

## **BAB III**

### **ANALISIS DAN DESAIN SISTEM**

#### **III.1 Analisa Sistem yang Berjalan**

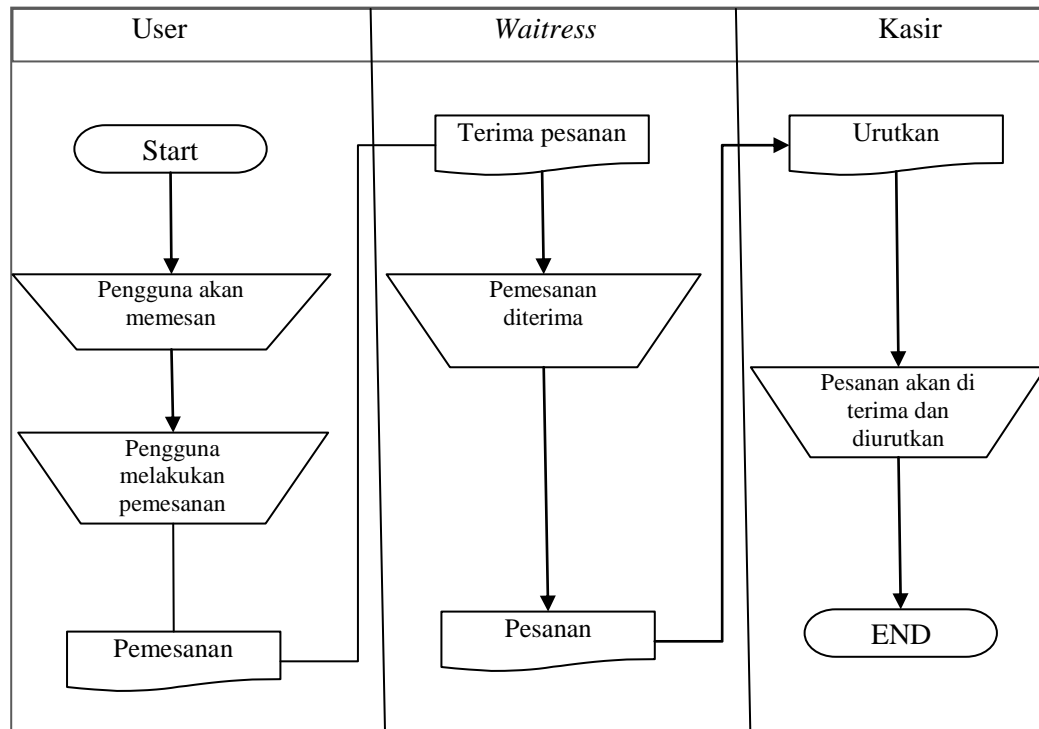
Sistem yang sedang berjalan belum tersedia metode *multilevel feedback queue* untuk penjadwalan data pemesanan makanan. Penerapan algoritma *multilevel feedback queue* dalam pembuatan aplikasi pemesanan makanan pada restoran. Pada sistem yang berjalan, pemesanan makanan belum menggunakan metode *multilevel feedback queue* dan masih dijalankan secara manual.

Sistem yang ada sebelumnya, biasanya pemesanan makanan diurutkan secara manual dan berdasarkan data manual dan tidak ada keteraturan dalam pemesanan makanan.

##### **III.1.1 Analisa Input**

Pada sistem yang berjalan, belum ada sistem pemesanan makanan sebagai tempat pemesanan makanan dengan menggunakan metode *multilevel feedback queue*, sehingga orang yang melakukan pemesanan makanan masih dilakukan secara tidak teratur dan belum menggunakan metode *multilevel feedback queue*. Sehingga pemesanan masih menggunakan data manual serta tidak ada keteraturan dalam penyusunan pemesanan makanan, dan tata cara yang tidak jelas dalam menentukan prioritas pemesanan yang terlebih dahulu.

### III.1.2 Analisa Proses



**Gambar III.1. Flow of Document Pengguna Mencari Solusi**

Berikut merupakan penjelasan dari *flow of document* tersebut diatas adalah sebagai berikut

1. *Start.*
2. Pengguna melakukan pemesanan
3. Pengguna mendatangi atau tempat pemesanan.
4. Pesanan pengguna diterima.
5. Pesanan di proses
6. *End*

### **III.1.3 Analisa Output**

Keluaran atau *output* data dari sistem yang sedang berjalan setelah di *input* dan diproses akan ditampilkan hasilnya dalam bentuk *output*. Adapun sebagai output dari proses yang dilakukan adalah hasil dari proses data pemesanan dalam bentuk diberikan makanan yang dipesan oleh pengguna.

### **III.2 Evaluasi Sistem yang Berjalan**

Setelah penulis melakukan analisa terhadap sistem yang berjalan pada sistem pemesanan makanan tersebut, maka penulis dapat menarik kesimpulan atas sistem yang sedang berjalan yakni mengetahui kelemahan sistem yang ada. Adapun kelemahan sistem yang sedang berjalan setelah mengevaluasi sistem yang berjalan, maka ada beberapa kendala antara lain sebagai berikut:

1. Dalam proses pemesanan makanan yang dilakukan, terlalu banyak membuang waktu dan tenaga untuk biaya pembelian buku.
2. Data pemesanan makanan yang tidak beraturan.
3. Tidak adanya metode penetapan prioritas pemesanan makanan.
4. Merugikan pengguna yang melakukan pemesanan makanan.
5. Dari sistem yang lama dapat dilihat ke efektifan waktunya, seperti melakukan penetapan prioritas proses pemesanan makanan terhadap data pemesanan makanan.
6. Memerlukan waktu yang lama dan memerlukan ketelitian yang tinggi, sedangkan pada rancangan yang di usulkan proses penetapan prioritas pemesanan makanan dapat dilakukan dengan mudah dan cepat.

### III.3 Desain Sistem

Kelemahan sistem yang sedang berjalan perlu dipikirkan dan mencari solusi terbaik. Kelemahan ini dapat diperkecil dengan merancang suatu sistem yang dapat menutupi kelemahan pada sistem yang berjalan tersebut. Dalam hal ini penulis akan mendesain dan memberikan gambaran yang jelas mengenai rancang bangun sistem yang akan diusulkan sebagai alternatif perbaikan pada sistem yang sedang berjalan.

Pada tahap ini perlu membatasi rancang bangun sistem yang diusulkan agar lebih mudah dalam memahami sistem nantinya. Tahap ini terdapat beberapa bagian yakni, disain konfigurasi sistem, desain secara global dan disain sistem secara detail.

#### III.3.1 Penerapan Metode *Multilevel Feedback Queue*

Algoritma Metode *Multilevel Feedback Queue* ini didefinisikan melalui beberapa parameter, antara lain:

1. Jumlah antrian.
2. Algoritma penjadwalan tiap antrian.
3. Kapan menaikkan proses ke antrian yang lebih tinggi.
4. Kapan menurunkan proses ke antrian yang lebih rendah.
5. Antrian mana yang akan dimasuki proses yang membutuhkan.

Dengan pendefinisian seperti tadi membuat algoritma ini sering dipakai, karena algoritma ini mudah dikonfigurasi ulang supaya cocok dengan sistem. Tapi untuk mengetahui mana penjadwal terbaik, kita harus mengetahui nilai parameter tersebut. *Multilevel feedback queue* adalah salah satu algoritma yang berdasar

pada algoritma *multilevel queue*. Perbedaan mendasar yang membedakan *multilevel feedback queue* dengan *multilevel queue* biasa adalah terletak pada adanya kemungkinan suatu proses berpindah dari satu antrian ke antrian lainnya, baik dengan prioritas yang lebih rendah ataupun lebih tinggi, misalnya pada contoh berikut:

1. Semua proses yang baru datang akan diletakkan pada *queue* 0 ( quantum= 8 ms).
2. Jika suatu proses tidak dapat diselesaikan dalam 8 ms, maka proses tersebut akan dihentikan dan dipindahkan ke *queue* 1 ( quantum= 16 ms).
3. *Queue* 1 hanya akan dikerjakan jika tidak ada lagi proses di *queue* 0, dan jika suatu proses di *queue* 1 tidak selesai dalam 16 ms, maka proses tersebut akan dipindahkan ke *queue* 2.
4. *Queue* 2 akan dikerjakan bila *queue* 0 dan 1 kosong, dan akan berjalan dengan algoritma FCFS.

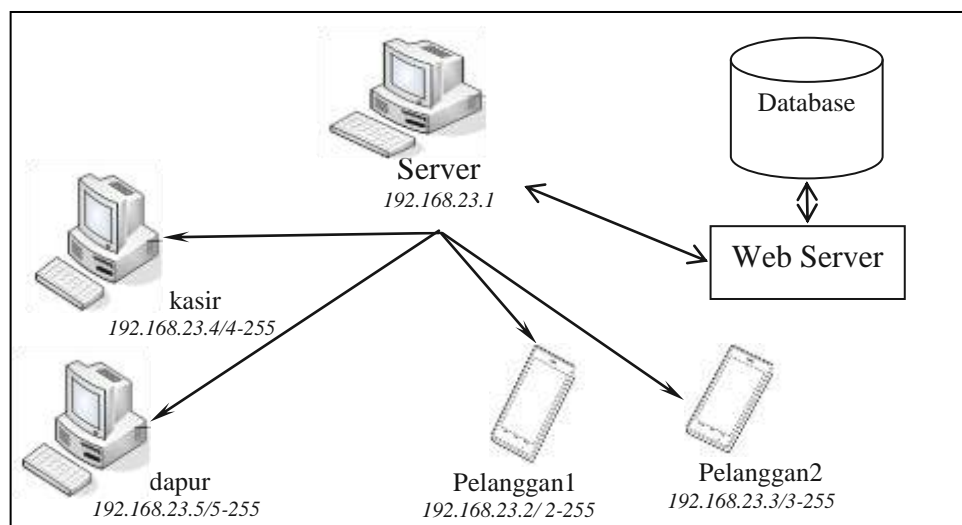
Disini terlihat bahwa ada kemungkinan terjadinya perpindahan proses antar *queue*, dalam hal ini ditentukan oleh *time* quantum, namun dalam prakteknya penerapan algoritma *multilevel feedback queue* akan diterapkan dengan mendefinisikan terlebih dahulu parameter-parameternya, yaitu:

1. Jumlah antrian.
2. Algoritma internal tiap *queue*.
3. Aturan sebuah proses naik ke antrian yang lebih tinggi.
4. Aturan sebuah proses turun ke antrian yang lebih rendah.
5. Antrian yang akan dimasuki tiap proses yang baru datang.

Contoh:

Terdapat tiga antrian; Q1=10 ms, FCFS Q2=40 ms, FCFS Q3=FCFS proses yang masuk, masuk ke antrian Q1. Jika dalam 10 ms tidak selesai, maka proses tersebut dipindahkan ke Q2. Jika dalam 40 ms tidak selesai, maka dipindahkan lagi ke Q3. Berdasarkan hal-hal di atas maka algoritma ini dapat digunakan secara fleksibel dan diterapkan sesuai dengan kebutuhan sistem. Pada zaman sekarang ini algoritma *multilevel feedback queue* adalah salah satu yang paling banyak digunakan.

### III.3.2 Desain Konfigurasi Sistem



**Gambar III.2 Desain Konfigurasi Sistem**

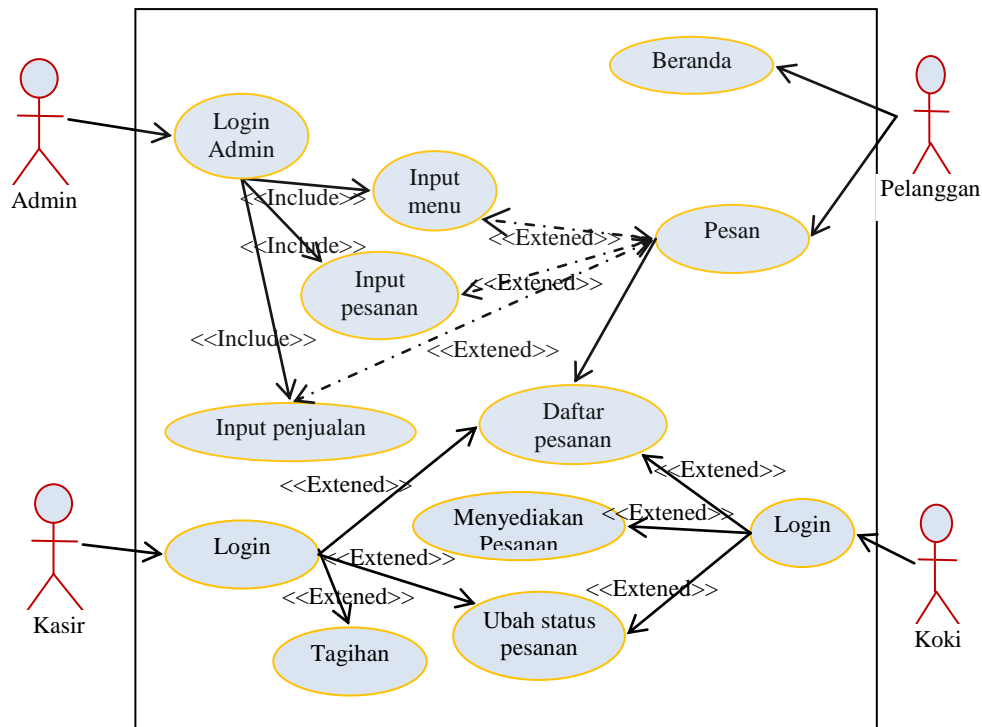
**Table III.1 Konfigurasi Sistem**

User	Konfigurasi IP Adress
Admin	192.168.23.1
Dapur	192.168.23.5/5-255
Kasir	192.168.23.4/4-255
Pelanggan 2	192.168.23.3/3-255
Pelanggan 1	192.168.23.2/2-255

### III.3.3 Desain Sistem Secara Global

Perancangan sistem secara global akan menjelaskan gambaran umum sistem serta model sistem yang akan diusulkan. Karena sistem yang diusulkan akan menghasilkan sebuah perangkat lunak yang berorientasi objek, maka perlu melakukan pemodelan sistem berdasarkan objek-objek yang digunakan. Dalam pemodelan ini penulis menggunakan *Unified Modeling Language* (UML). Pada tahap pemodelan ataupun disain sistem secara global, penulis akan merancang sistem berdasarkan kebutuhan sistem yang akan diusulkan, seperti pembuata *use case diagram*, *sequence diagram* dan *class diagram*. Desain secara detail sistem global sebagaimana telah dijelaskan di atas tidak dapat menggambarkan secara keseluruhan proses yang terjadi dalam sistem, sehingga dibutuhkan disain sistem secara detail yang dapat menjelaskan alur proses yang terjadi di dalam sistem tersebut. Adapun disain sistem secara detail yang diusulkan akan dijelaskan satu persatu berikut ini.

### III.3.3.1 Use case Diagram



**Gambar III.3. Use case Diagram**

*Use case diagram* tersebut digunakan untuk memahami bagaimana interaksi pengguna sistem dengan sistem yang dipakai secara keseluruhan. Pada *use case diagram* ini juga akan menjelaskan kegiatan apa saja yang dapat dilakukan oleh pengguna sistem dan batasan dalam mengakses sistem. *Use case diagram* ini dapat dijelaskan secara detail melalui narasi *use case* sebagai berikut :

#### III.3.3.1.1 Narasi Use Case Home

**Tabel III.2. Narasi Use Case Home**

Nama usecase	Home	
Tipe usecase	Essential	
Priority	High	
Actor	Admin, koki, kasir dan pengunjung (pengguna)	
Deskripsi	Use case ini admin, koki, kasir dan pengunjung memilih menu pada sistem	
Basic flow	Aktor	Sistem

	1. Memilih menu	2. Melakukan Validasi dan memberikan Informasi
	3. Mendapatkan informasi dari sistem	4. Menampilkan Informasi
Post condition	<i>User</i> dapat memilih menu pada sistem	
Extend	-	
Include	-	

### III.3.3.1.2 Narasi *Use case* Login Admin

**Tabel III.3. *Use case* Login Admin**

Nama usecase	<i>Login Admin</i>	
Tipe usecase	Essential	
Priority	High	
Actor	Admin	
Deskripsi	<i>Use case</i> ini admin untuk memasuki dan membuka akses pada Halaman Admin	
Basic flow	Aktor	Sistem
	1. Memasukkan <i>username</i> dan <i>password</i> dan login	2. Memeriksa <i>username</i> dan <i>password</i>
	3. Menerima informasi dari sistem	4. Menampilkan halaman utama admin
Post condition	<i>User</i> dapat memasukan <i>username</i> dan <i>password</i>	
Extend	-	
Include	-	

### III.3.3.1.3 Narasi *Use Case* Login Dapur

**Tabel III.3. *Use case* Login Dapur**

Nama usecase	<i>Login Admin</i>	
Tipe usecase	Essential	
Priority	High	
Actor	Dapur	
Deskripsi	<i>Use case</i> ini admin untuk memasuki dan membuka akses pada Halaman Dapur	
Basic flow	Aktor	Sistem
	1. Memasukkan	2. Memeriksa <i>username</i>

	<i>username</i> dan <i>password</i> dan login	dan <i>password</i>
	3. Menerima informasi dari sistem	4. Menampilkan halaman utama kasir
Post condition	<i>User</i> dapat memasukan <i>username</i> dan <i>password</i>	
Extend	-	
Include	-	

#### III.3.3.1.4 Narasi Use Case Login Kasir

**Tabel III.5. Use case Login Kasir**

Nama usecase	<i>Login Kasir</i>	
Tipe usecase	Essential	
Priority	High	
Actor	Kasir	
Deskripsi	<i>Use case</i> ini untuk memasuki dan membuka akses pada Halaman Kasir	
Basic flow	Aktor	Sistem
	1. Memasukkan <i>username</i> dan <i>password</i> dan login	2. Memeriksa <i>username</i> dan <i>password</i>
	3. Menerima informasi dari sistem	4. Menampilkan halaman utama kasir
Extend	-	
Include	-	

#### III.3.3.1.5 Narasi Use Case input data Menu

**Tabel III.6. Narasi Use Case data Menu**

Nama usecase	<i>Input data Menu</i>	
Tipe usecase	Essential	
Priority	High	
Actor	Admin	
Deskripsi	<i>Use case</i> ini admin untuk mengelola data Menu pada sistem	
Basic flow	Aktor	Sistem
	1. Menambah nama Menu yang akan digunakan sebagai penjualan.	2. Melakukan validasi dan memberikan informasi

	3. Menerima status dari proses yang sedang dilakukan.	4. Menampilkan data hasil proses.
Post condition	User dapat melihat daftar Menu pada halaman daftar Menu	
Extend	-	
Include	-	

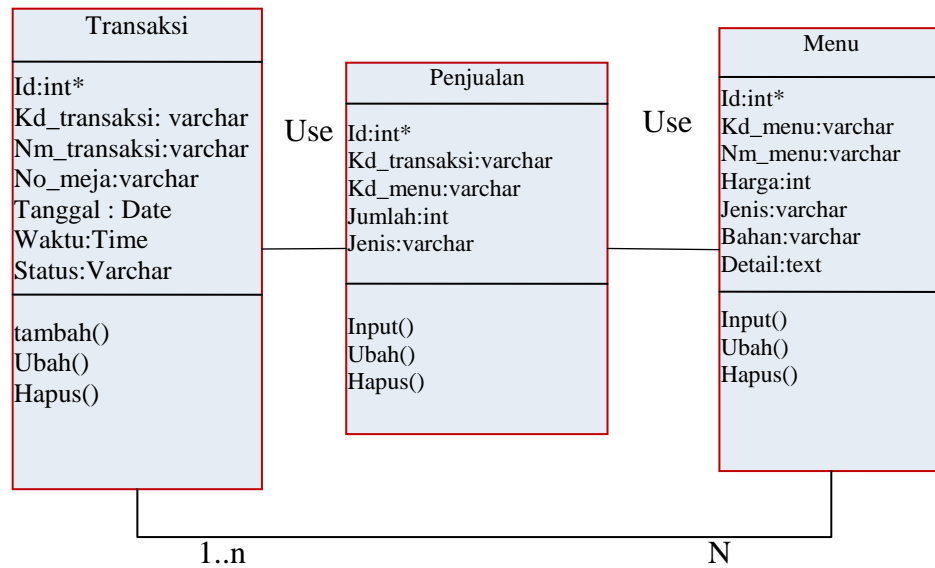
### III.3.3.1.6 Narasi *Use case* Input Penjualan

**Tabel III.7. Narasi *Use Case* Input penjualan**

Nama usecase	<b>Input penjualan</b>	
Tipe usecase	Essential	
Priority	High	
Actor	Admin	
Deskripsi	Use case ini admin untuk mengelola data penjualan pada sistem	
Basic flow	Aktor	Sistem
	1. Menambah nama penjualan yang akan digunakan sebagai pesanan.	2. Melakukan validasi dan memberikan informasi
	3. Menerima status dari proses yang sedang dilakukan.	4. Menampilkan data hasil proses.
Post condition	User dapat melihat daftar penjualan pada menu saat di transaksi	
Extend	-	
Include	-	

### III.3.3.2 *Class Diagram*

*Class diagram* sangat membantu penulis dalam visualisasi struktur kelas kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap tiap kelas didalam model disain dari suatu sistem. Adapun *class diagram* yang diusulkan dapat dilihat pada gambar berikut ini :



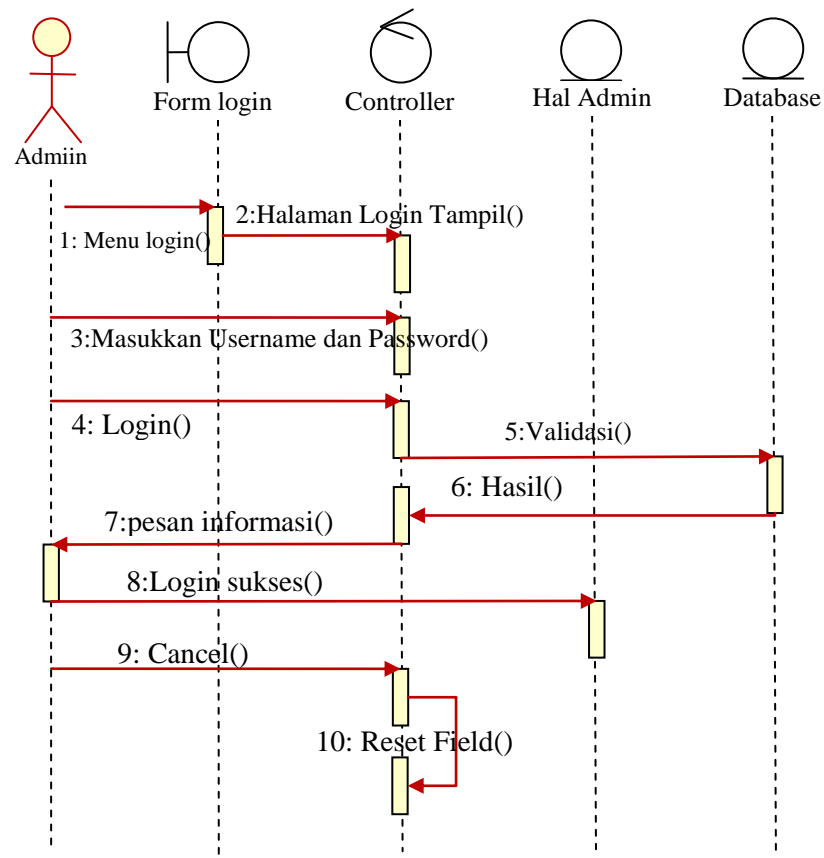
**Gambar III.4. Diagram Class Sistem**

### III.3.3.3 Sequence Diagram

*Sequence Diagram* menggambarkan interaksi antara sejumlah *object* dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* serta interaksi antar *object* yang terjadi pada titik tertentu dalam eksekusi sistem yang diusulkan. Adapun perancangan *sequence diagram*

#### III.3.3.3.1 Sequence Diagram Interaksi Login Admin

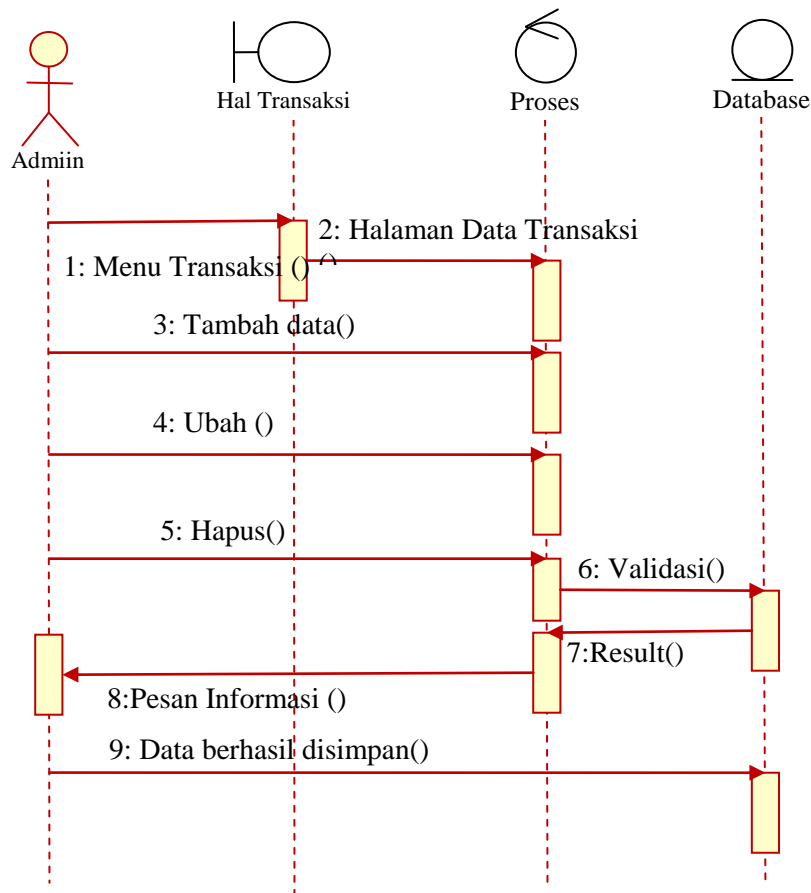
*Sequence Diagram Login* menjelaskan operasi *Login* admin dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. *Obyek-obyek* yang berkaitan dengan proses operasi *Login* Admin diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut, *Sequence Diagram Login* dapat dilihat pada gambar berikut:



**Gambar III.5. Sequence Diagram Login Admin**

### III.3.3.3.2 Sequence Diagram Transaksi

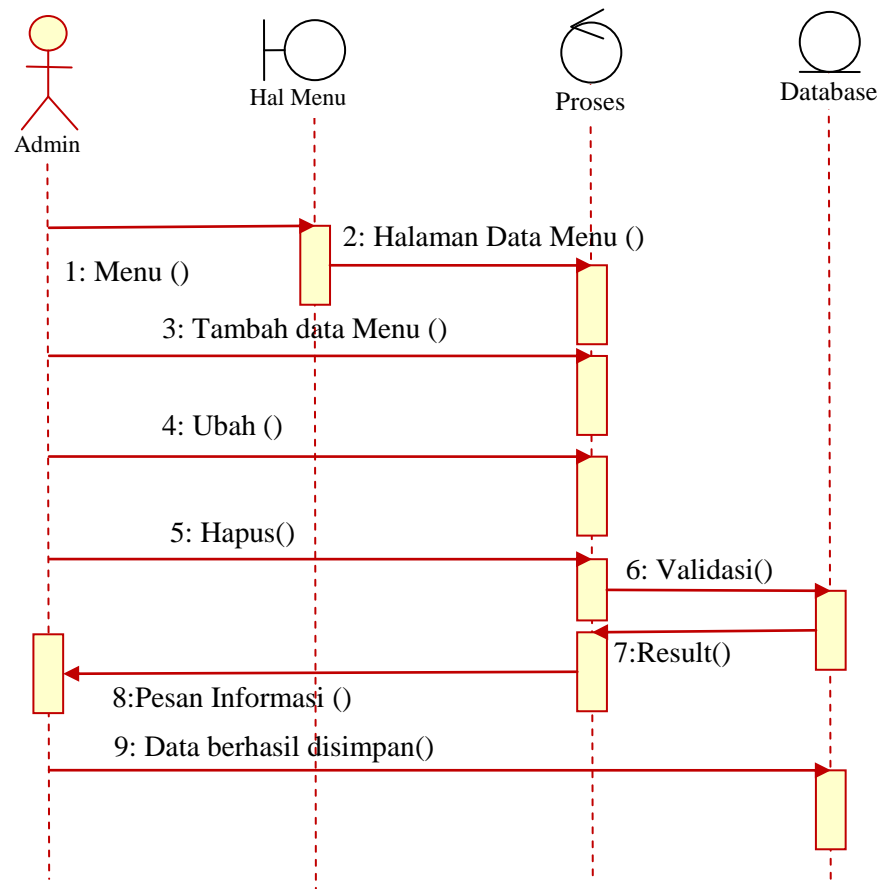
*Sequence Diagram* Transaksi menjelaskan operasi data Transaksi dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. *Obyek-obyek* yang berkaitan dengan proses operasi data Transaksi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut, *Sequence Diagram* data Transaksi dapat dilihat pada gambar berikut:



**Gambar III.6. Sequence Diagram Transaksi**

### III.3.3.3 Sequence Diagram Menu

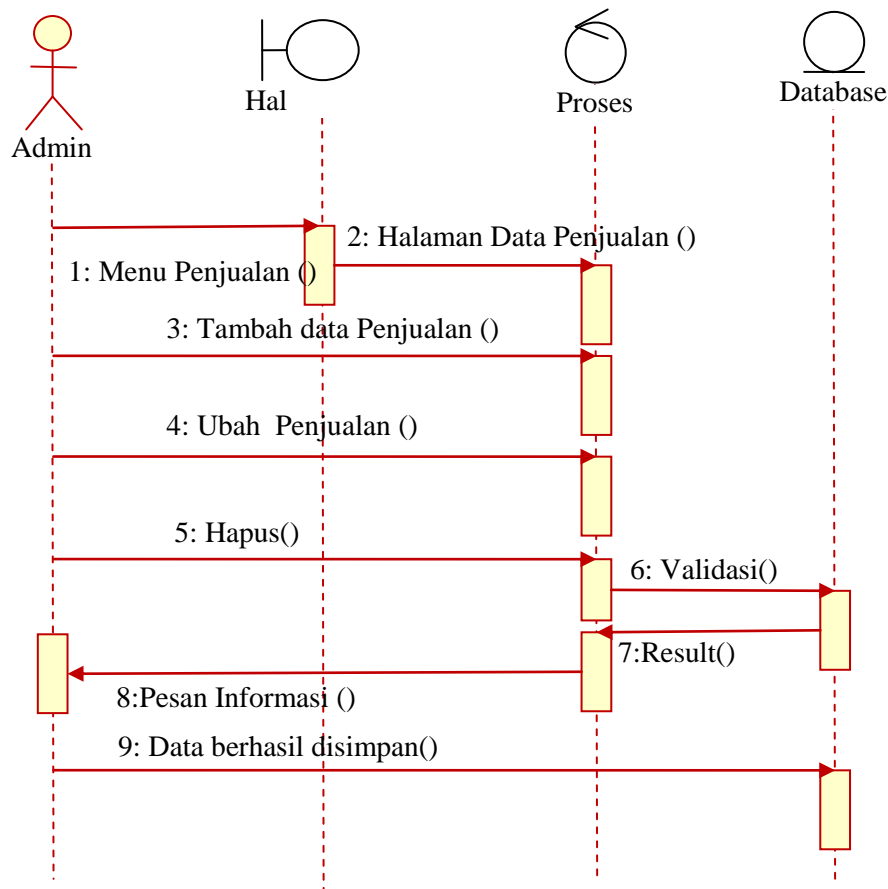
*Sequence Diagram Menu* menjelaskan operasi data Menu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. *Obyek-obyek* yang berkaitan dengan proses operasi data Menu diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut, *Sequence Diagram* data Menu dapat dilihat pada gambar berikut:



**Gambar III.7. Sequence Diagram Menu**

#### III.3.3.3.4 Sequence Diagram Penjualan

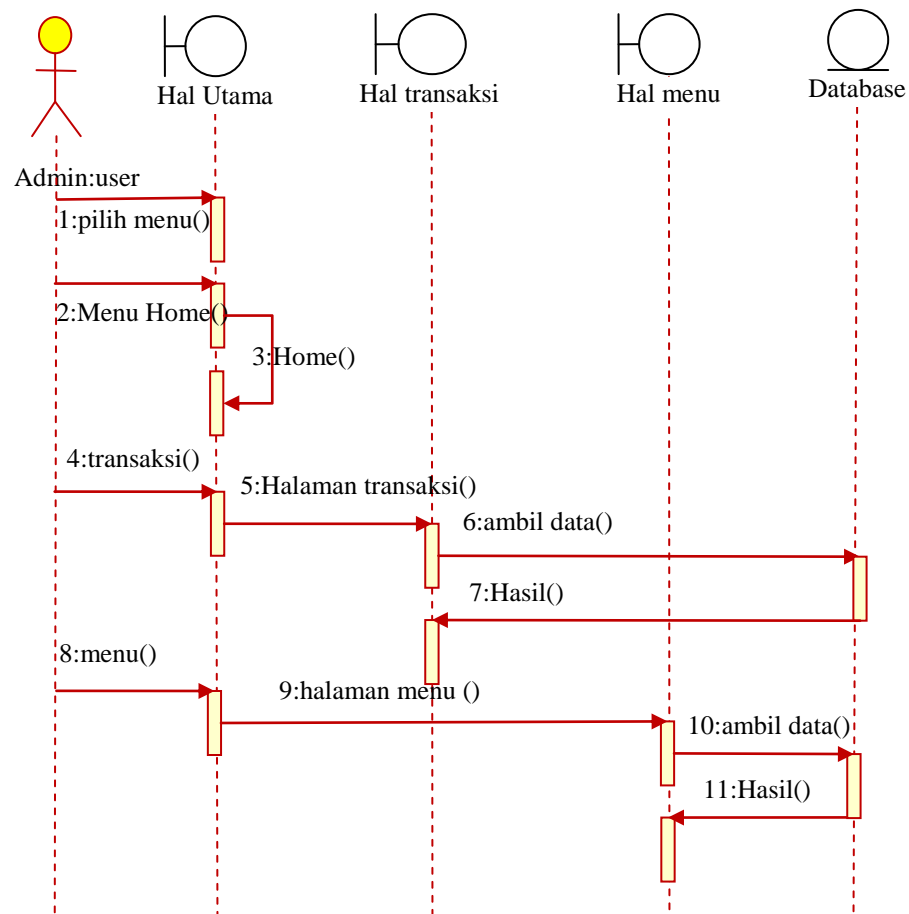
*Sequence Diagram* Penjualan menjelaskan operasi data Penjualan antara transaksi dan menu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Objek-objek yang berkaitan dengan proses operasi data Penjualan antara menu diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut, *Sequence Diagram* data Penjualan menu dapat dilihat pada gambar berikut:



**Gambar III.8. Sequence Diagram Penjualan**

### III.3.3.3.5 Sequence Diagram Halaman Utama

*Sequence Diagram* Halaman Utama menjelaskan operasi Halaman utama dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. *Obyek-obyek* yang berkaitan dengan halaman utama diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut, *Sequence Diagram* halaman utama dapat dilihat pada gambar berikut:



**Gambar III.9. Sequence Diagram Halaman Home**

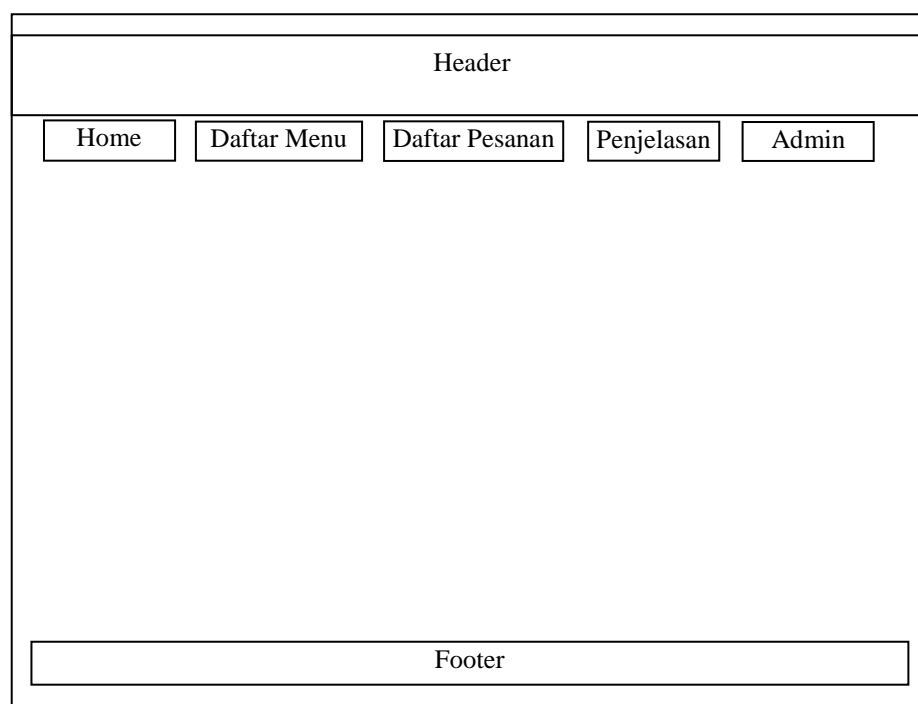
### III.3.4 Desain Sistem Secara Detail

#### III.3.4.1 Desain Output

Desain *output* Sistem global sebagaimana telah dijelaskan di atas tidak dapat menggambarkan secara keseluruhan proses yang terjadi dalam sistem, sehingga dibutuhkan desain sistem secara detail yang dapat menjelaskan alur proses yang terjadi di dalam sistem tersebut. Adapun desain sistem secara detail yang diusulkan akan dijelaskan satu persatu berikut ini:

### III.3.4.1.1 Desain *Output* Halaman Utama

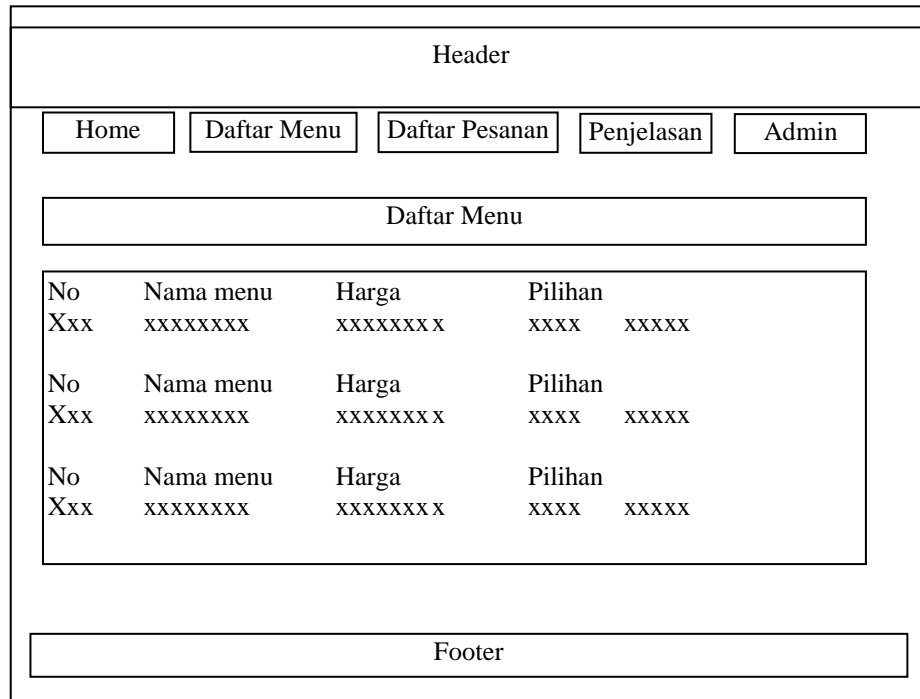
Halaman utama akan menampilkan menu halaman yang akan digunakan oleh *user* dan admin untuk memilih menu selanjutnya. Halaman ini menyajikan pilihan menu yang ada dalam sistem. Rancangan tampilannya seperti terlihat pada gambar di bawah ini:



**Gambar III.10. Desain Halaman Utama**

### III.3.4.1.2 Desain *Output* Halaman Daftar Menu

Halaman Pesanan akan menampilkan menu halaman yang akan digunakan oleh *user* dan untuk memilih menu Pesanan selanjutnya. Halaman ini menyajikan pilihan menu Pesanan yang ada dalam sistem. Rancangan tampilannya seperti terlihat pada gambar berikut ini.



**Gambar III.11. Desain Daftar Menu**

#### III.3.4.1.3 Desain Halaman Pesanan

Halaman Pesanan akan menampilkan menu halaman yang akan digunakan oleh *user* dan admin untuk memilih menu Pemesanan selanjutnya. Halaman ini menyajikan pilihan berapa jumlah yang akan dipesan yang ada dalam sistem. Rancangan tampilannya seperti terlihat pada gambar berikut ini.

Header					
Home	Daftar Menu	Daftar Pesanan	Penjelasan	Admin	
Daftar Menu Pesanan					
No	Nama menu	Harga	Jumlah	Total	Pilihan
Xxx	xxxxxxxx	xxxxxxx	xxxxx	xxxxx	xxxxx
Footer					

**Gambar III.12. Desain Daftar Pesanan**

### III.3.4.2 Desain *Input*

Sistem ini mempunyai beberapa halaman yang akan menjadi *input*. Dalam perancangannya, sistem yang diusulkan mempunyai tiga halaman sebagai keluaran akhir, yaitu.

#### III.3.4.2.1 Halaman *Login Admin*

Untuk sistem keamanan, halaman *login* meminta *username* dan *password* untuk masuk ke halaman admin *login*, rancangan tampilannya seperti terlihat pada gambar berikut ini.

The image shows a wireframe for an admin login page. It is enclosed in a rectangular border. At the top center is a box labeled 'Login Sistem'. Below it, on the left, is the text 'User Id:' followed by a horizontal text input field. Below that, on the left, is the text 'Password:' followed by another horizontal text input field. At the bottom center is a rectangular button labeled 'Login'.

**Gambar III.13. Desain Halaman *Login Admin***

### **III.3.4.3 Desain *Database* (Basis Data)**

Tahap ini merupakan tahap dimana penulis menempatkan data yang sudah ada pada bagian *server*. Data tersebut nantinya akan diproses oleh data yang sudah dibuat. Tempat untuk menampung data tersebut disebut dengan basisdata atau database strukturnya yang terdiri dari atas tabel-tabel yang dibuat dengan menggunakan program *Xampp server*. Adapun struktur tabelnya dapat dilihat sebagai berikut.

#### **III.3.4.3.1 Kamus Data**

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang *input*, *output* dan komponen data *store*. Pembentukan kamus data didasarkan pada alur data yang terdapat pada Diagram Alir data bersifat global (hanya menunjukkan nama alur datanya tanpa menunjukkan struktur dari alur data). Untuk menunjukkan struktur dari alur data secara rinci maka dibentuklah kamus data. Bentuk dari *form* kamus data dapat dilihat pada tabel berikut ini:

Penjualan	:	Id + Kd_transaksi+ kd_menu + jumlah+ jenis+ tanggal+bahan
Menu	:	Id + Kd_menu + nm_menu + bahan+ jenis + harga + detail
Transaksi	:	Id+ Kd_transaksi + nm_transaksi + no_meja + tanggal + waktu + status
Users	:	Username + password + nama_lengkap + email + no_telp + level

#### III.3.4.3.2 Normalisasi Basis Data (*Database*)

Pada ilmu *database* atau basis data, normalisasi digunakan untuk menghindari terjadinya berbagai anomali data dan tidak konsistensinya data. Ini merupakan fungsi *database* secara umum. Dalam beberapa kasus normalisasi ini sangat penting untuk menunjang kinerja *database* dan memastikan bahwa data dalam *database* tersebut aman dan tidak terjadi kesalahan jika mendapat perintah SQL terutama DML yaitu *update*, *insert*, dan *delete*.

Perlu diketahui dalam beberapa kasus Normalisasi *database* terkadang harus diubah menjadi bentuk dinormalisasi, terutama untuk data yang telah besar dan membengkak. Denormalisasi ini ditujukan untuk meningkatkan performance dengan meletakkan beberapa *field* menjadi satu tabel sehingga mudah di tarik. Dinormalisasi ini sering digunakan untuk menarik data yang besar dari *database*.

##### A. Normalisasi *Database* 1NF

Bentuk normal yang pertama atau 1NF mensyaratkan beberapa kondisi dalam sebuah *database*, berikut adalah fungsi dari bentuk normal pertama ini.

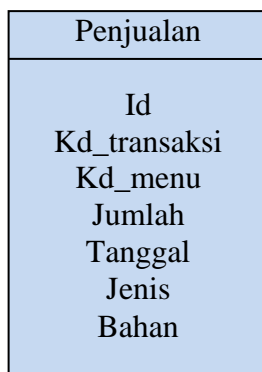
Menghilangkan duplikasi kolom dari tabel yang sama. Buat tabel terpisah untuk masing-masing kelompok data terkait dan mengidentifikasi setiap baris dengan kolom yang unik (*primary key*).



**Gambar III.14 Normalisasi Database 1NF**

### B. *Second normal form (2NF)*

Syarat untuk menerapkan normalisasi bentuk kedua ini adalah data telah dibentuk dalam 1NF, berikut adalah beberapa fungsi normalisasi 2NF. Menghapus beberapa subjek data yang ada pada tabel dan menempatkan mereka pada tabel terpisah. Menciptakan hubungan antara tabel baru dan tabel lama dengan menciptakan *foreign key*. Tidak ada atribut dalam tabel yang secara fungsional bergantung pada *candidate key* tabel tersebut.



**Gambar III.15. Normalisasi Database Bentuk 2NF**

### III.3.4.3.3 Desain Tabel

Tabel merupakan komponen utama pendukung *database*. Tabel juga merupakan pertemuan antara baris dan kolom yang memuat suatu data atribut. Tabel merupakan sumber data bagi setiap aplikasi *database* seperti aplikasi yang akan dirancang dalam penelitian ini. Adapun tabel-tabel data yang dirancang untuk digunakan dalam sistem yang diusulkan adalah sebagai berikut.

#### 1. Tabel Penjualan

Tabel Penjualan digunakan untuk menampung data Penjualan keseluruhan.

Berikut ditampilkan rancangan struktur data Penjualan.

Nama Database : pesananmakanan

Nama Tabel : Penjualan

Field Key : Id

**Tabel III.8. Tabel Penjualan**

No	Nama Field	Data Type	Width	Keterangan
1	Id	Int	1	Id penjualan
2	Kd_transaksi	Char	4	Kode transaksi
3	Kd_menu	Varchar	4	Kode menu
4	Jumlah	Varchar	10	Jumlah
5	Tanggal	Date		Tanggal
6	Jenis	Varchar	10	Jenis
7	Bahan	Varchar	20	Bahan

## 2. Tabel menu

Tabel menu digunakan untuk menampung data menu keseluruhan. Berikut ditampilkan rancangan struktur data menu.

Nama Database : pesananmakanan

Nama Tabel : menu

Field Key : kd\_ menu

**Tabel III.9. Tabel Menu makanan**

No	Nama Field	Data Type	Width	Keterangan
1	Id	Int	11	Id
2	Kd_menu	Varchar	4	nama_ menu
3	Nm_menu	Varchar	200	Nama menu
4	Detail	Text		Detail menu
5	Harga	Varchar	10	Harga menu
6	Bahan	Varchar	20	Bahan menu
7	Jenis	Varchar	20	Jenis menu

## 3. Tabel Transaksi

Tabel Transaksi digunakan untuk menampung hasil sementara Transaksi keseluruhan. Berikut ditampilkan rancangan struktur data Transaksi.

Nama Database : Pesananmakanan

Nama Tabel : Transaksi

Field Key :

**Tabel III.10. Tabel Transaksi**

No	Nama Field	Data Type	Width	Keterangan
1	Id	Varchar	1	Ip transaksi
2	kd_Transaksi	Varchar	4	Kd_transaksi
3	Nm_Transaksi	Varchar	4	Nama
4	Nilai	Varchar	4	Nilai
5	No_meja	Varchar	4	No meja
6	Tanggal	Date		Tanggal
7	Waktu	Time		Waktu
8	Status	Varchar	10	Status

#### 4. Tabel User

Tabel *User* digunakan untuk menampung hasil analisa keseluruhan. Berikut ditampilkan rancangan struktur data hasil analisa.

Nama Database : Pesananmakanan

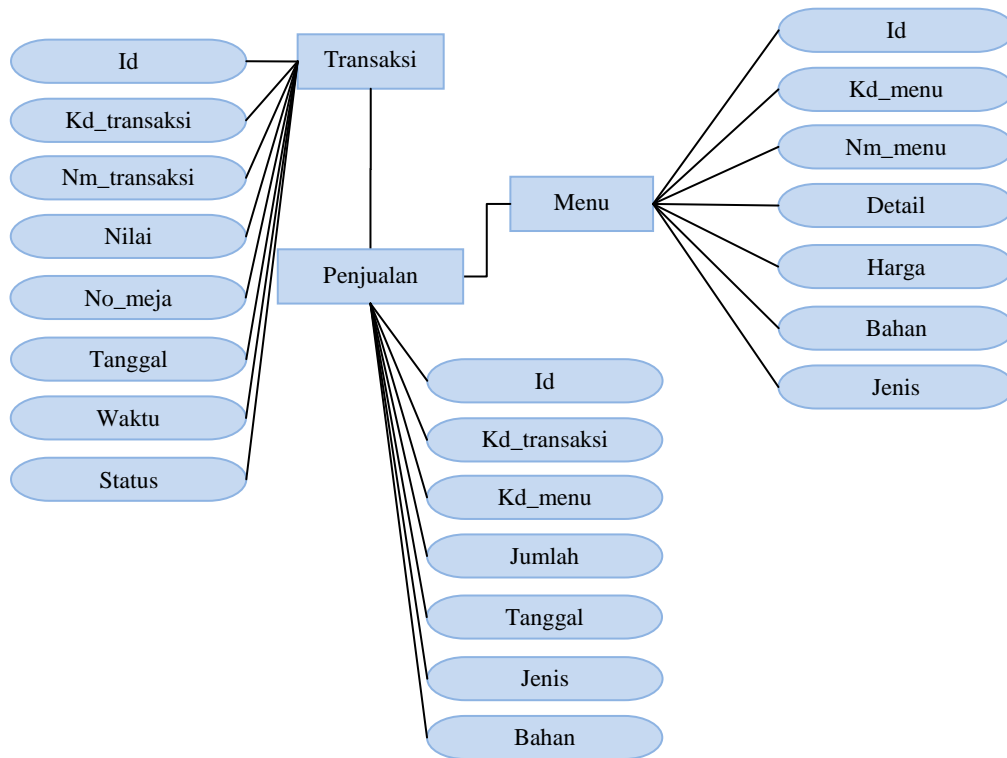
Nama Tabel : Users

Field Key :

**Tabel III.11. Tabel Users**

No	Nama Field	Data Type	Width	Keterangan
1	Username	Text	20	username
2	Password	Varchar	20	Password
3	Nama_lengkap	Text	50	Nama
4	Email	Text	50	Email User
5	No_telp	Text	20	Nomor telp
6	Level	Text	150	Level User
7	Blokir	Varchar	10	Blokir user
8	Id_session	Text	150	Session

### III.3.4.3.4 ERD ( Entity Relation Diagram)



Gambar III.16. *Entity Relation Diagram (ERD)*

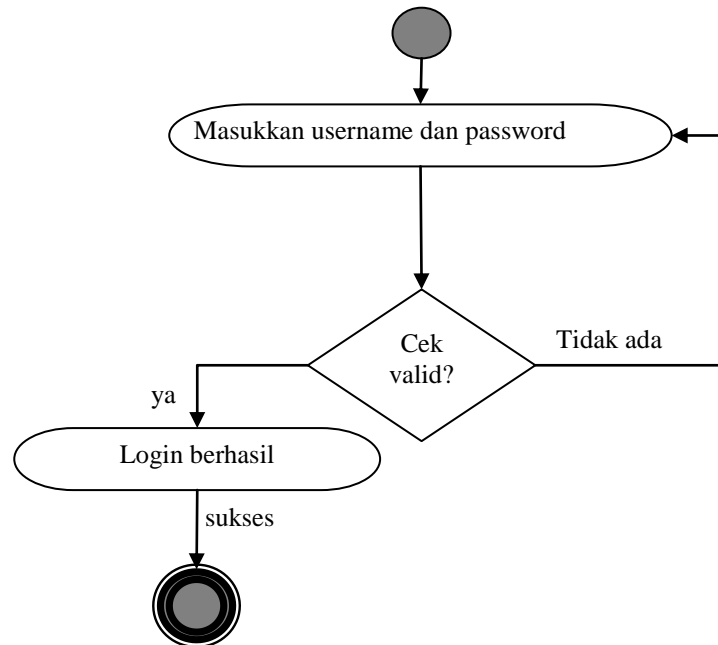
### III.3.4.4 Logika program

Logika program dari sistem yang diusulkan akan digambarkan dalam sebuah *activity diagram*. *Activity diagram* ini akan menjelaskan setiap kegiatan yang akan dilakukan pengguna pada sistem nantinya. Dengan menggambarkan setiap aktivitas dari sistem diharapkan sistem yang akan dibangun lebih mudah dipahami. Adapun *activity diagram* pada sistem yang diusulkan adalah sebagai berikut.

#### III.3.4.4.1 *Activity diagram Login Admin*

*Activity diagram Login Admin* ini akan menggambarkan kegiatan admin saat akan memasuki halaman administrator untuk mengolah master data. Adapun

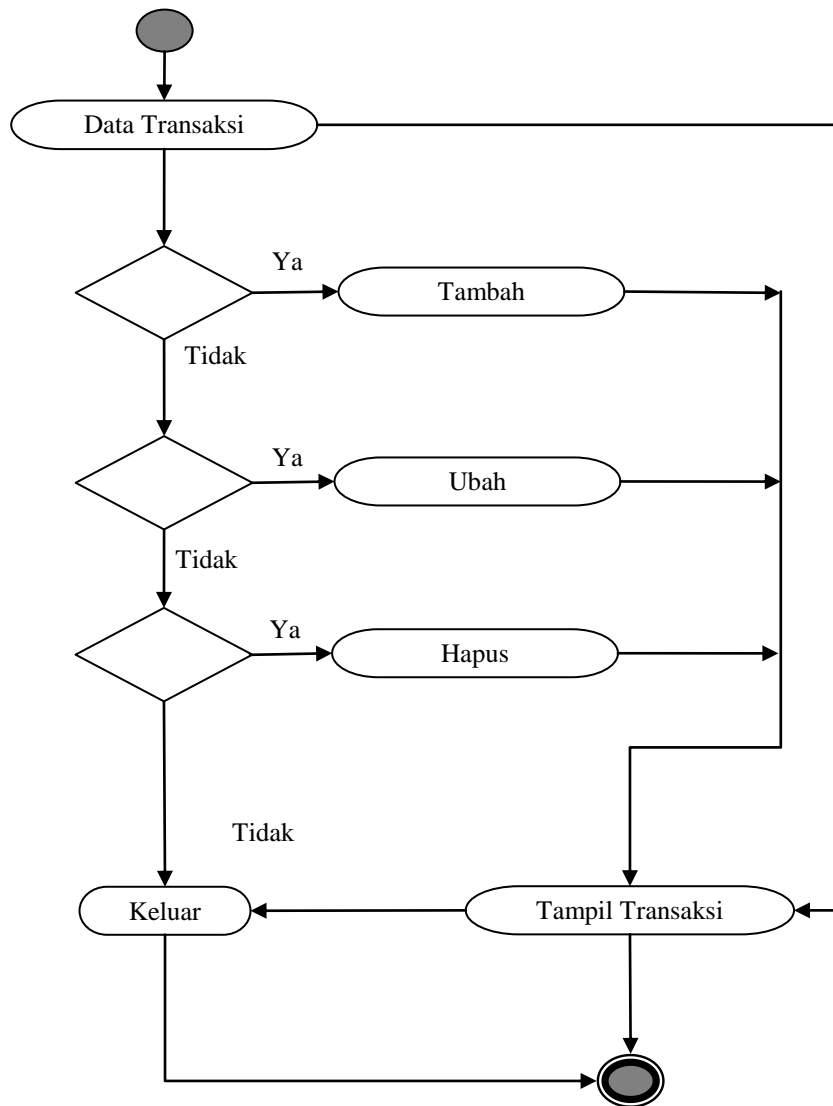
*activity diagram* Login Admin pada sistem yang diusulkan digambarkan pada diagram berikut.



**Gambar III.17. Activity Diagram Login Admin**

#### **III.3.4.4.2 Activity Diagram Data Transaksi**

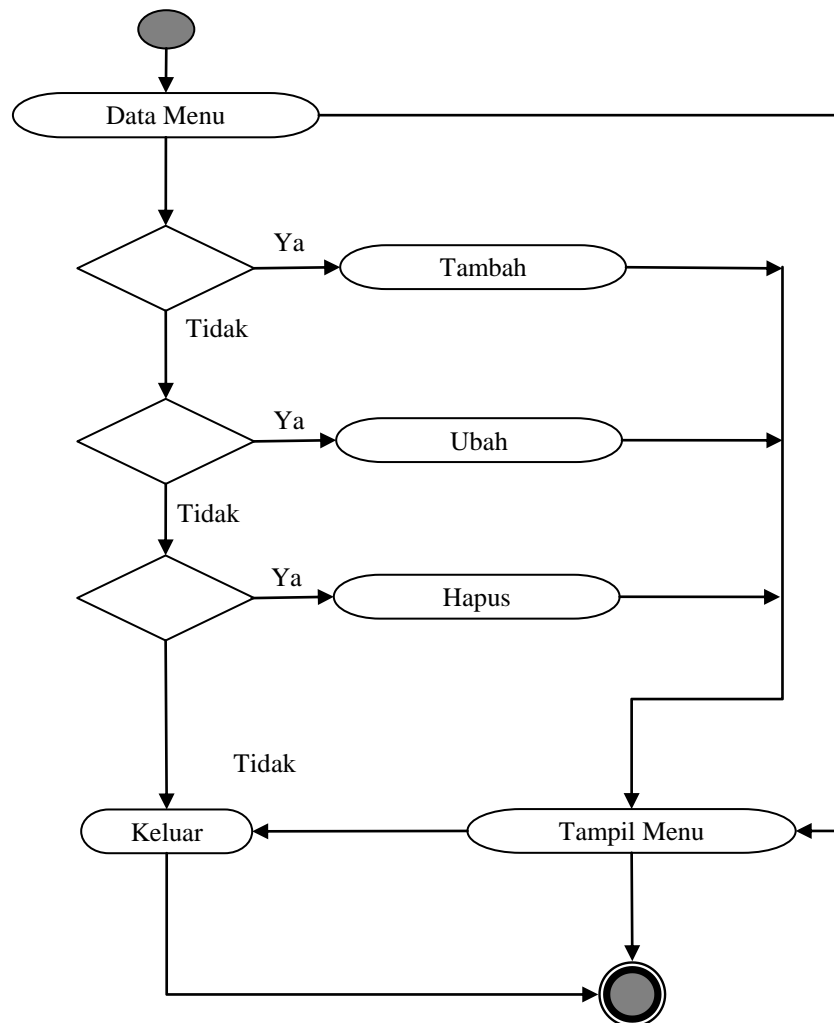
*Activity diagram* data Transaksi ini akan menggambarkan kegiatan admin pada halaman Transaksi untuk mengolah master data Transaksi, untuk ditambah, ubah, hapus menu yang ada pada sistem. Adapun *activity diagram* data Transaksi pada sistem yang diusulkan digambarkan pada diagram berikut.



**Gambar III.18. Activity Diagram Data Transaksi**

#### III.3.4.4.3 Activity Diagram Data Menu

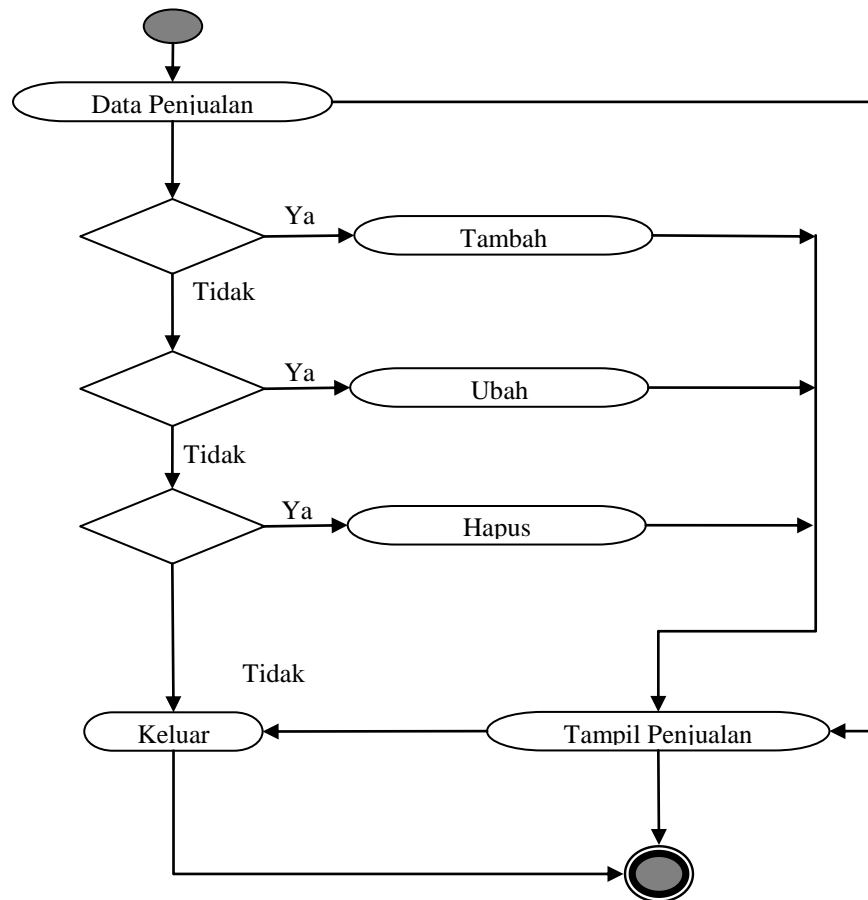
*Activity diagram* data Menu ini akan menggambarkan kegiatan admin pada halaman administrator untuk mengolah master data Menu, untuk ditambah, ubah, hapus Menu yang ada pada sistem. Adapun *activity diagram* data Menu pada sistem yang diusulkan digambarkan pada diagram berikut.



**Gambar III.19. Activity Diagram Data Menu**

#### III.3.4.4 Activity Diagram Penjualan

*Activity diagram* data Penjualan ini akan menggambarkan kegiatan admin pada halaman administrator untuk mengolah master data Penjualan menu yang diinputkan pelanggan, ubah, hapus Penjualan dari menu yang ada pada sistem. Adapun *activity diagram* data Penjualan menu pada sistem yang diusulkan digambarkan pada diagram berikut.



**Gambar III.20. Activity Diagram Data Penjualan**

### III.4 Alat Pendukung

Alat-alat yang mendukung dalam pembuatan sistem merupakan kebutuhan yang harus dipenuhi oleh penulis. Alat-alat tersebut berupa perangkat keras dan perangkat lunak.

Alat dan bahan yang digunakan dalam pengembangan *web apps* pemesanan menu restoran ini adalah:

#### a. Perangkat Keras:

1. Laptop Acer Procesor Intel Core i3 M370 2,40 GHz
2. RAM 2 GB
3. HDD 320 GB

b. Perangkat Lunak:

1. Windows 7 Home Premium 32 bit
2. Dreamweaver CS 6 (Mac OS/ Windows)
3. Android-SDK
4. X-Code 4
5. Coreldraw X5
6. Apache Server
7. MySql