



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

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam kegiatan kerja magang ini mendapat kedudukan sebagai *desktop programmer* yang bertugas merancang alur penyimpanan dan pengambilan data dari *desktop application* ke *database*. Juga mendapat tugas untuk membuat rancangan *database*, *user interface*, dan sistem *privilege/access control* untuk membatasi hak akses tiap kelompok *user* sesuai dengan arahan dari bapak Anton Hermanto sebagai pembimbing lapangan.

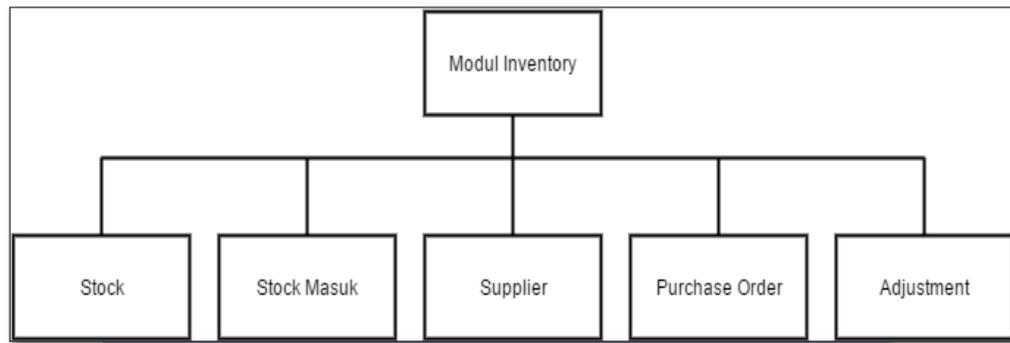
Koordinasi dilakukan dengan bertemu langsung saat di kantor atau melalui *LINE Messenger* apabila beliau sedang tidak berada di kantor. Dalam prosesnya, progress akan dilaporkan setiap harinya langsung sehingga jika terdapat revisi langsung diinfokan.

3.2 Tugas yang dilakukan

Selama pelaksanaan kerja magang, tugas utama yang diberikan adalah untuk membuat suatu sistem *Point Of Sales (POS)*, yaitu sistem yang bertujuan untuk membantu pemilik dalam melayani pemesanan transaksi, pemeliharaan data, penampilan transaksi, dan pembuatan laporan, sistem ini juga harus bisa mencetak *invoice*, dan membuat laporan terkait penjualan-penjualan yang telah dilakukan.

Pembangunan sistem *Point Of Sales (POS)* ini dilakukan seorang diri, dimana berikut adalah deskripsi tugas yang dilakukan yaitu:

1. Merancang *database* dari aplikasi yang memuat data-data *employee*, bahan baku menu, kategori menu, menu, detail menu, pesanan, detail pesanan, *purchase order*, stok *adjustment*, *supplier*, resep material, penambahan stok, struk.
2. Membuat *Content Management System (CMS)* yang terdiri dari beberapa modul :
 - a. Modul *User Privilege Control*, berfungsi untuk membuat *user group* baru dan mengatur *privilege/hak akses user group* tersebut dalam mengoperasikan sistem *Point Of Sales* ini.
 - b. Modul *Cashier* berfungsi untuk menampilkan menu yang tersedia, memilih menu, menyimpan menu yang dipesan oleh pelanggan, mencatat nama pelanggan dan tanggal transaksi penjualan, serta dapat mencetak struk hasil transaksi.
 - c. Modul *Material* berfungsi untuk menampilkan, menambah, mengedit, dan menghapus bahan baku yang tersedia.
 - d. Modul *Menu* berfungsi untuk membuat menu, menambah material atau bahan baku ke dalam menu, dan membagi setiap menu ke dalam menu *category* seperti *Foods*, *Beverages*, dan *Packages*, serta dapat mengedit dan menghapus menu yang tersedia.
 - e. Modul *Inventory*, di dalam modul *inventory* terdapat beberapa sub-menu lagi seperti *Stok*, *Stok Masuk*, *Supplier*, *Purchase Order*, dan *Adjustment*. Dimana masing – masing sub menu memiliki fungsi sebagai berikut :



Gambar 3.1 Modul Inventory

1. *Stok* berfungsi untuk menampilkan hasil proses penambahan dan pengurangan stok yang dilakukan.
 2. *Stok masuk* berfungsi untuk menambah stok, menampilkan stok dan mencatat staf yang melakukan penambahan stok.
 3. *Supplier* berfungsi untuk menyimpan data – data *supplier* yang sudah bekerja sama.
 4. *Purchase order* berfungsi untuk mencatat pemesanan pembelian suatu material ke *supplier* yang tersedia.
 5. *Adjustment* berfungsi untuk mencatat transaksi penambahan ataupun pengurangan material yang ada sesuai dengan kondisi yang terjadi.
- f. Modul *Report* di dalam modul *report* terdapat beberapa sub-menu lagi seperti Analisa penjualan yang berfungsi menampilkan analisa penjualan harian. Data transaksi yang berfungsi menampilkan lebih detail lagi data penjualan. Laporan penjualan per-pembayaran berfungsi menampilkan data pembayaran yang dilakukan melalui tunai atau menggunakan kartu.

3.3 Uraian Pelaksanaan Kerja Magang

Berikut adalah tahap – tahap dari pelaksanaan kerja magang yang dikerjakan setiap minggunya.

3.3.1 Proses Pelaksanaan

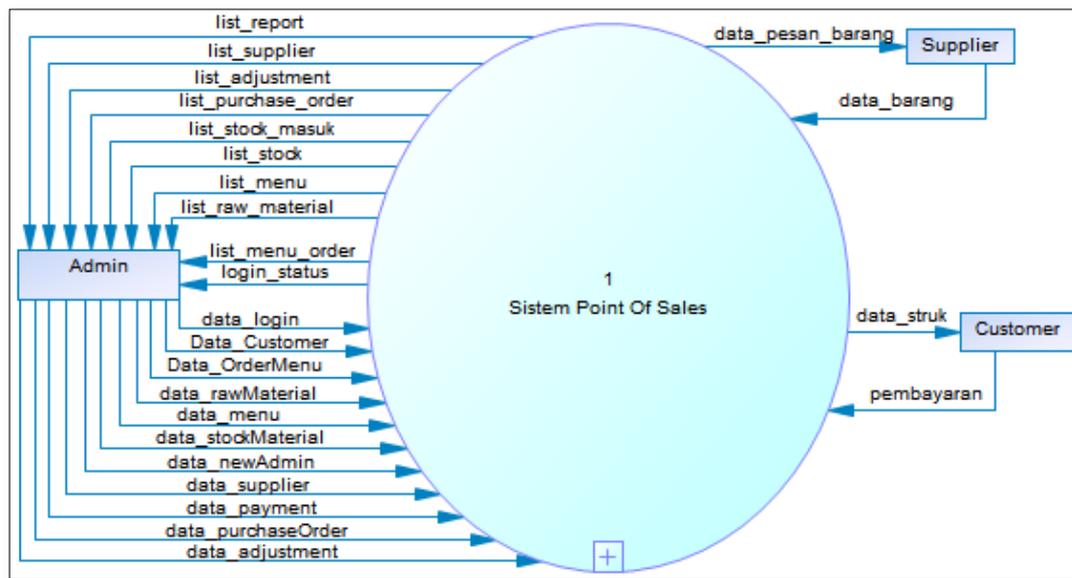
Sistem *Point Of Sales (POS)* ini dibangun dengan konsep *windows form-based*, sehingga dalam pembangunannya digunakan bahasa pemrograman *C#*. *Database* yang digunakan adalah *MySQL*. Pada minggu awal pelaksanaan magang diberikan sosialisasi peraturan yang berlaku dalam perusahaan, perkenalan dengan para pegawai yang bekerja di *Steak@Town* dan sosialisasi proyek yang akan dikerjakan.

Setelah tahap perkenalan dan pembelajaran selesai, Pak Anton Hermanto selaku pembimbing magang dan *project manager* mulai menjelaskan mengenai program yang akan dibuat, setelah peserta magang memahami proses bisnis dari program yang akan dibuat, Pak Anton mulai memberikan tugas bagi peserta magang.

A. Data Flow Diagram (DFD)

Data Flow Diagram adalah sebuah grafik yang merepresentasikan alur dari data melalui sebuah sistem informasi, pemodelan aspek prosesnya. Sebuah DFD biasa digunakan sebagai langkah awal untuk membuat gambaran dari sistem secara menyeluruh, yang nantinya bisa diuraikan.

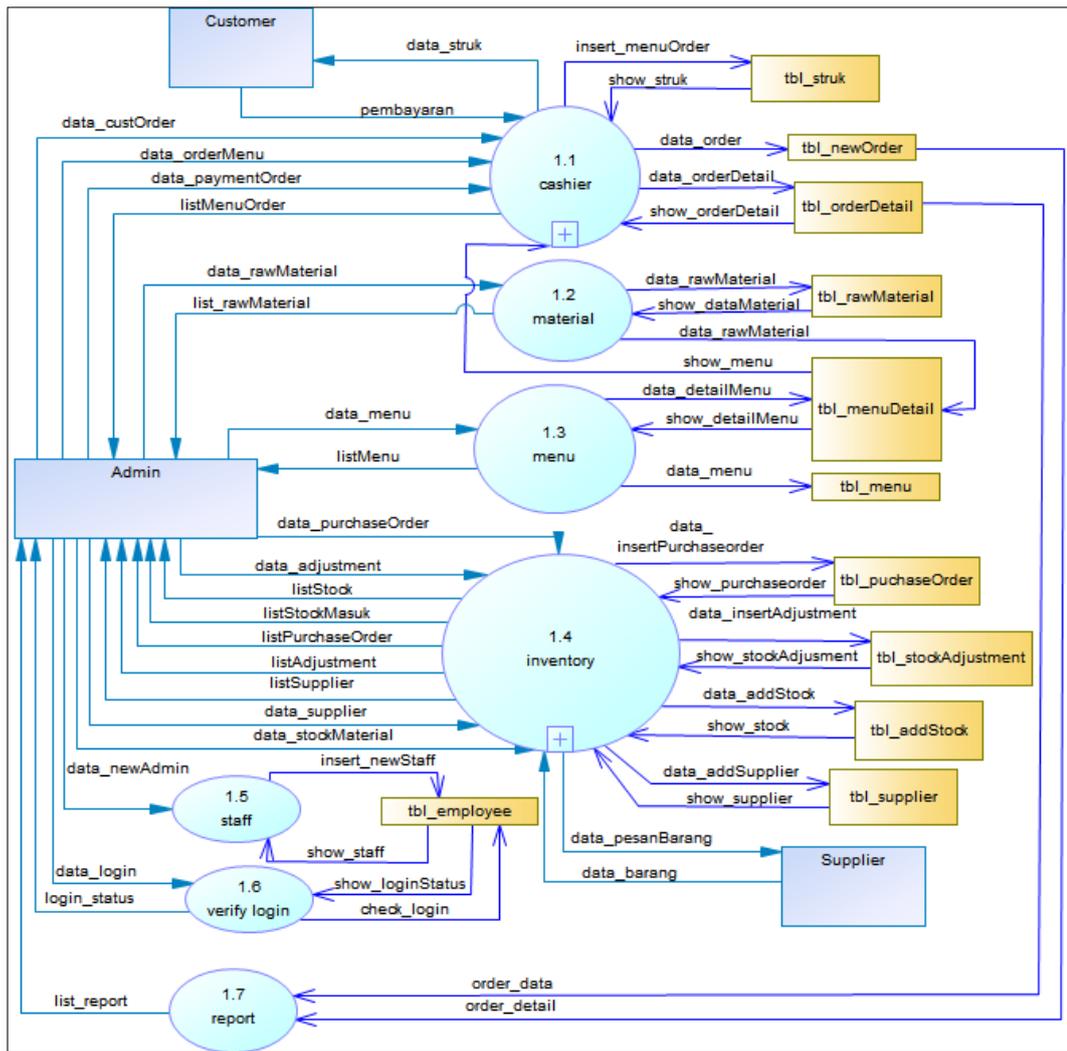
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.2 Context Diagram Sistem Point Of Sales

Gambar 3.2 merupakan Context Diagram Sistem Point Of Sales . Di dalamnya terdapat 1 entitas, yaitu admin. Admin mengirim data ke dalam sistem berupa *data_login*, *data_customer*, *data_OrderMenu*, *data_rawMaterial*, *data_menu*, *data_stokMaterial*, *data_Supplier*, *data_paymentOrder*, dan *data_newAdmin*. Sedangkan dari sistem akan mengirim data berupa *listReport*, *listSupplier*, *listAdjustment*, *listPurchaseOrder*, *listStokMasuk*, *listStok*, *listMenu*, *listMaterial*, *listMenuOrder*, dan *login status*.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.3 Diagram Level 1 Sistem Point Of Sales

Pada diagram *Level 1* pada Gambar 3.3 terdapat 7 proses utama, yaitu *cashier* yang menerima data *data_custOrder* berupa nama dan nomer meja, data *data_OrderMenu* berupa nama – nama menu yang dipesan oleh *customer*, lalu menyimpan data tersebut ke dalam *tbl_newOrder*. Di dalam *cashier* juga terdapat proses untuk mengolah struk yang nantinya akan diberikan kepada *customer*.

Kedua adalah proses *material* dimana di dalam proses ini menerima *data_material* berupa nama, harga, *category* dan resep jika diperlukan. Nama,

harga dan *category* akan disimpan ke dalam *tbl_RawMaterial*. Di dalam proses ini juga akan menampilkan *list raw material* yang telah di *input*.

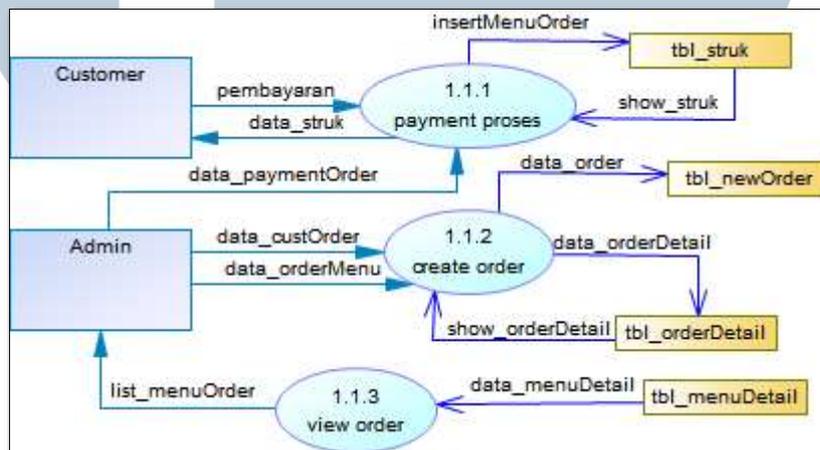
Ketiga adalah proses menu yang merupakan proses CRUD(*Create, Read, Update, Delete*) dimana di dalam proses menu ini hanya menerima *data_menu* dari admin yang berupa nama menu, harga, *category*, dan bahan – bahan dari menu tersebut, setelah itu di dalam proses ini akan langsung menyimpan nama, menu, *category* dan bahan – bahan tersebut ke dalam *tbl_menu* dan *tbl_menuDetail*. Di dalam *tbl_menuDetail* sendiri membutuhkan data – data *raw material* dari proses material.

Keempat adalah proses inventory yang merupakan proses CRUD(*Create, Read, Update, Delete*) untuk transaksi *inventory*. Proses ini menerima masukan dari admin berupa *data_supplier* dan *data_stokMaterial* yang berisikan jumlah stok yang di tambah melalui proses *add stok, adjustment, dan purchase order*. Proses ini akan menyimpan transaksi *purchase order* ke dalam *tbl_purchaseOrder*, transaksi *adjustment* ke dalam *tbl_stokAdjustment*, transaksi *add stok* ke dalam *tbl_addStok* dan transaksi tambah *supplier* ke dalam *tbl_supplier*. Proses ini menampilkan data ke admin berupa *listStok, listStokMasuk, listPurchaseOrder, listAdjustment*, dan data *list supplier*.

Kelima adalah proses staff yang merupakan proses CRUD(*Create, Read, Update, Delete*) untuk proses staff. Proses ini menerima masukan berupa *data_newAdmin* yang berisikan nama admin, nomor telpon, *email*, alamat, *password* dan status dari admin. Di dalam proses ini dapat menyimpan data yang diinput langsung ke dalam *tbl_employee*.

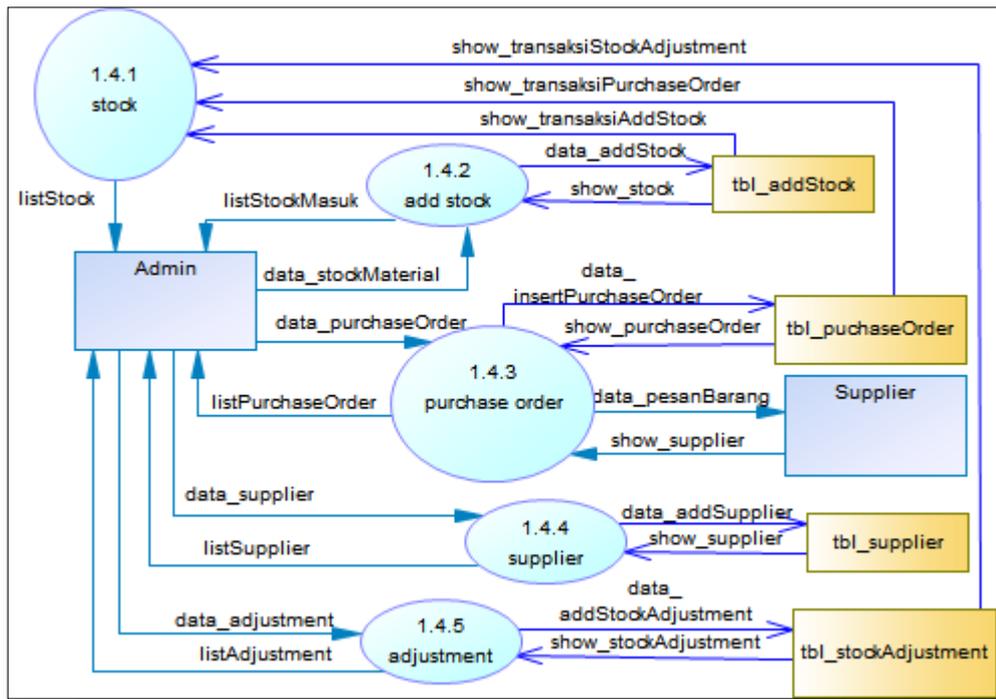
Keenam adalah proses *verify login* yang memproses verifikasi *login user*, proses ini menerima *username* dan *password* dan lalu mencocokkan dengan *table tbl_employee*. Apabila data yang diterima cocok dengan data yang ada di dalam *database* maka proses ini akan mengirim *login_status*.

Ketujuh adalah proses *report* yang memproses *report* yang nantinya akan di tampilkan ke admin berupa *list report*. Di dalam proses ini menerima data berupa *data_order* dan *order_detail* dari proses material yang nantinya akan di olah menjadi *report*.



Gambar 3.4 Diagram *Level 2* Proses *Payment*

Pada gambar 3.4 terdapat tiga proses yakni *payment proses*, *create order*, dan *view order*. *Payment proses* menunjukan proses penampilan total biaya makan yang harus di bayar *customer* dengan sebelumnya *customer* harus memberi masukan jenis pembayaran apa yang akan dilakukan, apakah *debit* ataupun *cash*. Proses *create order* untuk menerima data berupa pesanan *customer* yang akan disimpan di dalam *tabel new order* dan *order detail*. Selanjutnya adalah *view order* adalah untuk meihat *order* yang sebelumnya telah dipesan oleh *customer*.

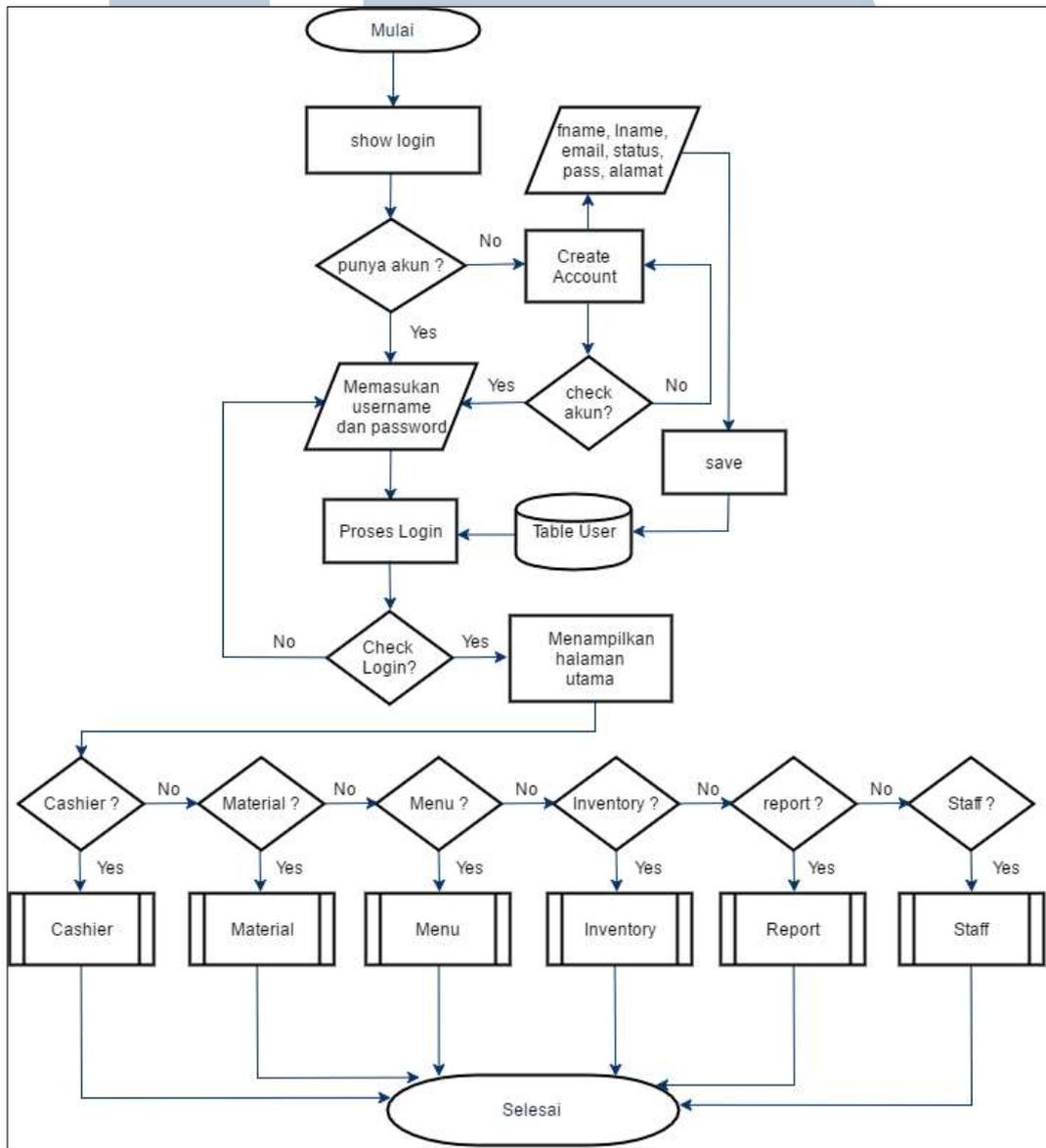


Gambar 3.5 Diagram Level 2 proses management stock

Pada gambar 3.5 terdapat lima proses yaitu *stock*, *add stock*, *purchase order*, *supplier*, dan *adjustment*. Proses *stock* untuk menampilkan hasil transaksi dari proses penambahan *stock*, *purchase order*, dan *adjustment*. Proses *add stock* untuk menambah dan mengedit jumlah *stock*. Proses *purchase order* untuk melakukan transaksi *purchase order* ke *supplier* dan untuk menerima barang dari *supplier*. Proses *supplier* untuk pencatatan *supplier* yang akan menjalin kerja sama pembelian barang. Terakhir adalah proses *adjustment* yang dilakukan untuk menyesuaikan jumlah barang yang ada disistem dengan jumlah barang yang ada digudang dengan memasukkan data *adjustment* dan data tersebut akan langsung dimasukkan ke dalam *database* yang nantinya akan di tampilkan dalam bentuk list transaksi *adjustment*.

B. Flowchart

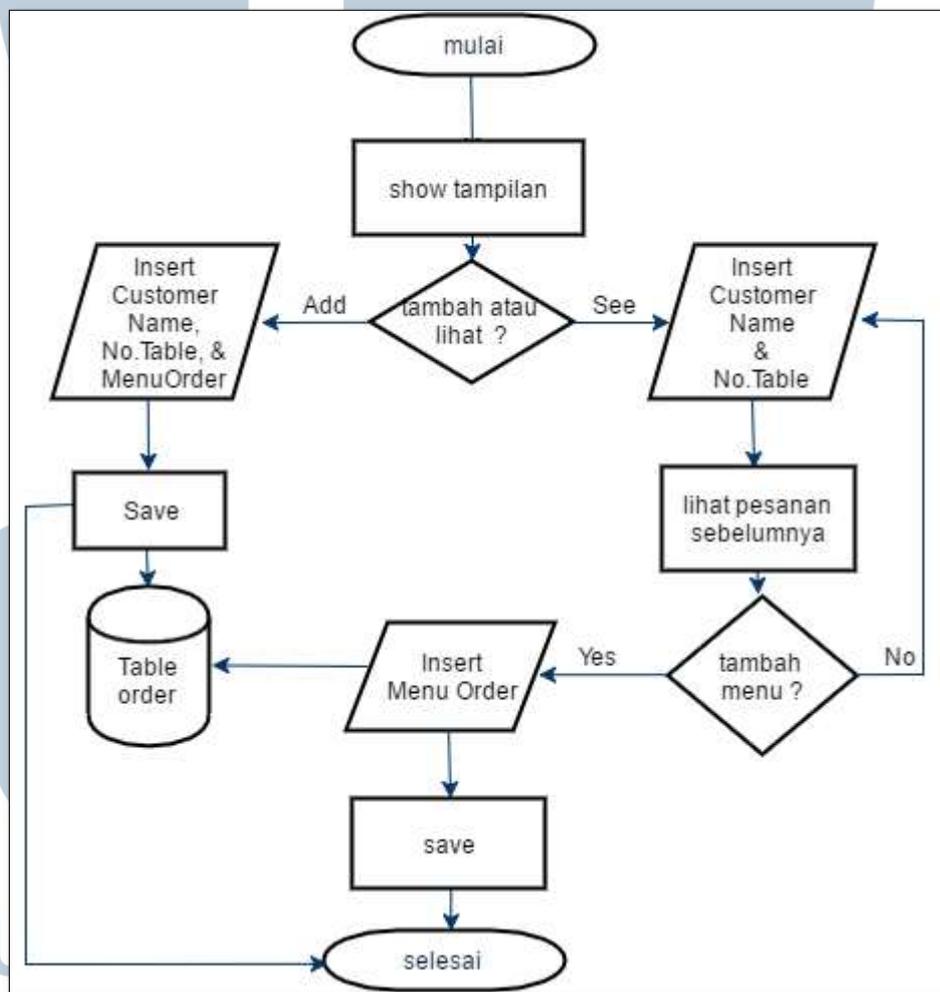
Flowchart atau Bagan aliran adalah bagan yang menunjukkan aliran di dalam program atau prosedur sistem secara logika. Bagan aliran digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.



Gambar 3.6 Flowchart Sistem

Gambar 3.6 Merupakan *Flowchart System* pada sistem *Point of Sales*. Setiap user yang menggunakan sistem ini wajib untuk memasukkan *username*

serta *passwords* yang sesuai dengan data yang ada pada *database*. Bila staf tidak memiliki *account*, maka staf akan dibuatkan *account* oleh admin. Setelah itu staf dapat melakukan proses *login* dengan cara memasukkan *username* dan *password* pada halaman depan. Setelah itu, data *login* akan dicocokkan ke dalam table *user* yang berada di *database*. Bila data *login* tersebut cocok, maka selanjutnya sistem akan menampilkan halaman utama. Pada halaman utama terdapat enam submenu seperti menu *cashier*, *material*, *menu*, *inventory*, *report*, dan *staff*.



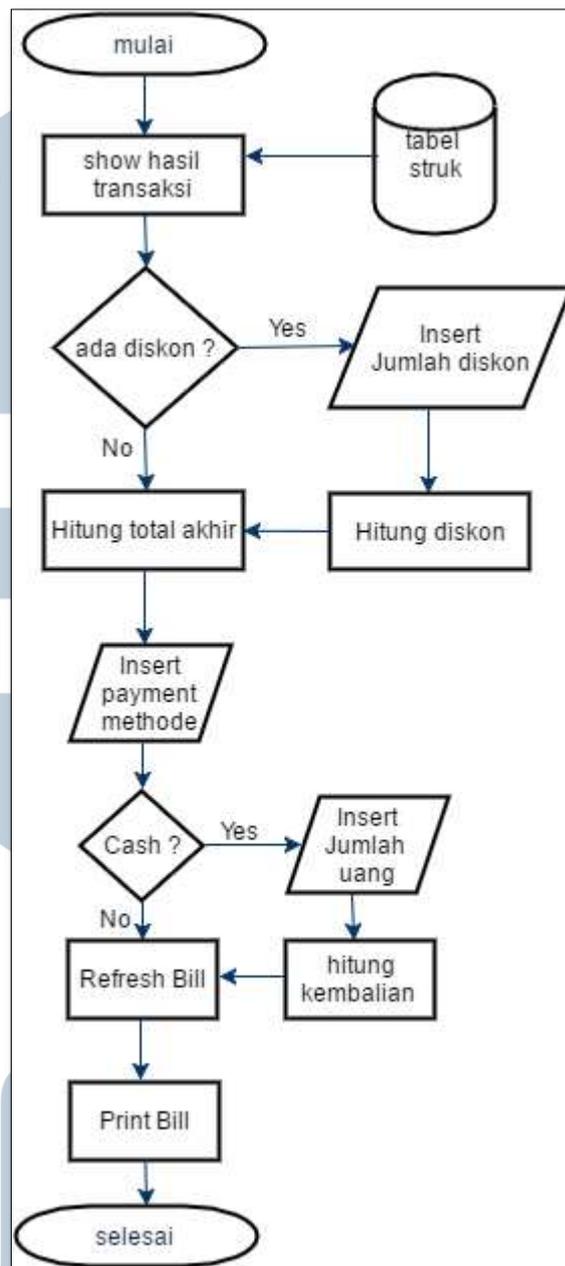
Gambar 3.7 Flowchart Cashier

MULTIMEDIA
NUSANTARA

Gambar 3.7 adalah *Flowchart Cashier*. Pada alur ini admin dapat menentukan apakah dia akan menambah pesanan dari pelanggan baru atau melihat pesanan sebelumnya. Bila ingin melakukan pesanan dari pelanggan baru, maka admin dapat menekan tombol *New Order*, kemudian memasukkan nama pelanggan beserta nomer mejanya. Setelah itu, admin akan memasukkan menu yang dipesan oleh pelanggan. Bila telah selesai, pesanan harus disimpan untuk kemudian secara otomatis tersimpan pada *table order* pada *database*.

Bila admin ingin melihat pesanan dari pelanggan sebelumnya, admin hanya perlu memasukkan nama pelanggan, kemudian akan ditampilkan pesanan dari pelanggan yang telah dipilih. Bila ingin menambahkan pesanan tambahan, admin tinggal memasukkan *menu order* baru, kemudian *table order* pada *database* akan diupdate setelah admin menekan tombol *save*.

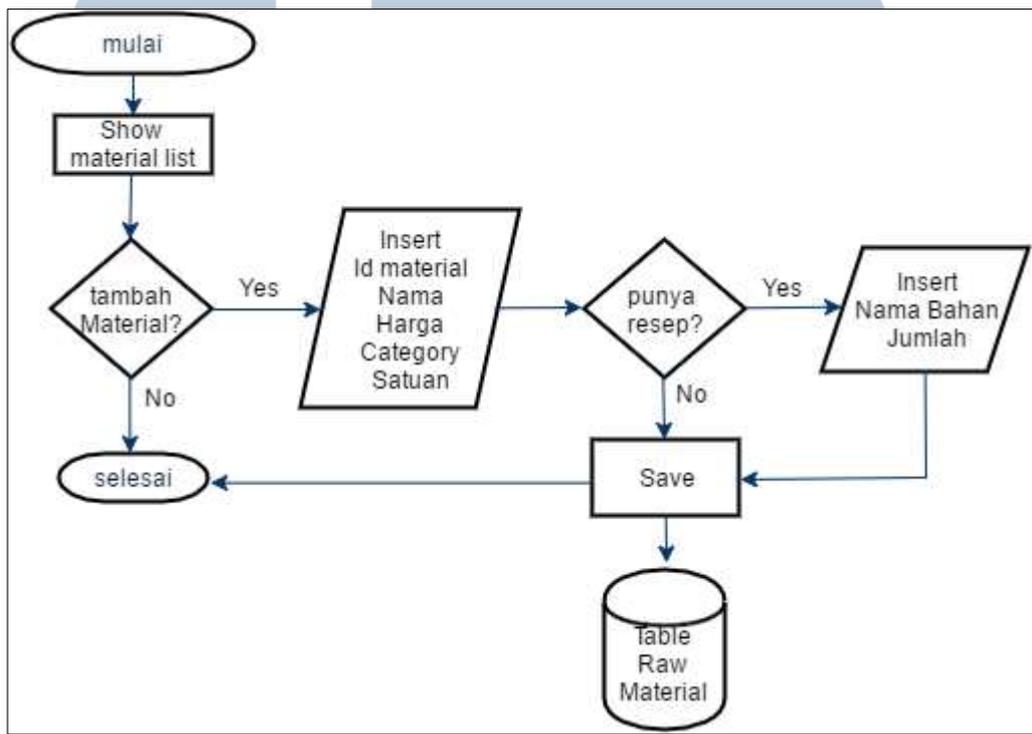




Gambar 3.8 Flowchart Payment

Gambar 3.8 merupakan *Flowchart Payment*. Alur ini terjadi setelah admin menekan tombol *Print* pada halaman *cashier*. Apabila terdapat diskon, maka admin akan memasukkan jumlah diskon, kemudian sistem akan menghitung jumlah diskon dan total akhir. Bila tidak ada diskon, maka sistem akan menampilkan total pembayaran. Setelah itu, admin akan memasukkan jenis

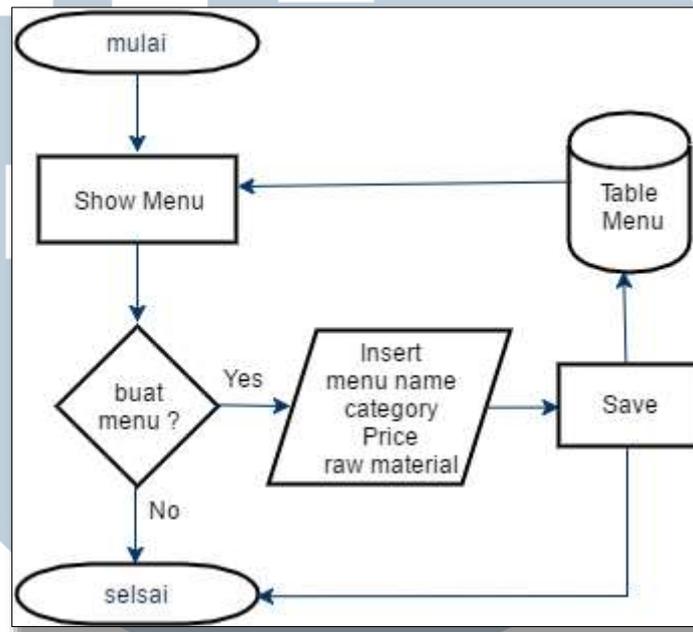
pembayaran yang akan dilakukan oleh pelanggan. Bila pelanggan menggunakan uang tunai, maka admin harus memasukkan jumlah uang yang diberikan pelanggan, kemudian *refresh bill*. Bila menggunakan kartu kredit ataupun kartu debit, maka admin akan melakukan *refresh bill*. Setelah itu, *bill* dapat dicetak.



Gambar 3.9 Flowchart Material

Pada Gambar 3.9, ditunjukkan sebuah *Flowchart* Material dimana alur pada menu Material terjadi. Pada alur ini setelah admin menekan tombol menu Material, maka akan diperlihatkan *Material List*. Bila admin ingin menambahkan material baru, maka admin harus menekan tombol *Tambah Produk*, kemudian memasukkan ID material, nama material, serta harga satuan. Setelah itu, apabila material tersebut memiliki resep, maka admin harus mencentang pilihan *Punya Resep*, bila tidak maka admin harus menekan tombol *Save*, sehingga akan tersimpan pada table *raw material* pada *database*. Kemudian, apabila admin

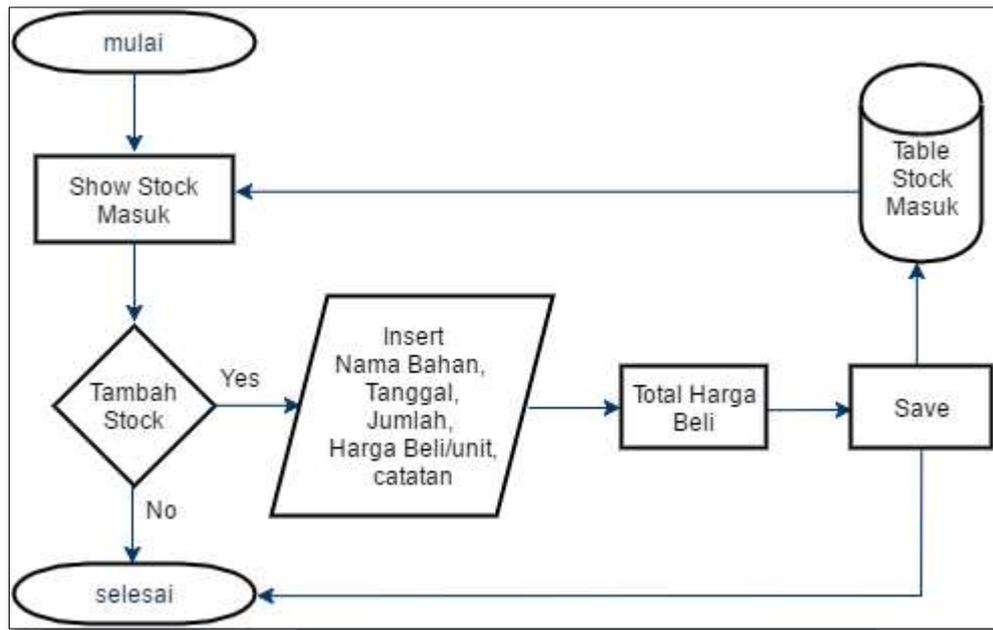
mencentang pilihan *Punya Resep*, maka admin harus memasukkan nama bahan dari material tersebut beserta jumlahnya. Setelah itu, material baru disimpan ke dalam tabel *raw material* pada *database*.



Gambar 3.10 Flowchart Menu

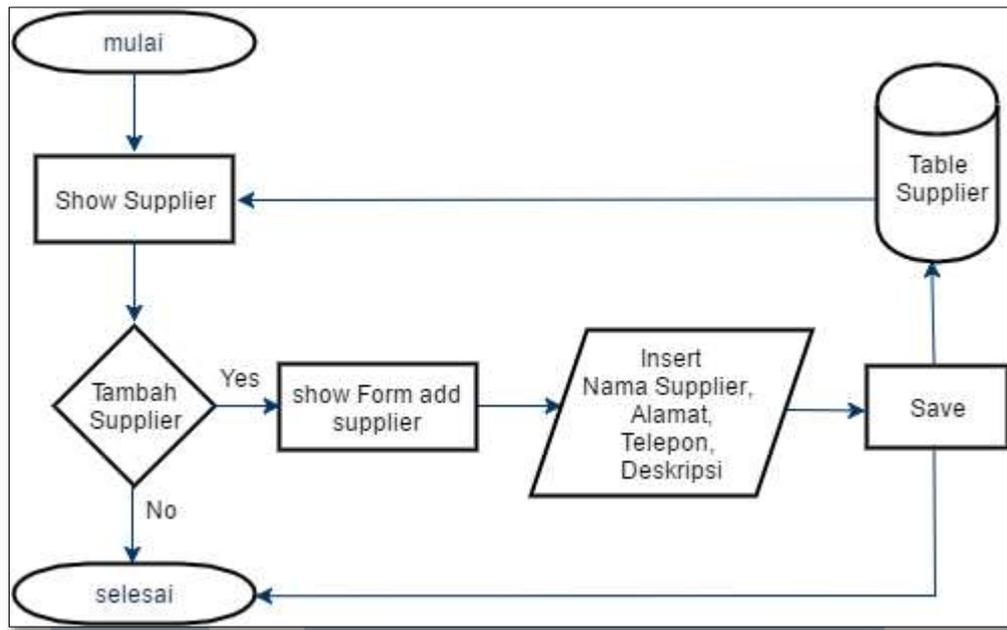
Pada Gambar 3.10 memperlihatkan sebuah *Flowchart* Menu. Alur yang terjadi ketika admin menekan tombol menu adalah sistem akan menampilkan menu yang telah ada. Bila admin ingin menambahkan menu, maka admin harus memasukkan nama menu, kategori, harga menu, serta bahan baku yang dibutuhkan untuk membuat menu tersebut. Setelah itu menu disimpan ke dalam tabel menu dalam *database*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



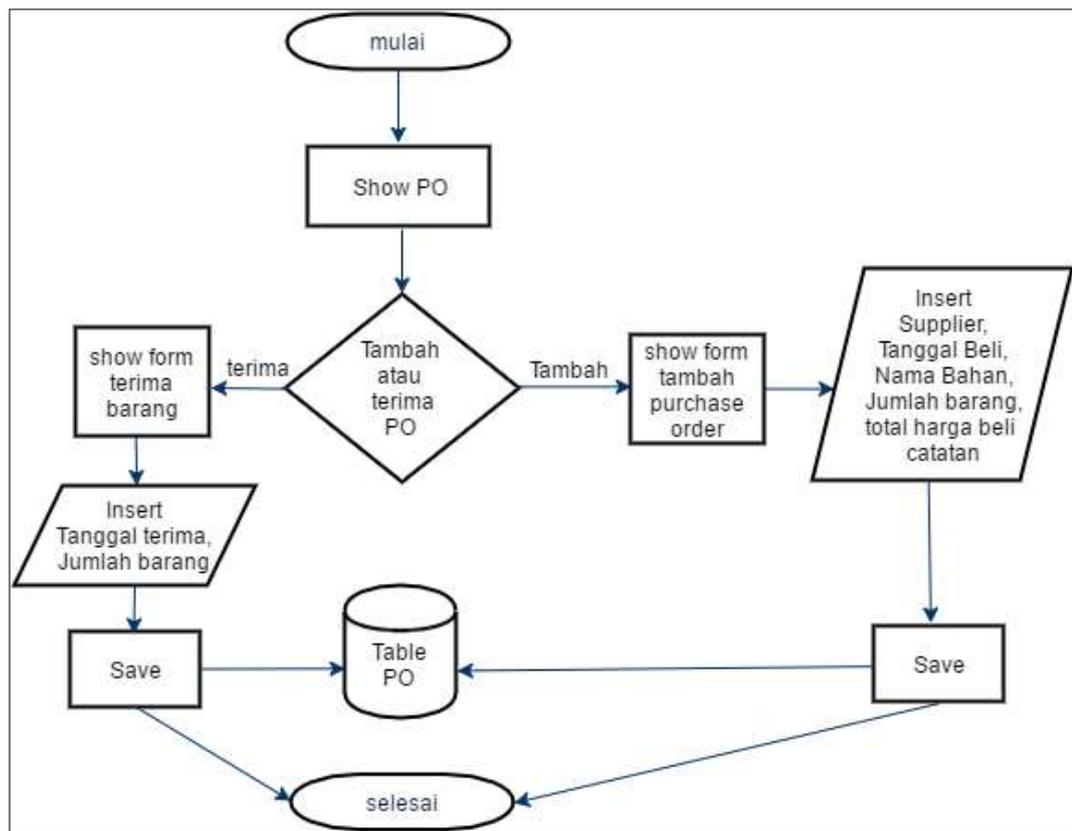
Gambar 3.11 Flowchart Stok Masuk

Gambar 3.11 menunjukkan *Flowchart* Stok Masuk. Pada alur ini, pertama kali admin menekan menu *Stok Masuk*, maka akan ditampilkan *history list* stok masuk sebelumnya. Apabila admin ingin memasukkan stok baru, maka admin harus memasukkan nama bahan, tanggal stok masuk, jumlah stok, harga beli per unit, serta catatan setelah menekan tombol *Add Stok*. Kemudian sistem akan secara otomatis mengalkulasikan total harga beli. Setelah itu admin akan menekan tombol *Save*, lalu secara otomatis data akan tersimpan ke dalam tabel *Stok Masuk* pada *database*.



Gambar 3.12 FlowChart *Supplier*

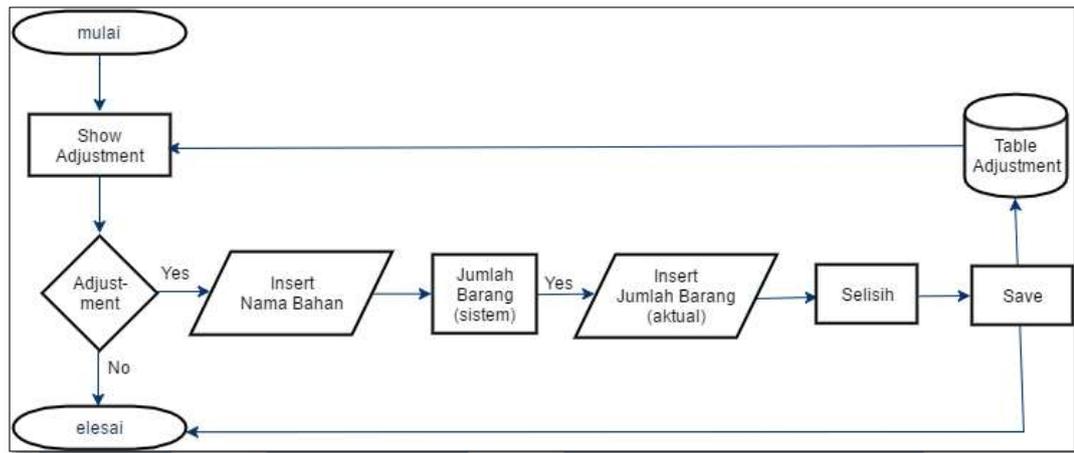
Pada gambar 3.12 menunjukkan *flowchart supplier*. Pada alur ini proses akan menunjukan *list supplier*, dimana jika admin ingin menambah *supplier* maka admin harus menekan tombol tambah *supplier* pada menu, selanjutnya admin harus memasukkan data *supplier* seperti nama *supplier*, alamat, telepon, dan deskripsi dari *supplier* tersebut, setelah itu jika ingin menyimpan maka admin harus menekan tombol *save* dan data otomatis akan tersimpan ke dalam *database*.



Gambar 3.13 Flowchart *Purchase Order*

Pada gambar 3.13 menunjukkan *flowchart Purchase Order*. Pada alur ini akan ditunjukkan proses transaksi *purchase order* yang sudah pernah dilakukan. Ketika admin ingin menambah *transaksi purchase order* maka admin harus menekan tombol tambah *purchase order* di dalam menu. Setelah itu admin harus memasukkan data berupa nama *supplier*, tanggal beli, nama bahan, dan jumlah barang yang akan dibeli, setelah itu untuk menyimpan data –data yang telah diinput maka admin harus menekan tombol *submit* dan data akan secara langsung tersimpan ke dalam *database Purchase Order*.

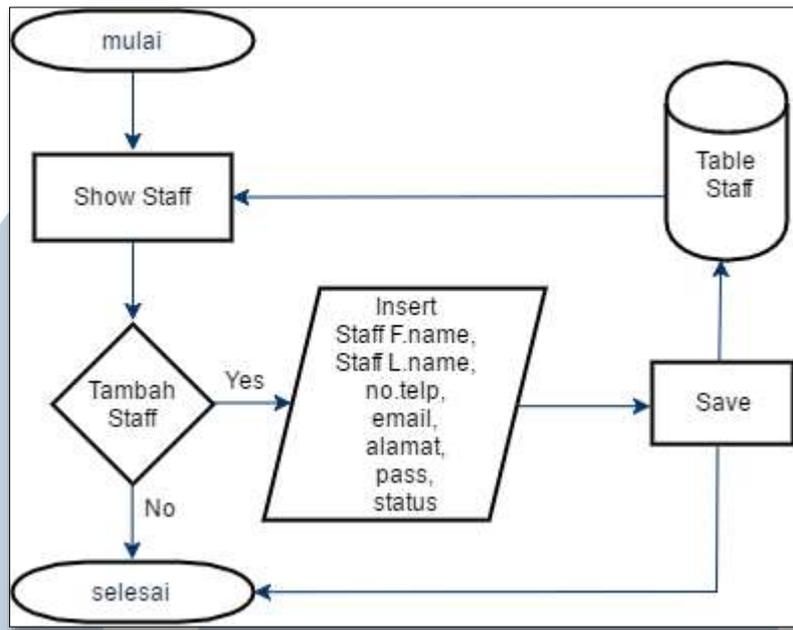
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.14 Flowchart *adjustment*

Pada gambar 3.14 menunjukkan alur dari *flowchart adjustment*. Pada proses ini akan menunjukkan proses transaksi *adjustment*. Proses *adjustment* adalah proses penambahan stok tanpa harus membeli atau memesan ke *supplier*, proses ini bertujuan untuk menyesuaikan stok yang terdapat pada gudang, jika terdapat barang rusak ataupun barang yang jumlahnya tidak sesuai dengan sistem maka sistem akan langsung mengupdate stok sesuai dengan data stok aslinya.

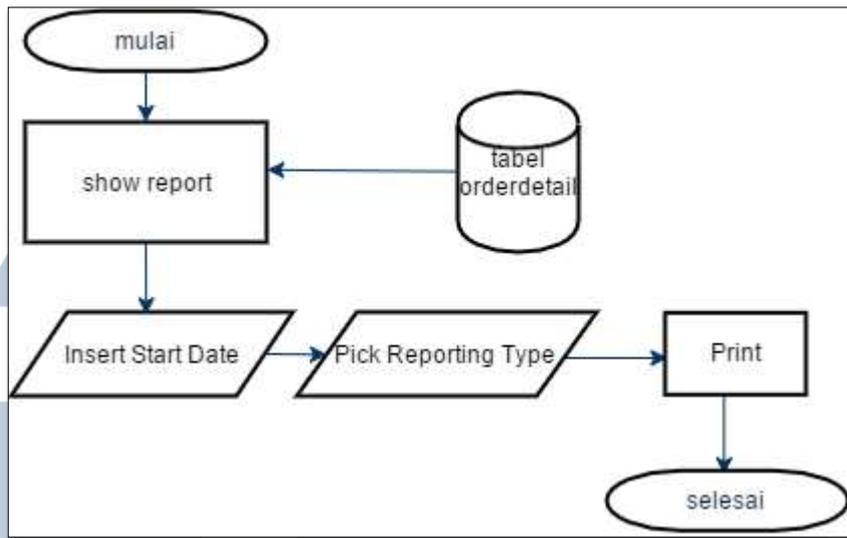
Proses utama dalam proses ini adalah menunjukan transaksi *adjustment* yang telah dilakukan sebelumnya. Jika admin ingin menambah transaksi maka admin dapat menekan tombol *adjustment* yang terdapat di dalam tampilan *adjustment* setelah itu admin harus mengisi data berupa nama bahan, dan jumlah barang yang sesuai dengan stok aslinya, setelah itu sistem akan menghitung jumlah selisih stok, untuk menyimpan data yang telah dimasukkan maka admin harus menekan tombol *submit* dan data secara langsung akan menyimpan ke dalam *database*.



Gambar 3.15 Flowchart Staff

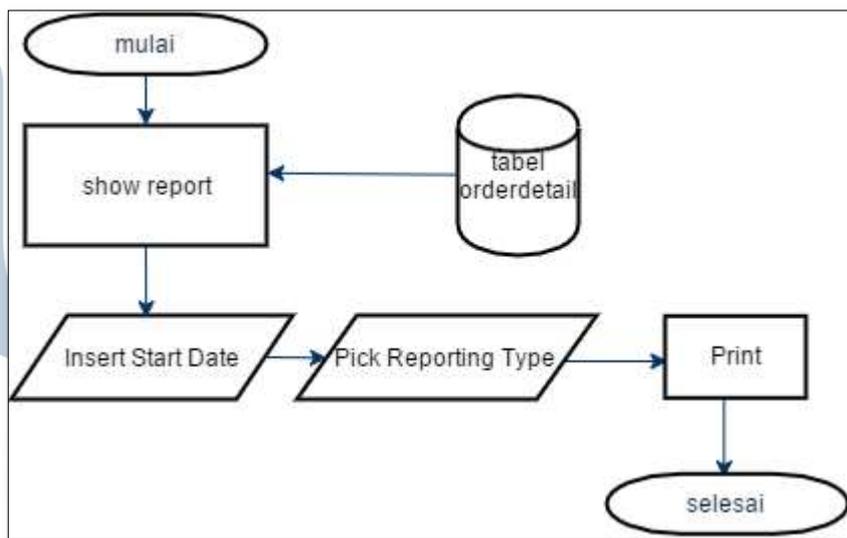
Pada gambar 3.15 menunjukkan alur dari *Flowchart staff*. Alur dari proses ini pertama akan di tampilkan data *staff* yang telah mendaftarkan diri. Jika ada *staff* yang ingin mendaftarkan diri maka *staff* harus menginput nama depan, nama belakang, nomor telpon, *email*, alamat, *password*, dan status dari *staff* tersebut, setelah itu admin dapat menyimpan data dengan menekan tombol *save* dan data langsung tersimpan ke dalam sistem.





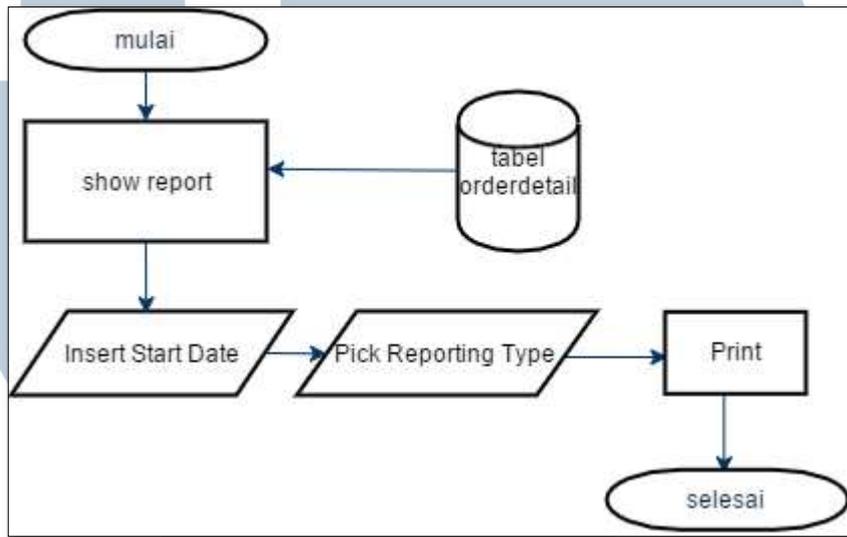
Gambar 3.16 Flowchart laporan analisa penjualan

Pada gambar 3.16 menunjukkan alur laporan analisa penjualan dimana di dalam proses ini akan menunjukkan data – data penjualan seperti jumlah penjualan perhari, total nominal perhari, dan rata – rata nominal perhari yang di dapatkan dari pihak *restaurant*.



Gambar 3.17 Flowchart analisa data transaksi

Pada Gambar 3.17 menunjukkan alur dari analisa data transaksi dimana di dalam proses ini akan menampilkan detail data dari transaksi seperti tanggal transaksi, *id order*, nama pelanggan, nomor meja, menu yang dipesan, subtotal, biaya *service*, biaya pajak, diskon dan total transaksi yang dilakukan.

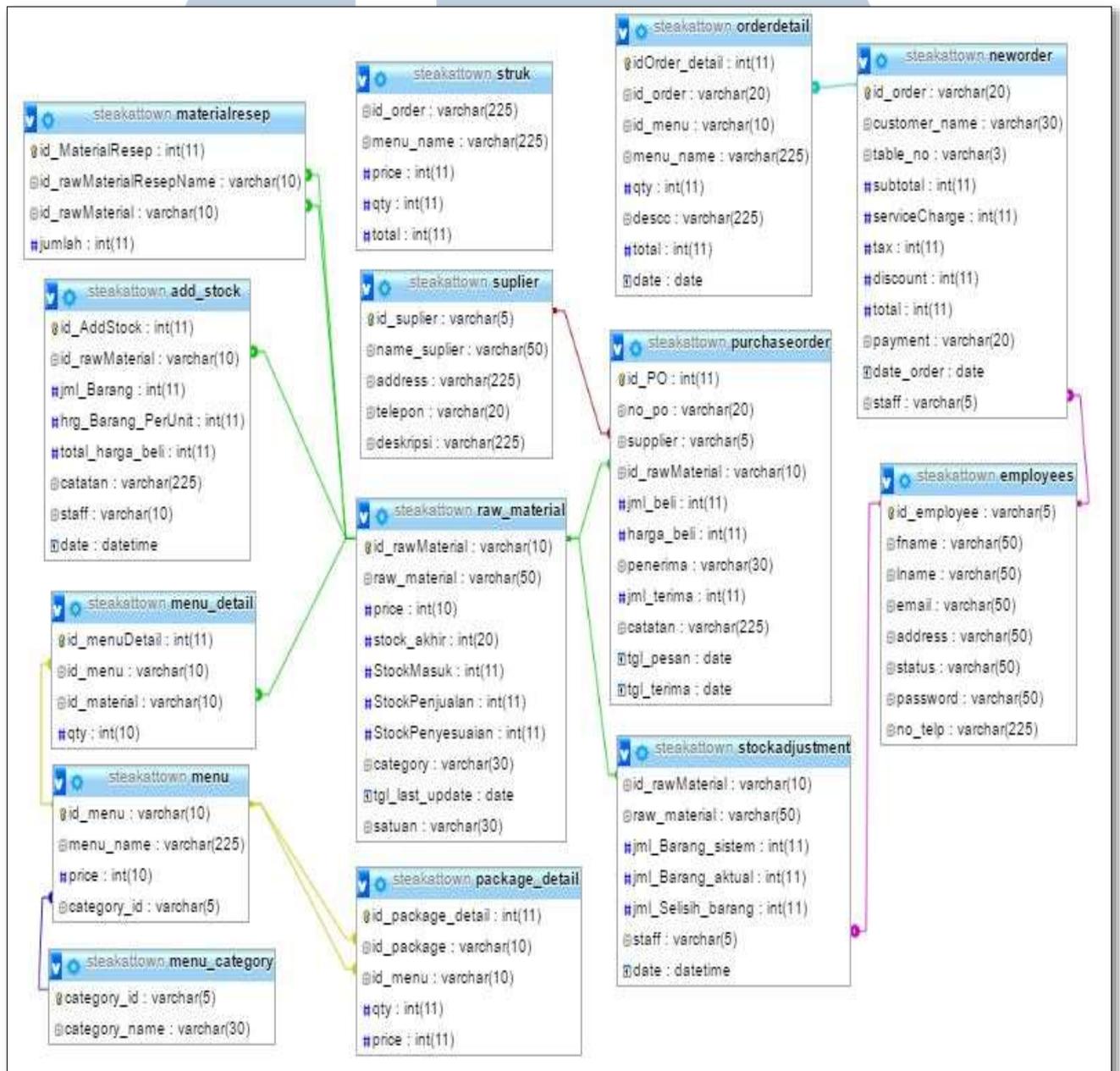


Gambar 3.18 Flowchart analisa data jenis pembayaran

Pada gambar 3.18 menunjukkan alur data dari analisa data transaksi dimana di dalam proses ini akan menampilkan jenis pembayaran yang langsung menggunakan cash ataupun menggunakan kartu, dan lengkap dengan jumlah total nominal dari pembayaran *cash* maupun pembayaran kartu.

C. Entity Relationship Diagram

Terdapat 14 tabel yang digunakan pada sistem *Point Of Sales*. Gambar 3.19 menunjukkan relasi antar tabel yang digunakan pada sistem *Point of sales* Steak@Town.



Gambar 3.20 Entity Relationship Diagram

MULTIMEDIA
NUSANTARA

Berikut adalah penjelasan serta struktur dari setiap tabel yang digunakan dalam pembuatan menu-menu tersebut.

D. Struktur Table

Berikut adalah tabel – tabel yang digunakan dalam sistem yang disimpan di dalam *database* MySQL.

1. Tabel employees

Fungsi : Menyimpan data – data pengguna sistem

Primary Key : id_employee

Foreign Key : -

Tabel 3.1 Tabel employees

No.	NAMA FILED	TIPE DATA	KETERANGAN
1	Id_employee	VARCHAR(5)	ID pekerja
2	Fname	VARCHAR(50)	Nama depan
3	Lname	VARCHAR(50)	Nama belakang
4	Email	VARCHAR(50)	Email pekerja
5	address	VARCHAR(50)	Alamat pekerja
6	Status	VARCHAR(50)	Posisi pekerja
7	Password	VARCHAR(50)	<i>Password login</i>
8	No_telp	VARCHAR(50)	Nomor telpon pekerja

Di dalam tabel *employees* terdapat delapan kolom, pertama terdapat kolom id_employee yang bertipe data *varchar* yang digunakan

sebagai nomor pegawai yang bekerja di dalam restoran, kedua adalah *fname* yang merupakan nama depan dari pegawai yang bertipe data *varchar*, ketiga adalah *lname* yang bertipe data *varchar* merupakan nama belakang dari pekerja, keempat adalah *address* merupakan alamat dari pekerja, kelima adalah *status* merupakan jabatan yang dimiliki pekerja sebagai akses masuk ke dalam sistem kasir, keenam adalah *email* merupakan alamat *email* yang dimiliki oleh pekerja, ketujuh adalah *password* merupakan kata sandi yang dimiliki oleh pekerja untuk dapat masuk ke dalam aplikasi, dan kedelapan adalah *no_telp* merupakan nomor telepon yang dimiliki oleh pegawai untuk dapat dihubungi.

2. Tabel *raw_material*

Fungsi : Menyimpan data – data bahan baku

Primary Key : *id_rawMaterial*

Foreign Key :-

Tabel 3.2 Tabel *raw_material*

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	<i>Id_rawMaterial</i>	Varchar(10)	ID bahan baku
2	<i>rawMaterial</i>	Varchar(10)	Nama bahan baku
3	<i>Price</i>	Int(10)	Harga bahan baku
4	<i>Stock_akhir</i>	Int(20)	Jumlah Stock akhir bahan baku
5	<i>StockMasuk</i>	Int(11)	Jumlah Stock masuk bahan baku

Di dalam tabel *raw_material* terdapat sepuluh variabel, yaitu *id_rawMaterial* sebagai id material, *rawMaterial* sebagai nama bahan baku, *price* menunjukkan harga, *stock_akhir* menunjukkan *stock* akhir yang terdapat pada sistem, *stockMasuk* menunjukkan jumlah *stock* yang masuk, *stockPenjualan* menunjukkan *stock* penjualan yang terjadi, *stockPenyesuaian* menunjukkan hasil *stock* penyesuaian jika terdapat perubahan jumlah bahan baku yang ada digudang, *category* merupakan jenis dari material yang ada di dalam sistem, dan *tgl_last_update* menunjukkan tanggal berapa transaksi penambahan *stock* itu dilakukan, dan satuan merupakan satuan dari *stock* yang tersedia.

3. Tabel *menu_category*

Fungsi : menyimpan data *category* dari menu

Primary Key : *category_id*

Foreign Key : -

Tabel 3.3 Tabel *menu_category*

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	<i>Category_id</i>	Varchar(5)	ID kategori
2	<i>Category_name</i>	Varchar(5)	Nama kategori

Di dalam tabel *menu_category* terdapat dua variabel yaitu, *category_id* sebagai id dari kategori dan *category_name* sebagai nama kategori.

4. Tabel menu

Fungsi : menyimpan menu – menu yang telah dibuat.

Primary Key : id_menu

Foreign Key : category_id

Tabel 3.4 Tabel menu

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	Id_menu	Varchar(10)	ID menu
2	Menu_name	Varchar(225)	Nama menu
3	Price	Int(10)	Harga menu
4	Category_id	Varchar(5)	Kategori dari menu

Di dalam tabel menu terdapat empat variabel yaitu id_menu sebagai id dari menu yang tersedia, menu_name sebagai nama menu, price menunjukkan harga menu, dan kategori_id sebagai *foreign key* dari tabel kategori.

5. Tabel menu_detail

Fungsi : menyimpan detail data dari menu seperti bahan baku apa saja yang digunakan.

Primary Key : id_menuDetail

Foreign Key : id_menu, id_material

Tabel 3.5 Tabel menu_detail

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	Id_menuDetail	Int(11)	ID menu detail

Tabel 3.5 Tabel menu_detail (lanjutan)

No.	NAMA FIELD	TIPE DATA	KETERANGAN
2	Id_menu	Varchar(10)	ID menu
3	Id_material	Varchar(10)	ID material dari menu yang digunakan
4	Qty	Int(10)	Jumlah porsi bahan baku yang digunakan.

Di dalam tabel menu_detail terdapat empat variabel yaitu id_material sebagai id dari detail menu, id_menu sebagai id dari menu yang akan dijabarkan detailnya, id_material sebagai id material yang akan menjadi bahan baku dari menu.

6. Tabel material_resep

Fungsi : untuk menyimpan bahan baku yang memiliki resep, seperti saus – sausan yang memiliki bahan baku.

Primary Key : id_MaterialResep

Foreign Key : id_rawMaterialResepName, id_rawMaterial.

Tabel 3.6 Tabel material_resep

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	Id_MaterialResep	Int(11)	ID Material resep
2	Id_rawMaterialResepName	Varchar(10)	ID dari nama menu yang memiliki resep.
3	Id_rawMaterial	Varchar(10)	ID dari bahan baku

Tabel 3.6 Tabel material_resep (lanjutan)

No.	NAMA FIELD	TIPE DATA	KETERANGAN
4	Jumlah	Int(11)	Jumlah bahan baku

Di dalam tabel material_resep terdapat empat variabel dimana id_rawMaterialResepName, id_rawMaterial merupakan *foreign key* dari tabel yang sama yaitu tabel raw_material, dimana di dalam tabel ini dapat menggunakan material sebagai bahan baku dari material lainnya.

7. Tabel add_stock

Fungsi : untuk menyimpan transaksi penambahan *stock*

Primary Key : id_addStock

Foreign Key : id_rawMaterial

Tabel 3.7 Tabel add_stock

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	Id_AddStock	Int(11)	ID tambah stock
2	Id_rawMaterial	Varchar(10)	Id bahan baku
3	Jml_Barang	Int(11)	Jumlah bahan baku yang ditambah
4	Hrg_Barang_PerUnit	Int(11)	Harga per unit
5	Total_harga_beli	Int(11)	Total harga beli
6	Catatan	Int(11)	Catatan
7	Staff	Varchar(225)	Nama staff
8	Date	Datetime	Tanggal transaksi

Di dalam tabel add_stock terdapat delapan variabel dimana id_rawMaterial merupakan *foreign key* dari tabel raw_material yang berguna sebagai variabel untuk menambah *stock*.

8. Tabel purchaseorder

Fungsi : Untuk menyimpan transaksi pemesanan barang dan pengambilan barang yang diterima.

Primary Key : id_PO

Foreign Key : supplier, id_rawMaterial

Tabel 3.8 Tabel purchaseorder

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	Id_PO	Int(11)	ID purchase order
2	No_PO	Varchar(20)	Nomor PO
3	Supplier	Varchar(5)	ID Supplier
4	Id_RawMaterial	Varchar(10)	ID bahan baku
5	Jml_beli	Int(11)	Jumlah beli
6	Harga_beli	Int(11)	Harga beli
7	Penerima	Varchar(30)	Orang yang menerima
8	Jml_terima	Int(11)	Jumlah barang yang diterima
9	Catatan	Varchar(225)	Catatan
10	Tgl_pesan	Date	Tanggal pesan
11	Tgl_terima	Date	Tanggal terima

Di dalam tabel purchaseorder terdapat sebelas variabel dimana variabel supplier menjadi *foreign key* ke dalam tabel supplier dan id_rawMaterial menjadi *foreign key* ke dalam tabel raw_material.

9. Tabel stockadjustment

Fungsi : menyimpan transaksi penambahan stock barang yang sesuai dengan keutuhan barang aslinya.

Primary Key : -

Foreign Key : id_rawMaterial

Tabel 3.9 Tabel stockadjustment

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	Id_rawMaterial	Varchar(50)	ID material
2	Raw_material	Varchar(50)	Nama material
3	Jml_Barang_sistem	Int(11)	Jumlah barang di dalam sistem
4	Jml_Barang_aktual	Int(11)	Jumlah barang asliny
5	Jml_Selisih_barang	Int(11)	Selisih jumlah barang
6	Staff	Varchar(5)	Nama staff
7	Date	Datetime	Waktu transaksi

Di dalam tabel stockadjustment terdapat tujuh variabel dimana id_material menjadi *foreign key* ke dalam tabel raw_material.

10. Tabel supplier

Fungsi : untuk menyimpan data supplier.

Primary Key : id_supplier

Foreign Key : -

Tabel 3.10 Tabel supplier

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	Id_supplier	Varchar(5)	ID Supplier
2	Name_supplier	Varchar(50)	Nama supplier
3	Address	Varchar(225)	Alamat supplier
4	Telepon	Varchar(20)	Nomer telpon supplier
5	Deskripsi	Varchar(225)	Deskripsi supplier

11. Tabel neworder

Fungsi : menyimpan transaksi order yang baru.

Primary Key : id_order

Foreign Key : staff

Tabel 3.11 Tabel new order

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	Id_order	Varchar(20)	ID order
2	Customer_name	Varchar(30)	Nama pelanggan
3	Table_no	Varchar(3)	Nomer meja
4	Subtotal	Int(11)	Subtotal
5	serviceCharge	Int(11)	Biaya pelayanan

Tabel 3.11 Tabel new order (lanjutan)

No.	NAMA FIELD	TIPE DATA	KETERANGAN
6	Tax	Int(11)	Pajak
7	Discount	Int(11)	Diskon
8	Total	Int(11)	Total
9	Payment	Varchar(20)	Pembayaran
10	Date_order	Date	Tanggal pemesanan
11	Staff	Varchar(5)	Nama staff

Di dalam tabel neworder terdapat sebelas variabel dimana variabel staff menjadi foreign key ke dalam tabel *employees* untuk mendapatkan data dari pegawai yang bekerja melayani pelanggan.

12. Tabel orderdetail.

Fungsi : untuk menyimpan data transaksi order lebih detail lagi seperti menyimpan data menu yang di order.

Primary Key : idOrder_detail.

Foreign Key : id_order, id_menu.

Tabel 3.12 Tabel orderdetail.

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	IdOrder_detail	Int(11)	ID detail order
2	Id_order	Varchar(20)	ID order
3	Id_menu	Varchar(10)	ID menu
4	Menu_name	Varchar(225)	Nama menu

Tabel 3.12 Tabel orderdetail (lanjutan)

No.	NAMA FIELD	TIPE DATA	KETERANGAN
5	Qty	Int(11)	Jumlah menu
6	Descc	Varchar(225)	Catatan
7	Total	Int(11)	Total
8	Date	Date	Tanggal order

Di dalam tabel newOrderDetail terdapat delapan variabel, dimana variabel id_order dan id_menu menjadi foreign key ke dalam tabel order dan menu.

13. Tabel struk.

Fungsi : untuk menyimpan sementara hasil transaksi customer yang akan membayar.

Primary Key : id_order

Foreign Key : -

Tabel 3.13 Tabel struk.

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	Id_order	Varchar(225)	ID order
2	Menu_name	Varchar(225)	Nama menu
3	Price	Int(11)	Harga menu
4	Qty	Int(11)	Jumlah menu
5	Total	Int(11)	Total keseluruhan

14. Tabel package_detail.

Fungsi : untuk menyimpan detail dari paket menu yang dibuat.

Primary Key : id_package_detail

Foreign Key : id_package, id_menu.

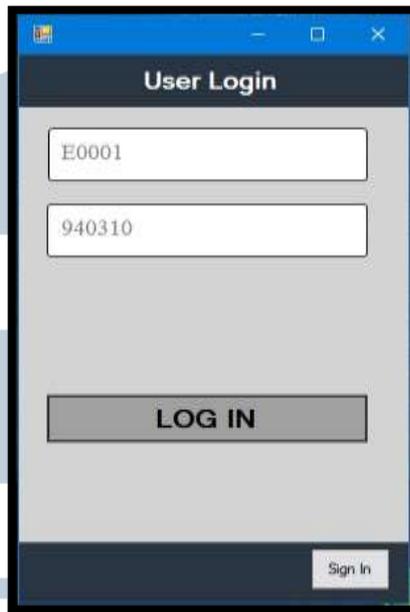
Tabel 3.14 Tabel package_detail.

No.	NAMA FIELD	TIPE DATA	KETERANGAN
1	Id_package_detail	Int(11)	Id detail paket
2	Id_package	Varchar(10)	Id paket
3	Id_menu	Varchar(10)	Id menu
4	Qty	Int(11)	Jumlah menu
5	Price	Int(11)	Harga paket

15. Implementasi

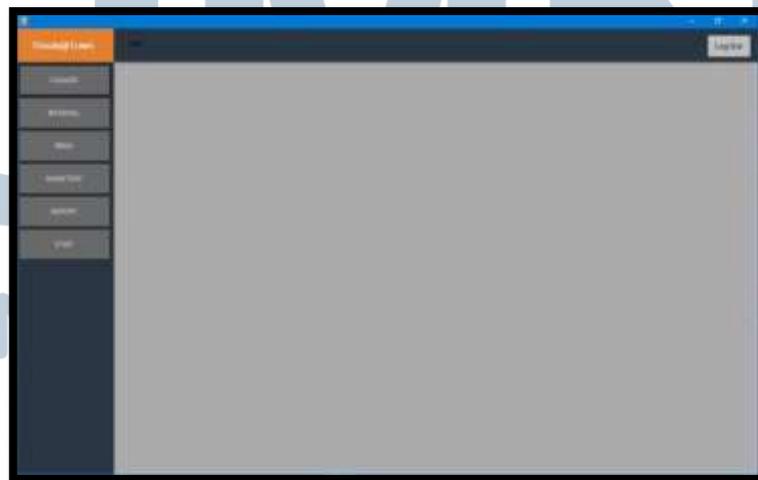
Pada bagian ini menjelaskan implementasi dari rancangan yang telah dijelaskan di bab-bab sebelumnya. Saat pertama kali user membuka aplikasi sistem Steak@Town, akan ditampilkan halaman *login* seperti pada gambar

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



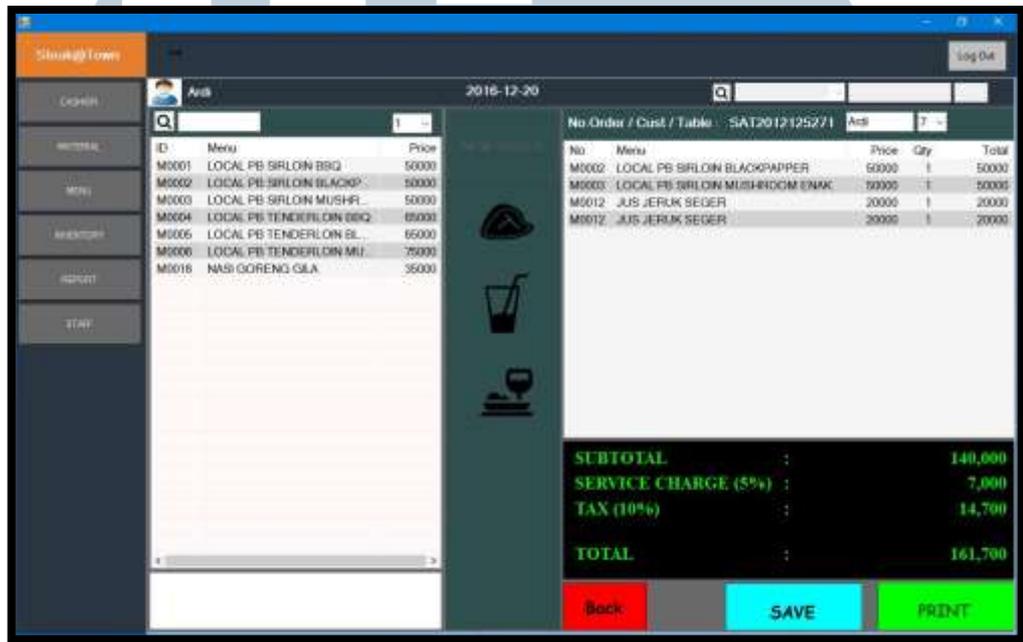
Gambar 3.20 Tampilan Login.

Di dalam *form login* sendiri *user* harus memasukkan *username* dan *password* ke dalam *field* yang tersedia dengan benar. Setelah *user* berhasil login maka *user* akan langsung masuk ke dalam menu utama seperti pada gambar 3.21, di dalam menu utama terdapat enam menu utama di tambah dengan tombol *logout* jika *user* ingin keluar dari aplikasi.



Gambar 3.21 Tampilan Main Menu.

Jika user ingin melakukan transaksi kasir maka *user* cukup menekan tombol menu *cashier*. Setelah itu user akan masuk ke dalam *form cashier* seperti gambar 3.22.



Gambar 3.22 Tampilan Cashier

Jika *user* ingin melakukan transaksi penjualan menu maka *user* cukup menekan tombol *new order* dan langsung dapat memilih menu yang tersedia sesuai dengan *category* menu, *user* cukup mengklik dua kali menu yang akan dipilih. Untuk menyimpan data hasil transaksi user harus memasukkan nama pelanggan dan nomor meja yang ditempati terlebih dahulu dan transaksi selesai. Jika pelanggan ingin menambah menu, user cukup mencari dan memilih nama pelanggan di dalam *combobox* yang terdapat di sebelah pojok kanan atas, setelah itu akan muncul menu yang telah dipesan sebelumnya oleh pelanggan dan user dapat kembali menambah menu yang ingin dipesan oleh pelanggan ataupun jika pelanggan ingin membayar maka dapat langsung menekan menu *print* yang

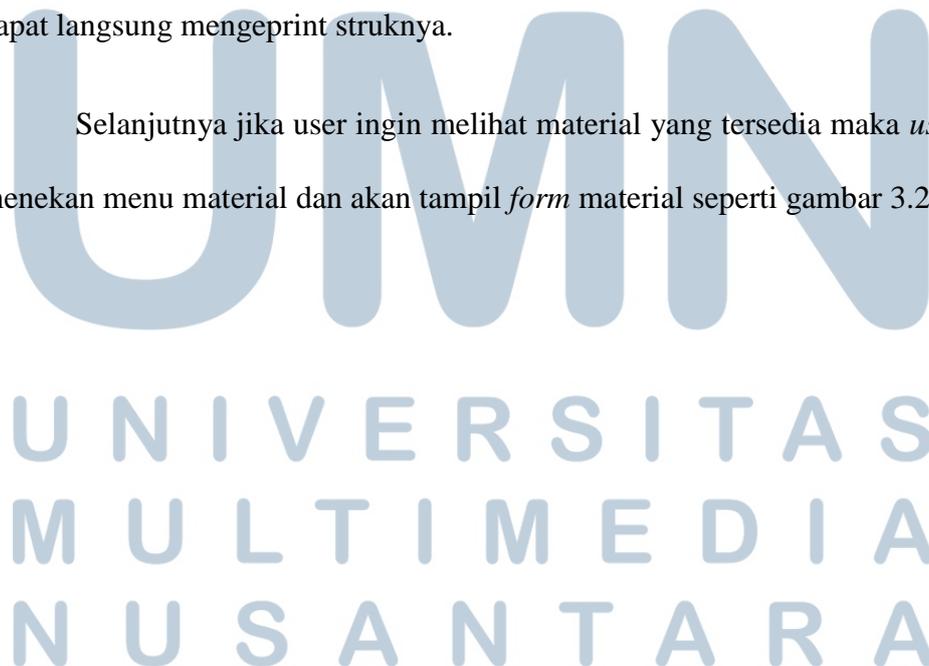
terdapat di sebelah pojok kanan bawah maka akan tampil *form print* seperti gambar 3.23.

ITEM	QTY	PRICE	TOTAL
LOCAL PB SIRLOIN	1	80,000	80,000
LOCAL PB SIRLOIN	1	80,000	80,000
MUSHROOM ENAK	2	20,000	40,000

Gambar 3.23 Tampilan Form Print.

di dalam *form print user* dapat memasukkan diskon apabila ada diskon, memasukkan jenis pembayaran setelah itu *mereshfresh* kembali struk yang ada, dan dapat langsung mengeprint struknya.

Selanjutnya jika user ingin melihat material yang tersedia maka *user* dapat menekan menu material dan akan tampil *form* material seperti gambar 3.24 .





Gambar 3.24 Tampilan Form Material.

Di dalam form material *user* dapat melakukan penambahan material dengan cara menekan tombol “Tambah Produk” yang terdapat pada bagian atas kanan di dalam submenu material, setelah itu *user* dapat mengisi data material di dalam form “Tambah Produk” seperti gambar 3.25.

Tambah Produk

ID Produk: PM010

Nama: Daging kambing

Harga: 50000

Category: Material

Satuan: kg

Tambah Produk

Gambar 3.25 Tampilan Form Add Material.

Di dalam gambar 3.26 merupakan tampilan dari form submenu menu. Di dalam form ini user dapat melihat hasil menu yang telah dibuat yang di kategorikan ke dalam tiga kategori yaitu Foods, Beverages, dan Packages. unruk menambah menu user cukup menekan tombol “Create Menu” dan akan muncul gambar seperti gambar 3.26.

ID MENU	MENU NAME	PRICE
M0001	LOCAL PB BROWN BGG	3000
M0002	LOCAL PB BROWN BLACKPAPPER	3000
M0003	LOCAL PB BROWN MUSHROOM EMAC	3000
M0004	LOCAL PB TENDERLOW BGG	4000
M0005	LOCAL PB TENDERLOW BLACKPAPPER	4000
M0006	LOCAL PB TENDERLOW MUSHROOM	4000
M0007	RAM DORING GUA	3000

Gambar 3.26 Tampilan Form Menu.

Gambar 3.27 merupakan tampilan form “Create Menu” dimana user memasukkan data – data menu berupa nama menu, kategori, harga, dan bahan – bahan yang diperlukan untuk menu tersebut. Jika *user* ingin menambah bahan baku dari menu maka user cukup mengklik dua kali *listview* yang ada di sebelah kiri, dan untuk menambah jumlah bahan yang diperlukan maka user dapat mengklik *combobox* yang ada tengah – tengah *listview*.

Gambar 3.27 Tampilan Form Add Menu

Gambar 3.28 merupakan tampilan form inventory. Terdapat lima submenu di dalam menu inventory seperti stock, stock masuk, supplier, purchase order, dan adjustment.

ID	Name Bahan	Stock Awal	Stock Akhir	Pengeluaran	Penyesuaian	Stock Akhir
RM0001	LOCAL SIRLOIN	50	0	0	0	50
RM0002	LOCAL TENDERLOIN	40	0	0	0	40
RM0003	AUS SIRLOIN BB	50	0	0	0	50
RM0004	AUS T BONE BB	54	0	0	0	54
RM0005	SAUCE BBQ	47	0	0	0	47
RM0006	SAUCE BP	40	0	0	0	40
RM0007	SAUCE MUSHROOM	47	0	0	0	47
RM0008	FRENCH FRIES	46	0	0	0	46
RM0009	VEGETABLES	43	0	0	0	43
RM0010	COKE	50	0	0	0	50
RM0011	SPRITE	40	0	0	0	40
RM0012	FANTA	50	0	0	0	50
RM0013	TEA	50	0	0	0	50
RM0014	MANGGO PEACH	50	0	0	0	50
RM0015	MANGGO JUICE	50	0	0	0	50
RM0016	ORANGE JUICE	49	0	0	0	49
RM0017	LEMON TEA	50	0	0	0	50
RM0018	SUJWA	50	0	0	0	50
RM0019	MELON	50	0	0	0	50
RM0020	STRAWBERRY	50	0	0	0	50
RM0021	JERUK	50	0	0	0	50
RM0022	MILK CHOCOLATE	50	0	0	0	50
RM0023	MILK STRAWBERRY	50	0	0	0	50
RM0024	MILK VANILLA	50	0	0	0	50
RM0025	ICE CREAM CHOCOLATE	50	0	0	0	50
RM0026	ICE CREAM STRAWBERRY	50	0	0	0	50
RM0027	ICE CREAM VANILLA	50	0	0	0	50
RM0028	NASI PUTH	50	0	0	0	50
RM0029	AGUA BOTOL	51	0	0	0	51
RM0030	RICE	50	0	0	0	50
RM0031	BUTTER	50	0	0	0	50
RM0032	GULA	50	0	0	0	50
RM0033	MAYAK SAYUR	50	0	0	0	50
RM0034	SEASONING POWDER	50	0	0	0	50
RM0035	TERUNG	50	0	0	0	50

Gambar 3.28 Tampilan Form Inventory

1. Menu Stock, di dalam menu stock user hanya dapat melihat hasil transaksi dari stock awal, stock masuk, stock hasil penjualan, stock hasil penyesuaian dan stock akhir seperti pada gambar 3.29 dibawah ini.

ID	Name Bahan	Stock Awal	Stock Masuk	Pengeluaran	Penyesuaian	Stock Akhir
FW0001	LOCAL SPUDUM	50	0	0	0	50
FW0002	LOCAL TENDRILUM	40	0	0	0	40
FW0003	AUG STRAIN BE	50	0	0	0	50
FW0004	AUG STRAIN BE	54	0	0	0	54
FW0005	SAUCE BBQ	47	0	0	0	47
FW0006	SAUCE BBQ	49	0	0	0	49
FW0007	SAUCE MUSHROOM	47	0	0	0	47
FW0008	FRENCH FRIES	46	0	0	0	46
FW0009	VEGETABLES	43	0	0	0	43
FW0010	COKE	50	0	0	0	50
FW0011	SPRITE	49	0	0	0	49
FW0012	FANTA	50	0	0	0	50
FW0013	TEH	50	0	0	0	50
FW0014	MANGGO PEACH	50	0	0	0	50
FW0015	MANGGO JUICE	50	0	0	0	50
FW0016	ORANGE JUICE	49	0	0	0	49
FW0017	LEMON TEA	50	0	0	0	50
FW0018	GULA	50	0	0	0	50
FW0019	MELON	50	0	0	0	50
FW0020	STRAWBERRY	50	0	0	0	50
FW0021	JERUK	50	0	0	0	50
FW0022	MILK CHOCOLATE	50	0	0	0	50
FW0023	MILK STRAWBERRY	50	0	0	0	50
FW0024	MILK VANILLA	50	0	0	0	50
FW0025	ICE CREAM CHOCOLATE	50	0	0	0	50
FW0026	ICE CREAM STRAWBERRY	50	0	0	0	50
FW0027	ICE CREAM VANILLA	50	0	0	0	50
FW0028	NASI PUTHI	50	0	0	0	50
FW0029	AGUA BOTOL	81	0	0	0	81
FW0030	RICE	50	0	0	0	50
FW0031	BUTTER	50	0	0	0	50
FW0032	GULA	50	0	0	0	50
FW0033	MAYAK CAKUR	50	0	0	0	50
FW0034	SEASONS POWDER	50	0	0	0	50
FW0035	TER-BAK	50	0	0	0	50

Gambar 3.29 Tampilan Form Stock.

2. Menu stock masuk, di dalam menu stock masuk user dapat melihat transaksi penambahan stock dan juga dapat menambah transaksi penambahan stock dengan cara mengklik tombol “Add Stock” yang ada di dalam tampilan menu “Stock Masuk”. Di dalam form “Add Stock” user dapat memasukkan data berupa nama bahan, tanggal transaksi, jumlah, harga beli perunit, total harga beli secara keseluruhan dan catatan. Untuk menambah bahan ke dalam listview user dapat mengklik “Add” dan untuk menyimpannya user dapat mengklik data tombol “Submit”, seperti gambar 3.30 dibawah ini.

ID	Material	Jumlah	Harga Beli Per Unit	Total Harga Beli
RM0001	LOCAL SIPLOIN	1	300000	300000

Gambar 3.30 Tampilan Form Add Stock

Menu *supplier*, di dalam menu *supplier user* dapat melihat *list* nama *supplier* yang sudah tersimpan.

ID	Name	Address	Phone	Description
0.001	GAJAH KUNING	Depok	081234567	Suplier
0.002	REPOK KUNING KUNING	Tangerang Selatan (DKI Jakarta)	081234567	Suplier
0.003	REPOK KUNING	Tangerang	081234567	Suplier

Gambar 3.31 Tampilan Form Supplier.

User dapat menambah data *supplier* dengan cara mengklik tombol “Tambah *Supplier*”. Setelah itu *user* dapat menambahkan data – data *supplier* seperti nama, alamat, nomer telpon dan deskripsi dari *supplier* seperti gambar berikut.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

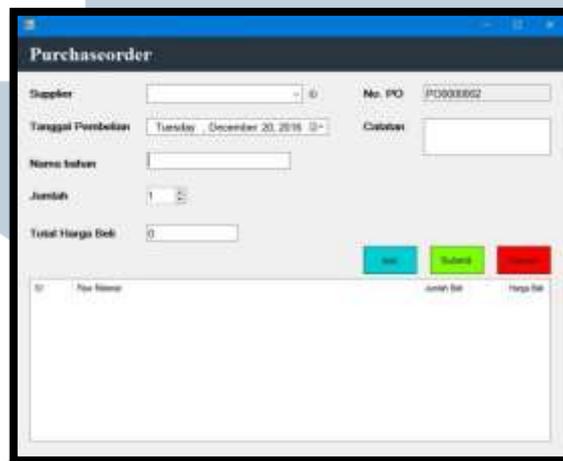
Gambar 3.32 Tampilan Form Add Supplier.

3. Menu *purchase order*, di dalam menu “*Purchase Order*” *user* dapat melihat data transaksi dari *purchase order* yang sudah dilakukan sebelumnya. Di dalam submenu ini juga *user* dapat melakukan penambahan transaksi *purchase order* baru dengan cara mengklik tombol “*Tambah Purchase Order*”.

Gambar 3.33 Tampilan Form Purchase Order.

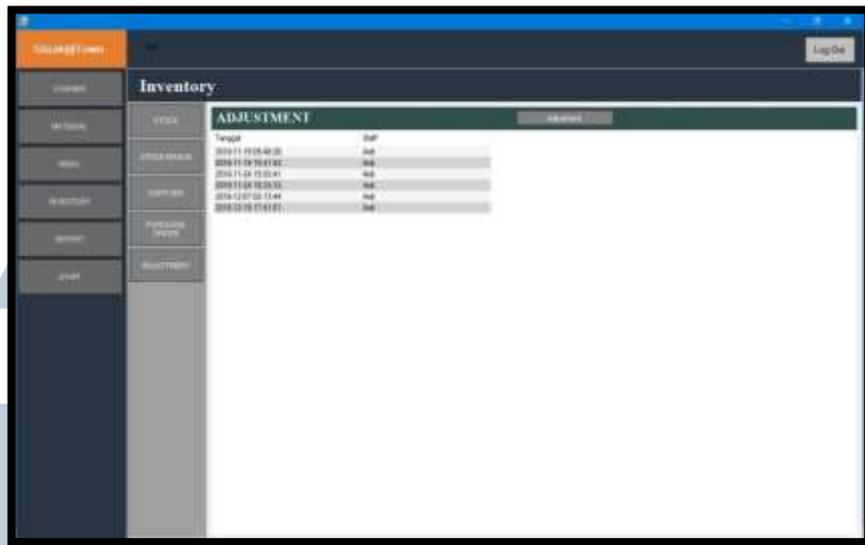
M U L T I M E D I A
N U S A N T A R A

setelah itu *user* dapat memasukkan data berupa nama *supplier* yang telah terdaftar, tanggal pembelian, nama bahan, jumlah bahan yang dipesan, total harga yang dibeli, dan catatan dari transaksi *purchase order* yang dilakukan. Setelah itu untuk menambah ke dalam *listview user* cukup mengklik tombol “Add” dan untuk menyimpannya *user* cukup menekan tombol “Submit”, gambar untuk *form purchase order* dapat dilihat seperti gambar di bawah ini.



Gambar 3.34 Tampilan Form Add Purchase Order.

4. Menu *adjustment* berfungsi untuk menyesuaikan jumlah bahan baku yang tersedia di dalam gudang dengan barang yang tercatat di dalam sistem. Di dalam sistem ini *user* dapat melihat transaksi *adjustment* yang telah dilakukan, dan *user* dapat menambah proses *adjustment* dengan cara mengklik tombol *Adjustment*



Gambar 3.35 Tampilan Form Adjustment.

setelah itu user dapat mengisi data – data *adjustment* seperti nama barang, jumlah barang di sistem, dan jumlah barang yang terdapat pada gudang, dan otomatis sistem akan menghitung selisih barang yang terjadi antara di sistem dan gudang seperti gambar di bawah ini.

Gambar 3.36 Tampilan Form Add Adjustment

Pada gambar 3.37 merupakan tampilan *form report* yang menunjukkan laporan data yang akan digunakan untuk menganalisa penjualan yang dilakukan. Di dalam *form* ini terdapat 3 submenu yaitu submenu analisa penjualan yang digunakan untuk menampilkan data

penjualan perhari, submenu kedua adalah data transaksi yang digunakan untuk menampilkan data transaksi lebih detail lagi. Sedangkan yang terakhir adalah submenu laporan penjualan pembayaran yang digunakan untuk menampilkan pembayaran yang dilakukan apakah menggunakan kartu atau secara tunai.

Tanggal	Jumlah Penjualan	Total Nominal Penjualan	Rata-Rata Nominal Penjualan
21 Desember 2016	1	0	0.0000
19 Desember 2016	1	0	0.0000
13 Desember 2016	1	0	0.0000
18 Desember 2016	1	0	0.0000
19 Desember 2016	1	0	0.0000
20 Desember 2016	3	0	0.0000
TOTAL	8	0	0

Gambar 3.37 Tampilan Form Report.

3.3.2 Kendala yang Ditemukan

Beberapa kendala yang ditemukan saat melakukan praktek kerja magang di Steak@Town antara lain:

1. *User requirement* yang berubah-ubah saat pengembangan sedang berlangsung, sehingga membuat timeline pengerjaan menjadi tertunda.

2. *Misscommunication* dalam pembagian tugas sehingga terkadang tugas yang sama dikerjakan oleh dua orang atau lebih.

3.3.3 Solusi Atas Kendala yang Ditemukan

Adapun solusi atas kendala yang ditemukan saat praktik kerja magang adalah sebagai berikut:

1. Berusaha mempelajari platform version-control secara mandiri melalui artikel-artikel di internet, dan bertanya kepada pembimbing lapangan apabila masih belum mengerti.
2. Mendiskusikan dan menganalisa perubahan user *requirement* apakah memungkinkan untuk dilakukan perubahan, karena apabila perubahannya bersifat major maka akan membuat pengembangan yang sudah dilakukan menjadi sia-sia dan menambah panjang masa pengembangan.
3. Menulis task *list* di papan tugas yang terdapat di kantor, sehingga tugas dapat di kerjakan sesuai dengan jadwal pengerjaan.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA