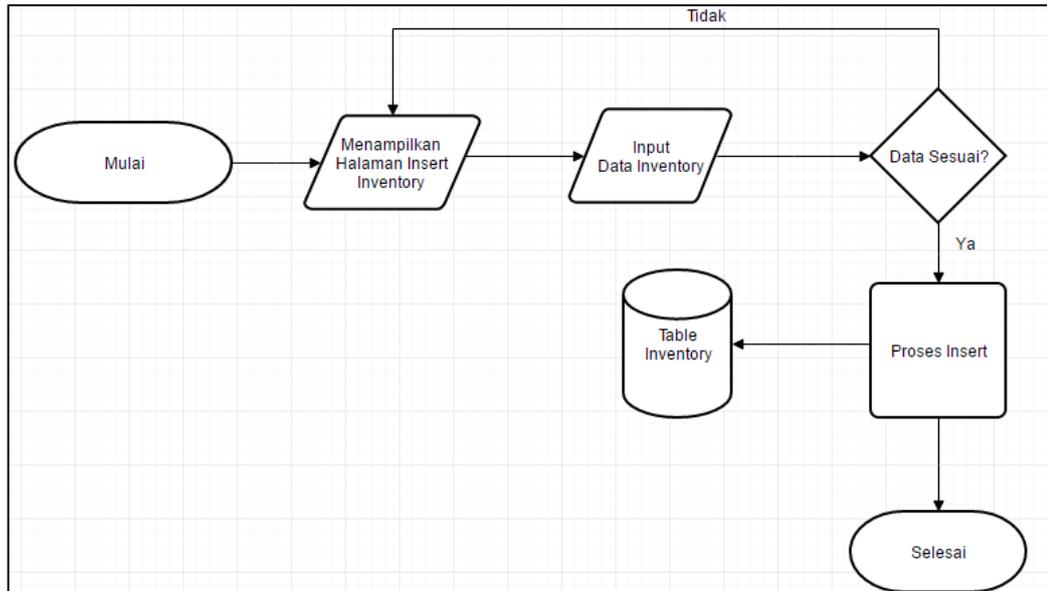
Gambar 3.4 Flowchart Menampilkan Data Inventory

Gambar 3.4 menunjukkan *flowchart* menampilkan data *inventory*. Pengguna ditampilkan daftar *inventory*. Kemudian, pengguna dapat menghapus data *inventory* atau mengubah data *inventory*.



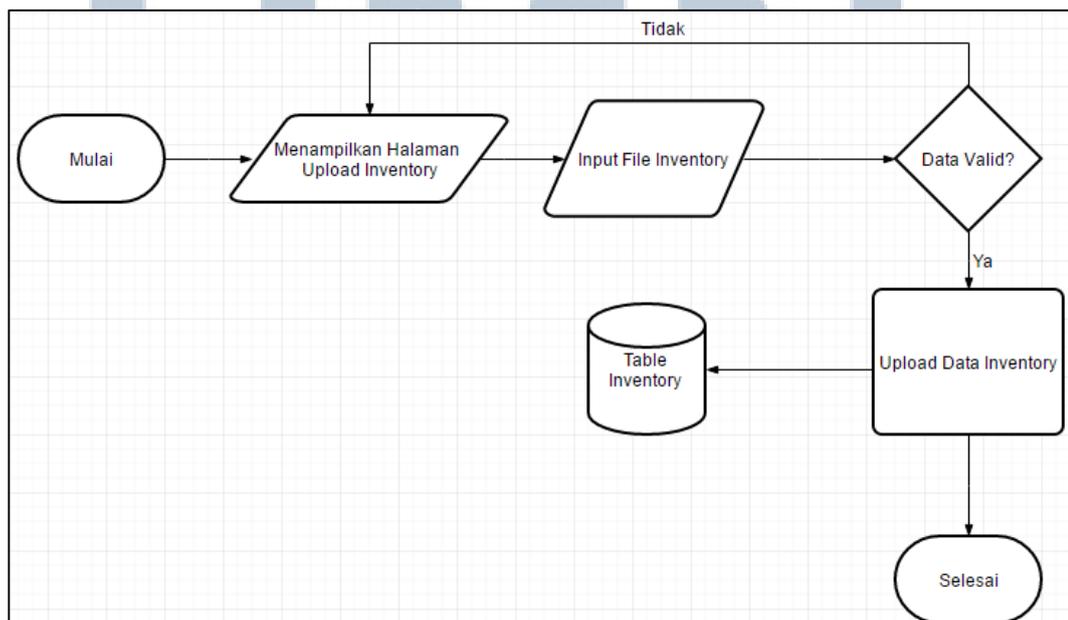
Gambar 3.5 Flowchart Menghapus Data Inventory

Gambar 3.5 menunjukkan *flowchart* menghapus data *inventory*. Pengguna memilih data *inventory* yang ingin dihapus. Kemudian, pengguna akan diminta melakukan konfirmasi bahwa data tersebut ingin dihapus. Jika pengguna mengkonfirmasi, maka data *inventory* dihapus. Jika tidak, maka pengguna akan diarahkan kembali ke halaman Menampilkan Data Inventory.



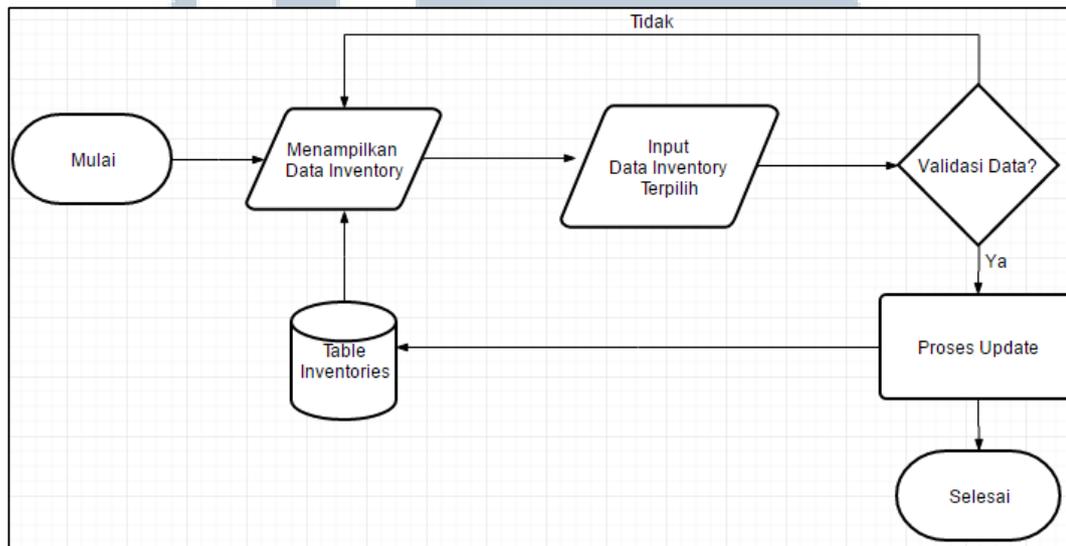
Gambar 3.6 Flowchart Insert Data Inventory

Gambar 3.6 menunjukkan *flowchart insert data inventory*. Pengguna memasukkan data *inventory* yang diinginkan. Kemudian, *sistem* akan melakukan validasi apakah data yang dimasukkan sesuai. Jika validasi berhasil, maka data *inventory* disimpan. Jika tidak, sistem akan mengarahkan pengguna ke Halaman Insert Inventory.



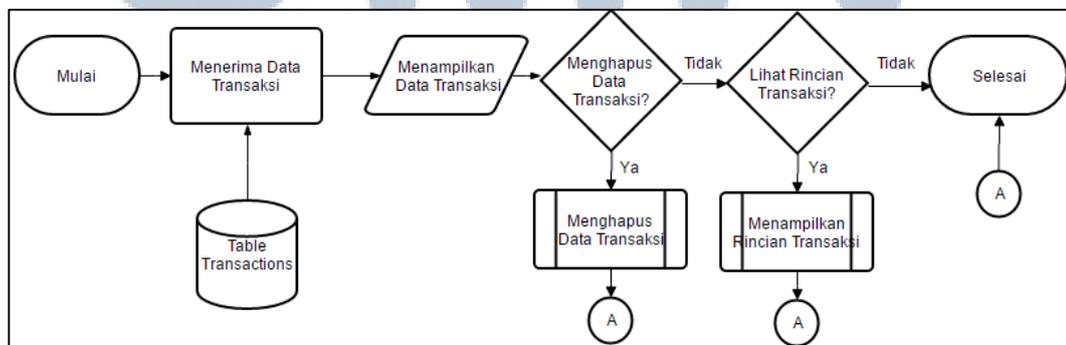
Gambar 3.7 Flowchart Upload Data Inventory

Gambar 3.7 menunjukkan *flowchart upload data inventory*. Pengguna akan memilih *file excel* yang ingin dipilih. Jika data yang dimasukkan sesuai, maka data *inventory* di upload ke *server*. Jika tidak, pengguna akan diarahkan ke Halaman Upload Inventory.



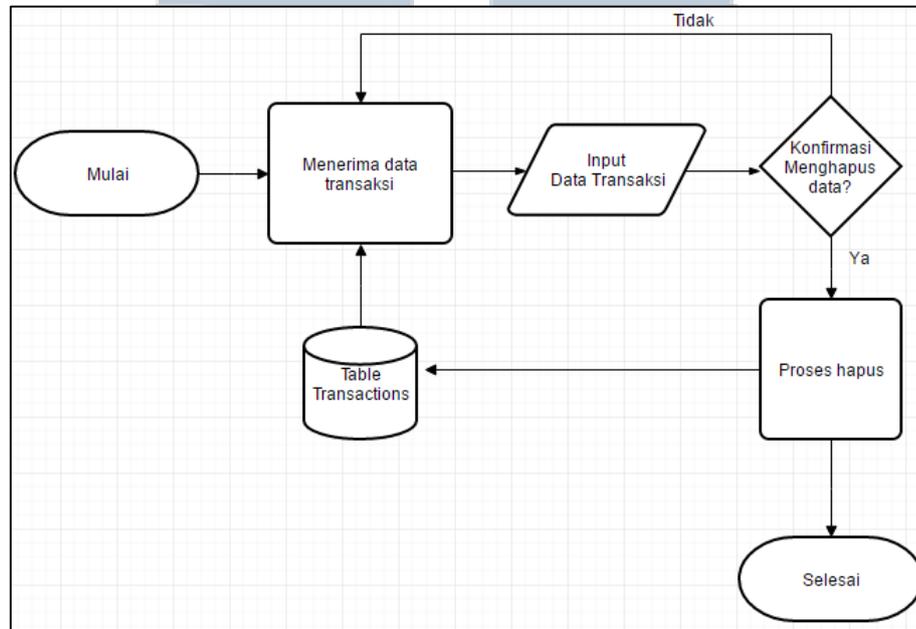
Gambar 3.8 Flowchart Mengubah Data Inventory

Gambar 3.8 menunjukkan *flowchart* mengubah data *inventory*. Sistem akan menampilkan data *inventory* yang akan diubah. Kemudian, pengguna memasukkan data *inventory* yang baru. Jika data *valid*, maka proses *update* berhasil. Jika data tidak *valid*, maka pengguna akan diarahkan kembali ke Halaman Update.



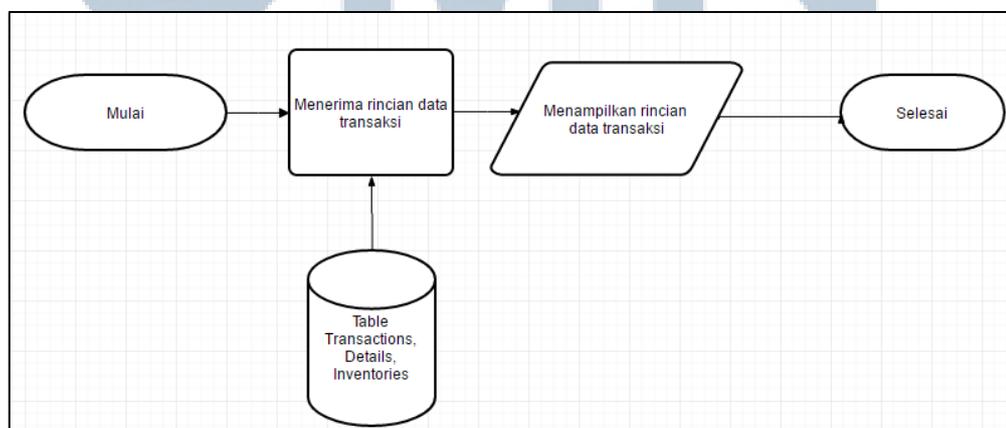
Gambar 3.9 Flowchart Melihat Data Transaksi

Gambar 3.9 menunjukkan *flowchart* melihat data transaksi. Pengguna ditampilkan daftar transaksi yang tercatat. Kemudian, pengguna dapat memilih untuk menghapus data transaksi atau melihat rincian transaksi.



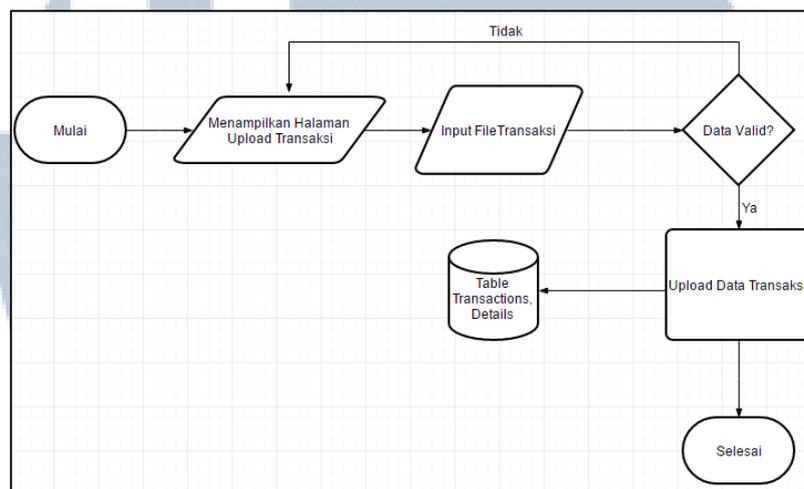
Gambar 3.10 Flowchart Menghapus Data Transaksi

Gambar 3.10 menunjukkan *flowchart* menghapus data transaksi. Pengguna memilih data transaksi mana yang ingin dihapus. Jika pengguna melakukan konfirmasi untuk menghapus, maka data transaksi dihapus. Jika pengguna membatalkan konfirmasi, maka data transaksi batal dihapus.



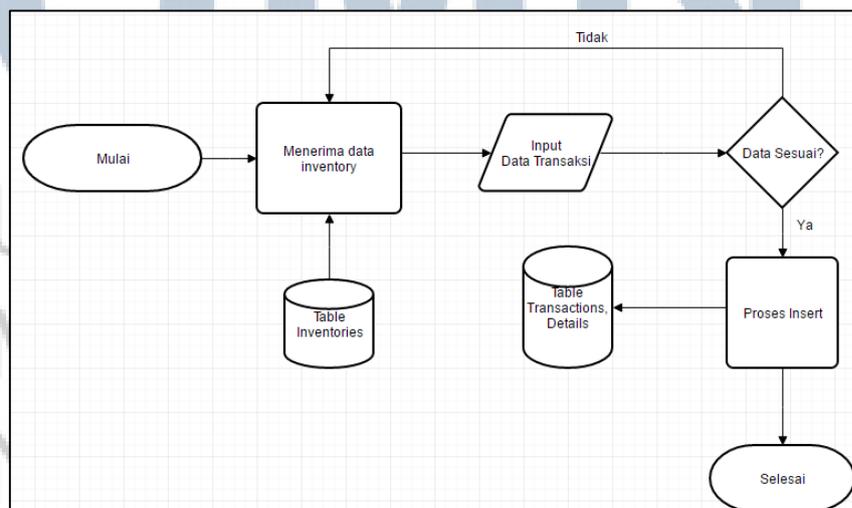
Gambar 3.11 Flowchart Menampilkan Rincian Transaksi

Gambar 3.11 menunjukkan *flowchart* rincian transaksi. Pengguna sebelumnya sudah memilih data transaksi yang ingin dilihat. Kemudian, sistem memunculkan rincian data transaksi, yaitu kode struk, tanggal transaksi, kode kasir, dan rincian barang yang terjual.



Gambar 3.12 Flowchart Upload Data Transaksi

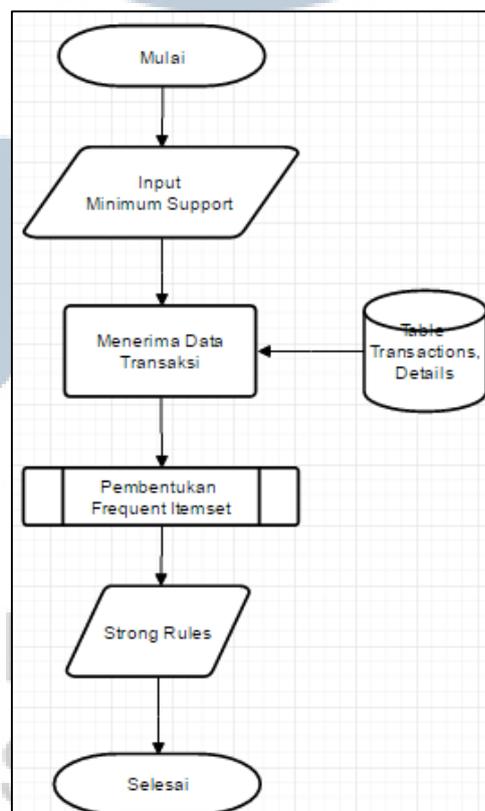
Gambar 3.12 menunjukkan flowchart upload data transaksi. Pengguna akan ditampilkan Halaman Upload Transaksi. Kemudian, pengguna memilih *file excel* yang ingin di-*input*. Jika data sesuai, maka data transaksi disimpan. Jika tidak, maka sistem akan menampilkan Halaman Upload Transaksi.



Gambar 3.13 Flowchart Insert Data Transaksi

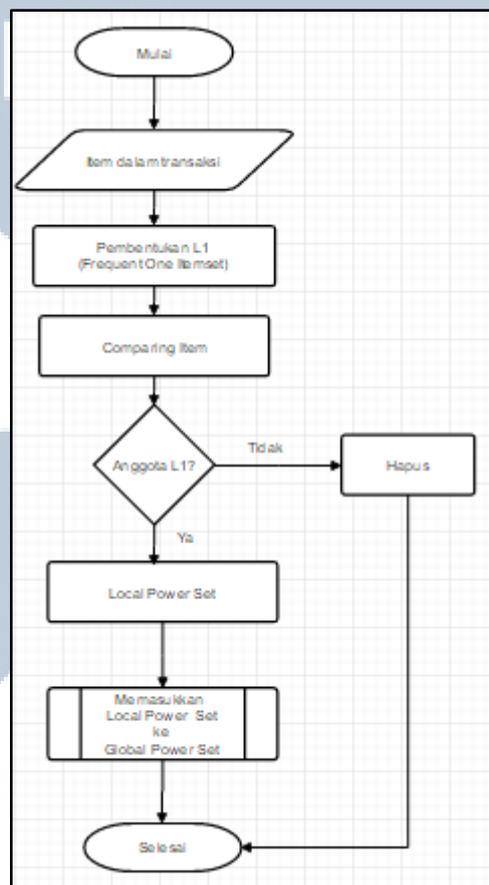
Gambar 3.13 menunjukkan *flowchart insert* data transaksi. Pengguna memasukkan kode struk, tanggal transaksi, dan nama barang beserta jumlahnya. Setelah itu, sistem akan melakukan validasi terhadap data yang di-*input*. Jika validasi berhasil, maka data transaksi disimpan.

Berikut akan digambarkan *flowchart* mengenai proses dari Aplikasi Market Basket Analysis menggunakan algoritma *Improved Apriori* Shankar Bargadiya. Gambar 3.14 menunjukkan alur proses cara kerja algoritma *Improved Apriori* Shankar Bargadiya. Sebelum dimulai melakukan pengambilan data dilakukan penetapan *minimum support* yang dilakukan oleh *user*. Kemudian, dilanjutkan dengan tahap pembentukan *frequent itemset*. Dari hasil perhitungan menggunakan algoritma *Improved Apriori* Shankar Bargadiya akan diperoleh *item* yang saling berkaitan. Hasil tersebut disebut sebagai *strong rules*.



Gambar 3.14 Flowchart Generated Rule

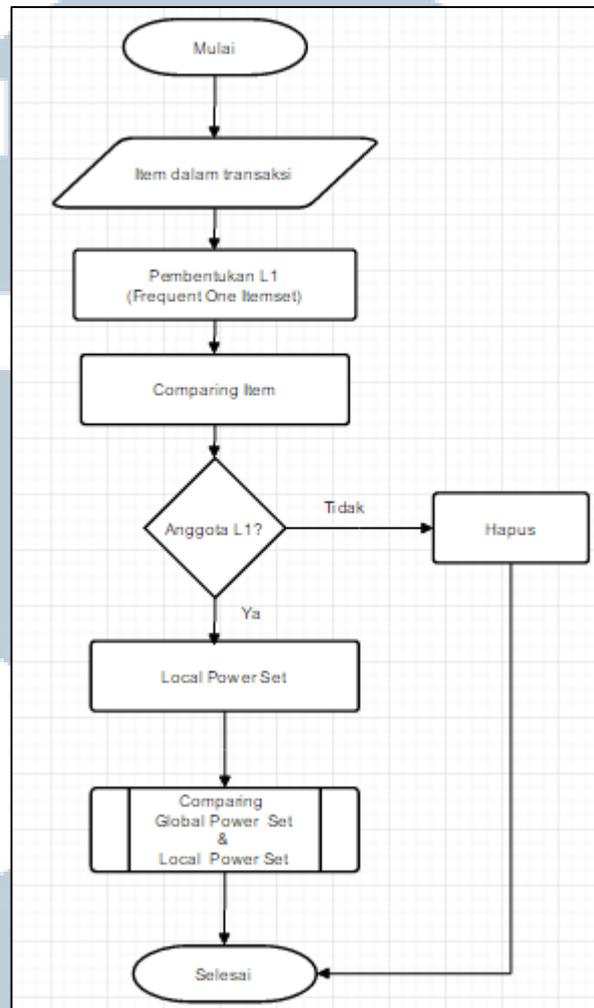
Gambar 3.14 menunjukkan pembentukan frequent itemset. Dilakukan pengecekan item yang muncul. Jika angka kemunculannya sama atau lebih dari minimum support, maka item tersebut ditetapkan *frequent 1-itemset*. Jika angka kemunculannya kurang dari minimum support, maka item tersebut dihapus. Setelah *frequent 1-itemset* terbentuk, maka dilakukan pembentukan *Global Power Set*. *Global Power Set* merupakan kandidat dari itemset. Kemudian, dilakukan pengambilan data dengan tujuan untuk membandingkan *item* yang ada dalam transaksi.



Gambar 3.15 Flowchart Pembentukan Frequent Itemset

Jika item bukan merupakan anggota *frequent 1-itemset*, maka item tersebut dihapus dari daftar transaksi tersebut. Setelah *Global Power Set* dan *Local Power*

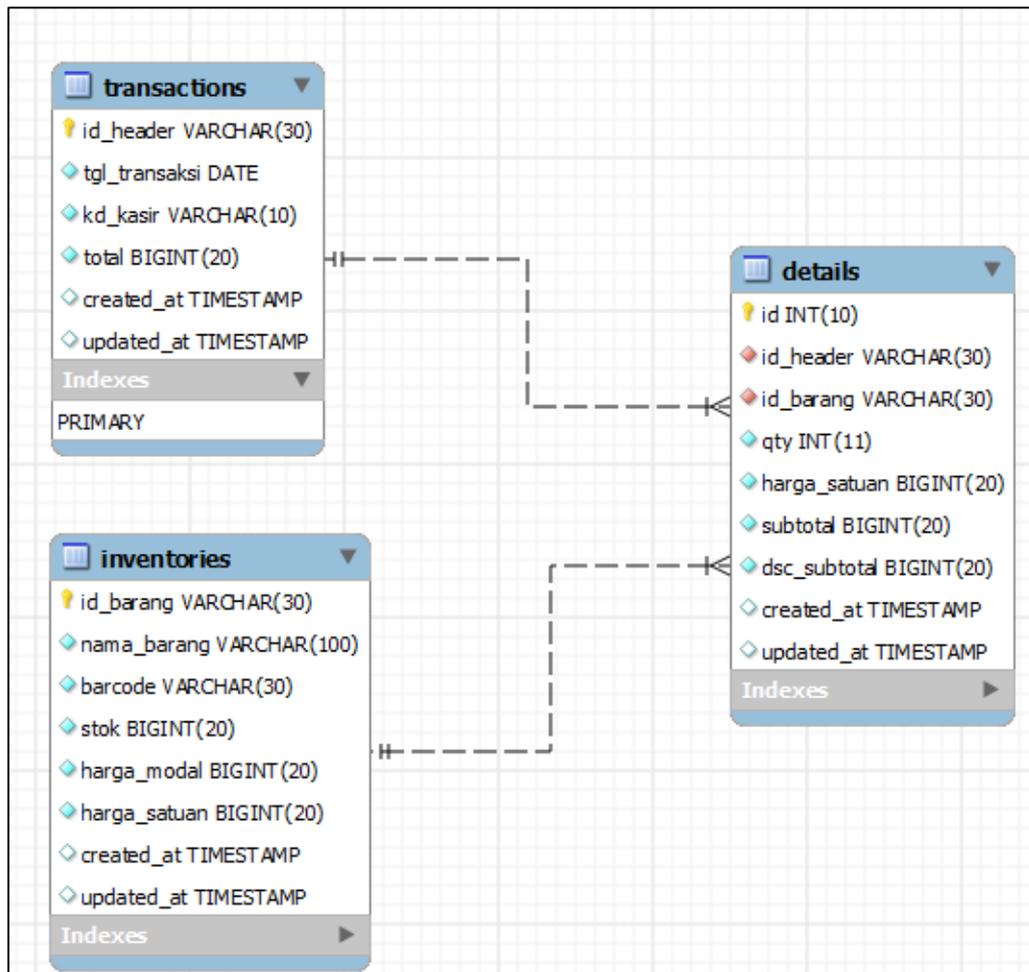
Set terbentuk, maka langkah selanjutnya adalah memasukkan *Local Power Set* ke dalam *Global Power Set* seperti yang ditunjukkan oleh Gambar 3.15



Gambar 3.16 Flowchart Memasukkan *Local Power Set* ke *Global Power*

Gambar 3.16 menunjukkan tahap jika perbandingan setiap kandidat *k-itemset* pada *Local Power Set*, maka *count* pada *Global Power Set* akan bertambah satu. Setelah kandidat *k-itemset* terbentuk dengan frekuensinya, maka dilakukan pengecekan. Jika frekuensi *k-itemset* kurang dari *minimum support*, maka kandidat *k-itemset* tersebut dihapus. Kandidat *k-itemset* yang tersisalah yang menjadi *frequent itemset* akhir yang disebut juga *strong rule*.

3.2.3 Entity Relationship Diagram



Gambar 3.17 Entity Relationship Diagram

Entity relationship diagram(ERD) dalam aplikasi ini memiliki hubungan antara tabel **transactions** dan **details** adalah *one to many*. Hal tersebut bermakna satu data tipe *transaction* bisa memiliki lebih dari satu data *transaction* dalam tabel **details** dan satu data **details** dalam tabel **details** hanya memiliki satu data tipe dalam tabel **transactions**. Hubungan antara tabel **details** dan tabel **inventories** adalah *one to many*. Hal tersebut bermakna satu data tipe **inventories** bisa memiliki lebih dari satu data **inventories** dalam tabel **details** dan satu data **details** dalam tabel **details** memiliki satu data tipe dalam tabel **inventories**.

3.2.4 Struktur Tabel

A. Tabel users

Primary key : id

Tabel 3.5 Struktur Tabel users

Attribute	Tipe	Ukuran	Keterangan
id	Int	10	Kode unik user
name	Varchar	255	Nama user
email	Varchar	255	Alamat email
password	Varchar	255	Password user
remember_token	Varchar	255	Token session
created_at	Timestamp		Waktu dibuat data
updated_at	Timestamp		Waktu data diganti

Tabel users berfungsi untuk menyimpan data yang berhubungan dengan data pengguna yang memiliki *attribute* id, name, email, password, remember_token, created_at, updated_at.

B. Tabel inventories

Primary key : id_barang

Tabel 3.6 Struktur Tabel inventories

Attribute	Tipe	Ukuran	Keterangan
id_barang	Varchar	30	Kode unik barang
nama_barang	Varchar	100	Nama barang
Barcode	Varchar	30	barcode
stok	Varchar	20	Persediaan barang
Harga_modal	BigInt	20	Harga beli
Harga_satuan	BigInt	20	Harga jual
created_at	Timestamp		Waktu dibuat data
updated_at	Timestamp		Waktu data diganti

Tabel inventories berfungsi untuk menyimpan data-data berkaitan dengan data barang yang memiliki *attribute* id, name, email, password, remember_token, created_at, updated_at.

C. Tabel transactions

Primary key : id_header

Tabel 3.7 Struktur Tabel transactions

Attribute	Tipe	Ukuran	Keterangan
id_header	Varchar	30	Kode unik struk
Tgl_transaksi	Timestamp		Tanggal transaksi terjadi
Kd_kasir	Varchar	10	Kode kasir
total	BigInt	20	Total transaksi
created_at	Timestamp		Waktu dibuat data
updated_at	Timestamp		Waktu data diganti

Tabel transaction berfungsi untuk menyimpan data-data berkaitan dengan data transaksi secara *general*. Tabel transactions memiliki *attribute* id_header, tgl_transaksi, kd_kasir, total, created_at, updated_at.

D. Tabel details

Primary key : id

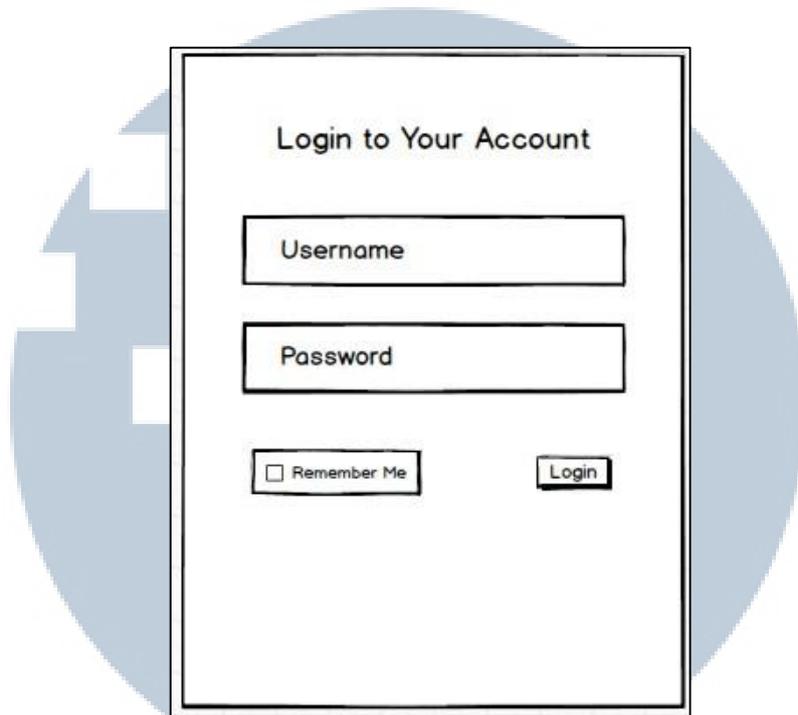
Foreign key : id_header, id_barang

Tabel 3.8 Struktur Tabel details

Attribute	Tipe	Ukuran	Keterangan
id	Int	10	Kode unik detail
id_header	Varchar	30	Kode unik struk
Id_barang	Varchar	30	Kode unik barang
qty	Int	11	Kuantitas barang
Harga_satuan	BigInt	20	Harga jual
subtotal	BigInt	20	subtotal
Dsc_subtotal	BigInt	20	Setelah di diskon
created_at	Timestamp		Waktu dibuat data
updated_at	Timestamp		Waktu data diganti

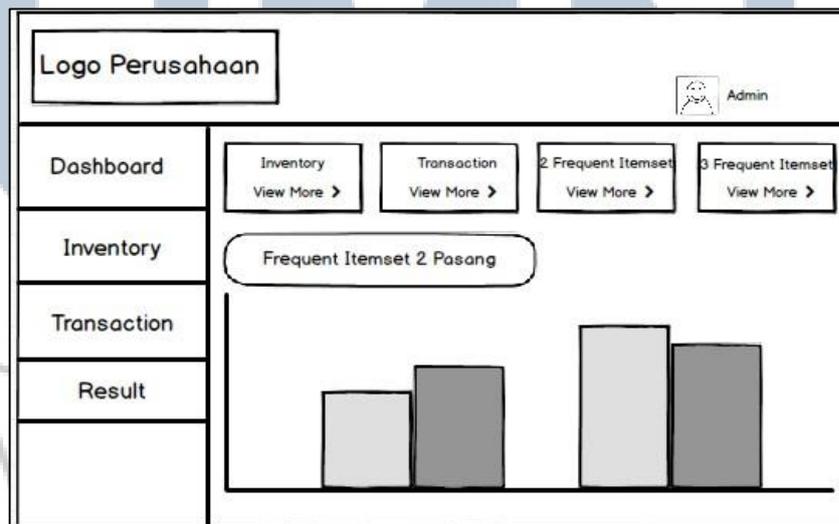
Tabel details berfungsi untuk menyimpan data-data berkaitan dengan data rincian transaksi. Tabel details memiliki *attribute* id, id_header, id_barang, qty, harga_satuan, subtotal, dsc_subtotal, created_at, updated_at.

3.2.5 Rancangan Antarmuka



Gambar 3.18 Rancangan Antarmuka Halaman Login

Gambar 3.18 menunjukkan rancang antarmuka halaman Login yang memiliki dua buah *input* yang masing-masing berfungsi untuk *username* dan *password*.



Gambar 3.19 Rancangan Antarmuka Halaman Home

Gambar 3.19 menunjukkan halaman Home yang menampilkan empat buah *button* yang terletak di atas mengarah kepada halaman Data Inventory, Data Transaction, Chart 2 Itemset, Chart 3 Itemset. Pada bagian utama halaman Home memiliki *chart* yang datanya berasal dari *cookie* hasil perhitungan *frequent itemset*.

Nama Barang	Barcode	Stok	Harga Modal	
ABC Battery A	123456789	5	3078	Edit (Delete)
ABC Battery B	234567891	0	3089	Edit (Delete)
ABC Battery B	345678912	3	4000	Edit (Delete)
Batter B	5678912345	2	3500	Edit (Delete)
Battery C	456789123	6	1500	Edit (Delete)

Gambar 3.20 Gambar Data Inventories dan Data Transaksi

Gambar 3.20 menunjukkan bahwa dalam halaman Data Inventories menampilkan data-data berkaitan dengan data barang. Dalam halaman Data Inventories memiliki empat buah kolom yaitu, nama barang, barcode, stok, harga modal. Di sebelah kanan terdapat dua buah tombol yang berfungsi untuk mengubah dan menghapus data.

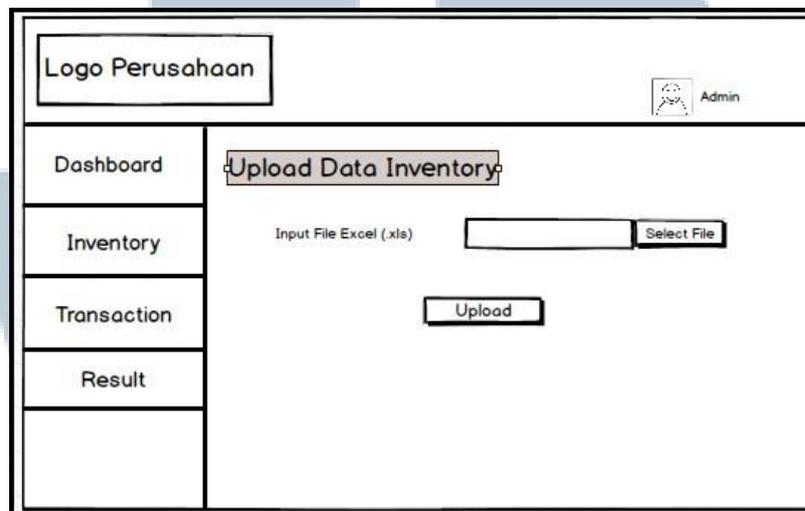
Gambar 3.21 Halaman Edit Data Inventories

Gambar 3.21 menunjukkan bahwa dalam halaman Edit Data Inventories memiliki enam komponen *input* untuk mengubah *data*, yaitu kode barang, nama barang, barcode, stok, harga modal, harga jual. Halaman Edit Data juga memiliki dua buah tombol untuk menyimpan data atau membatalkan perintah. Halaman Edit Inventories memiliki kemiripan dengan halaman Insert Inventories.

Nama Barang	Satuan	Jumlah	Subtotal	
	2000	2	4000	x del
Total			: Rp 4000	

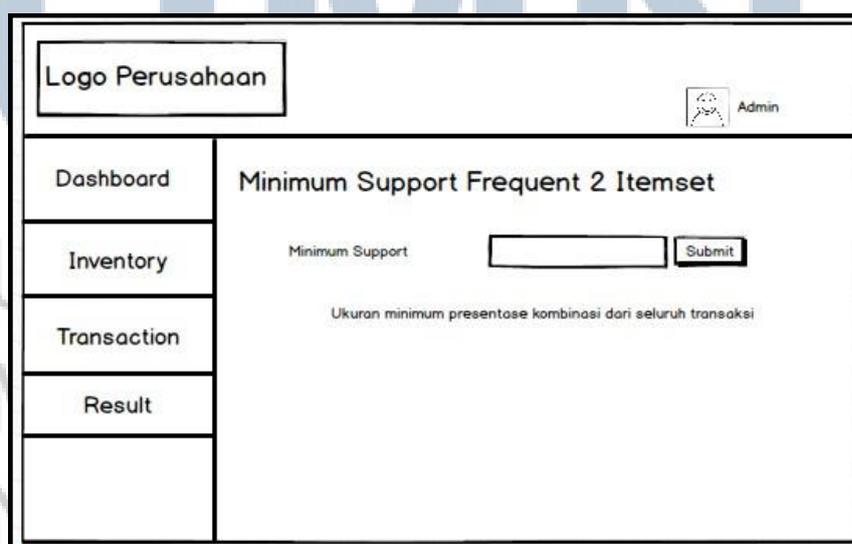
Gambar 3.22 Halaman Insert Transaction

Gambar 3.22 menunjukkan bahwa halaman Insert Transaction memiliki *input* berupa kode struk, kode kasir, tanggal transaksi, nama barang, jumlah barang serta total transaksi.



Gambar 3.23 Halaman Upload File Excel

Gambar 3.23 menunjukkan bahwa halaman Upload File Excel digunakan untuk halaman Upload Data Inventory dan Upload Data Transaction. Dalam halaman tersebut terdapat tombol *select file* untuk memilih *file excel* yang dituju. Kemudian, setelah selesai tekan tombol Upload untuk memulai proses *input* data.



Gambar 3.24 Halaman Insert Minimum Support

Gambar 3.24 menunjukkan bahwa halaman Insert Minimum Support memiliki satu buah input untuk memasukkan nilai *minimum support*. Kemudian memiliki tombol *submit* untuk langsung memproses *generated frequent itemset*.

