# BAB III
# ANALISIS DAN PERANCANGAN SISTEM

## 3.1. Analisis Sistem

Pada bagian analisis sistem merupakan penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru.

### 3.1.1 Analisis Masalah

Berdasarkan tinjauan analisis sistem yang telah dilakukan dengan melakukan evaluasi terhadap aplikasi yang akan dibangun. Beberapa permasalahan yang merupakan hasil dari analisis yang dilakukan dijadikan sebagai latar belakang dalam pembangunan apliksi E-Menu. Berikut permasalahan dari hasil analisis yang dilakukan :

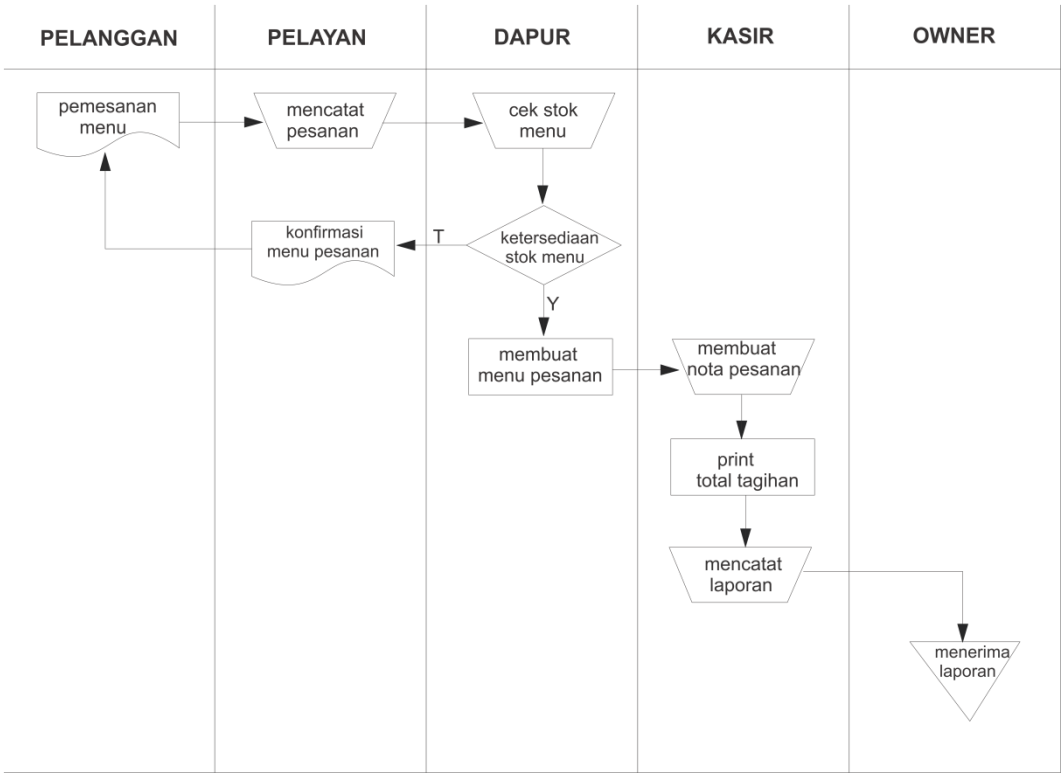
1. Kurang cepatnya proses pemesanan dan ketersediaan makanan karena jarak tempat makan jauh dari pantri.
2. Masih belum tertata rapih antrian pesanan di pantri.
3. Sulit untuk mencari atau menyimpan dokumen yang jumlahnya semakin banyak.
4. Terjadinya *redudansi* pada data stok barang.
5. Pembuatan laporan masih dilakukan secara konvensional sehingga sering terjadi keterlambatan, sedangkan laporan teresebut sangat berguna untuk pengontrolan dan pengawasan serta pengambilan keputusan.

### 3.1.2 Analisis prosedur yang sedang berjalan

Prosedur merupakan urutan kegiatan yang tepat dari tahapan-tahapan yang menerangkan mengenai proses apa yang dikerjakan, siapa yang mengerjakan proses tersebut, bagaimana proses tersebut dapat dikerjakan dan dokumen apa saja yang terlibat.

Berdasakan hasil wawancara dengan pihak Café Rumpoet, prosedur pemesanan makanan yang sedang berjalan di Café Rumpoet adalah sebagai berikut :

1. Pelanggan memilih menu yang akan dipesan. Setelah memilih menu yang akan dipesan, pelanggan memberitahu kepada pelayan cafe.
2. Pelayan menulis pesanan yang dipesan pelanggan pada nota pesanan 3 rangkap, untuk pantri, kasir dan pelanggan.
3. Petugas dapur atau koki menerima kertas pesanan yang diberikan oleh pelayan. Petugas dapur melihat ketersediaan pesanan yang dipesan, dan mengkonfirmasi kepada pelayan jika menu yang dipesan kosong, dan pelayan mengkonfirmasi kepada pelanggan.
4. Petugas dapur membuatkan makanan yang dipesan pelanggan.
5. Petugas kasir menerima kertas pesanan yang dipesan pelanggan.
6. Petugas kasir memberi tahu kepada pelanggan berapa total yang harus dibayar.
7. Pelanggan memberikan sejumlah uang ke petugas kasir sesuai dengan total harga yang harus dibayar.
8. Petugas kasir memberikan nota pembayaran kepada pelanggan.
9. Petugas kasir lalu mencatat laporan.
10. Pemilik menerima hasil pencatatan laporan.




**Gambar 3.1 *flowmap* prosedur yang sedang berjalan**

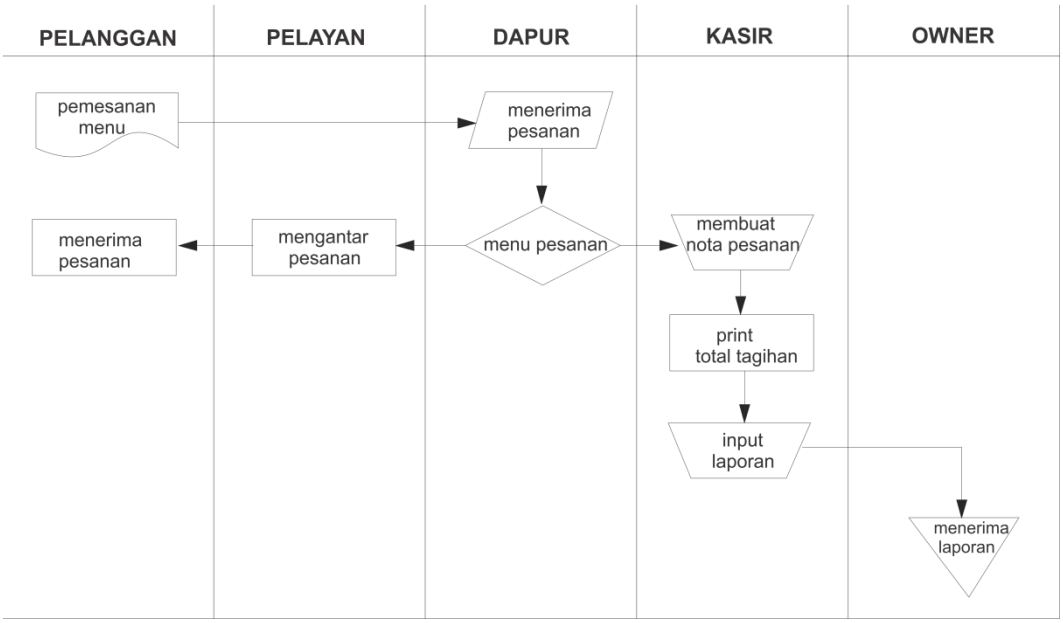
### 3.1.3 Analisis prosedur yang akan dibangun

Prosedur yang akan dibangun merupakan urutan kegiatan yang tepat dari tahapan-tahapan yang menerangkan mengenai proses apa yang akan dikerjakan, siapa yang akan mengerjakan proses tersebut, bagaimana proses tersebut dapat dikerjakan dan dokumen apa saja yang terlibat.

Berdasakan hasil wawancara dengan pihak Café Rumpoet, prosedur pemesanan yang akan dibangun di Café Rumpoet adalah sebagai berikut :

1. Pelanggan memilih menu di aplikasi android. Setelah memilih menu yang akan dipesan, pelanggan mengirim pesanan.
2. Petugas dapur menerima pemberitahuan pesanan pada aplikasi di komputer.
3. Petugas dapur membuatkan makanan yang dipesan pelanggan.
4. Pelayan mengantarkan pesanan pada pelanggan.
5. Petugas kasir memberi tahu kepada pelanggan berapa total yang harus dibayar.

6. Pelanggan memberikan sejumlah uang ke petugas kasir sesuai dengan total harga yang harus dibayar.
7. Petugas kasir memberikan nota pembayaran kepada pelanggan.
8. Petugas kasir lalu mencatat laporan.
9. Pemilik menerima hasil pencatatan laporan.




**Gambar 3.2 *flowmap* prosedur yang akan dibangun**

### 3.1.4 Analisis *Web Service*

Sistem yang akan dibangun merupakan sistem pemesanan menu makanan. Untuk aplikasi *mobile device* pada android agar dapat melakukan pemesanan, atau memasukan pesanan yang dipesan oleh pelanggan, maka harus disediakan perangkat yang dapat mentransfer data antara aplikasi *mobile device* dengan aplikasi pada dapur, dan kasir dalam hal ini *web base*.

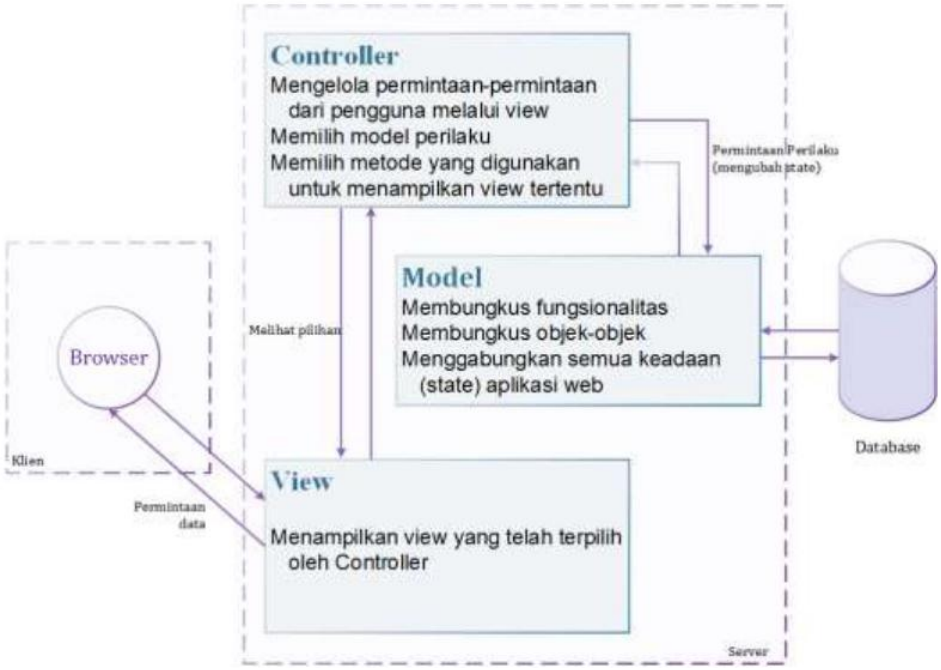
Dengan rincian langkah–langkah koneksi dari android ke *web base* sebagai berikut :

1. Pada aplikasi yang dirancang, saat aplikasi dijalankan secara otomatis *web base* akan bekerja untuk mem-parsing data dari *database* untuk digunakan oleh android.

2. *Parsing* data, dimulai dari *web service* mengambil *method* di dalam *web base* tersebut. Lalu sesuai dengan *method* tersebut akan mengakses data dari *database*.
3. Setelah akses data dari *database*, maka akan terbentuk data yang sudah terparsing, yang siap untuk digunakan oleh android.
4. Untuk pemanggilan data hasil parsing, saat pelanggan menjalankan aplikasi tersebut, maka aplikasi tersebut akan bisa terhubung dengan data hasil parsing melalui koneksi jaringan internet untuk mengoperasikannya.

### 3.1.5 Analisis *Framework CodeIgniter*

Aplikasi pada dapur, dan kasir ini berbasis *web base*. Dalam pembangunannya menggunakan *framework* berbasis *Model View Control* (MVC) yaitu *Codeigniter*. Dalam perancangannya penulis memisahkan antara logika aplikasi *web* dengan tampilan. Dalam prakteknya, hal itu memungkinkan halaman *web* mengandung *scripting* minimal karena tampilan dari *scripting* PHP. Seperti gambar 3.3 beserta penjelasan berikut.




**Gambar 3.3 Arsitektur *framework* CI**

Berikut penjelasan dari analisis *framework codeigniter*.

1. *Model*

   *Model* ini merupakan struktur data. Kelas *model* ini berisi fungsi yang membantu mengambil, memasukkan, dan meng-*update* informasi dalam *database*.

2. *View*

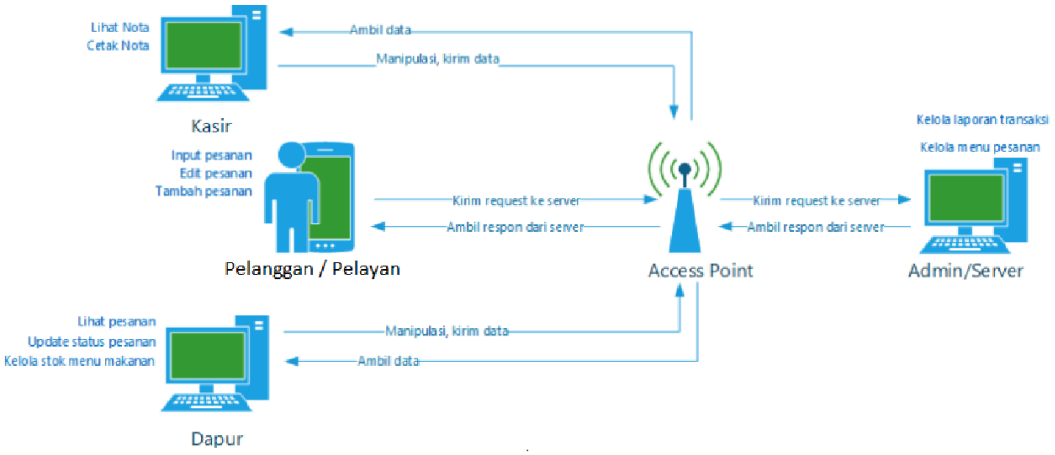
   *View* adalah informasi yang sedang disajikan kepada pengguna. View biasanya akan menjadi halaman antar muka *web*.

3. *Controller*

   *Controller* berfungsi sebagai perantara antara *Model*, *View*, dan sumber-sumber lain yang diperlukan untuk memproses permintaan HTTP dan menghasilkan suatu halaman antarmuka *web*.

### 3.1.6 Analisis Jaringan Arsitektur Sistem

Sesuai dengan analisis sistem yang sedang berjalan saat ini, maka sistem baru yang akan dibangun merupakan sistem pemesanan menu makanan di restoran Cafe Rumpoet yang terintegrasi secara *Access Point*, terdiri dari: aplikasi yang digunakan pelanggan, aplikasi pada admin, bagian dapur dan kasir.




**Gambar 3.4 Deskripsi sistem yang akan dibangun**

## 3.2 Spesifikasi Kebutuhan Perangkat Lunak

Menguraikan mengenai apa saja kebutuhan non fungsional dan fungsional dari aplikasi pemesanan menu makanan berbasis android platform di Café Rumpoet yang akan dibangun. Berikut uraiannya :

### 3.2.1 Kebutuhan Non Fungsional

Berikut merupakan tabel Spesifikasi Kebutuhan Perangkat Lunak Non-fungsional, seperti terlihat pada tabel 3.1 sebagai berikut.

**Tabel 3.1 Spesifikasi Kebutuhan Perangkat Lunak Non Fungsional *Mobile Base***

| Kode Kebutuhan | Kebutuhan |
|---|---|
| SKPL-NF-MD-1 | Aplikasi dibangun dalam dua sistem; sistem yang digunakan pelayan yaitu *mobile device* dengan OS android *platform* 4.0.3 |
| SKPL-NF- MD-2 | Aplikasi yang dibangun pada *mobile device* berbasis android sedangkan pada aplikasi *web base* menggunakan bahasa pemrograman php dengan *framework Codeigniter*. |

**Tabel 3.2 Spesifikasi Kebutuhan Perangkat Lunak Non Fungsional *Web base***

| Kode Kebutuhan | Kebutuhan |
|---|---|
| SKPL-NF-WB-1 | Aplikasi dapat berjalan pada *hardware* minimal processor dengan kecepatan 1.6 Ghz. |
| SKPL-NF- WB-2 | Pengguna dari sistem ini adalah pelayan, petugas dapur, dan petugas kasir. |
| SKPL-NF- WB-3 | Aplikasi ini dapat memberikan informasi *error* pada pengguna ketika terjadi kesalahan. |

### 3.2.2 Kebutuhan Fungsional

Berikut merupakan tabel Spesifikasi Kebutuhan Perangkat Lunak Fungsional, seperti yang terlihat pada tabel 3.2 sebagai berikut :

**Tabel 3.3 Spesifikasi Kebutuhan Perangkat Lunak Fungsional *Mobile Base***

| Kode Kebutuhan | Kebutuhan |
|---|---|

| | |
|---|---|
| SKPL-F-MB-1 | Aplikasi *mobile device* yang digunakan dapat melakukan *login* meja, *input* pesanan, *update* pesanan, dan lihat pesanan. |

**Tabel 3.4 Spesifikasi Kebutuhan Perangkat Lunak Fungsional *Web Base***

| Kode Kebutuhan | Kebutuhan |
|---|---|
| SKPL-F- WB-1 | Aplikasi pada bagian dapur dapat melakukan *update* status pesanan dan data stok. |
| SKPL-F- WB-2 | Aplikasi pada bagian kasir dapat melakukan *input* pembayaran, cetak nota, dan laporan. |

## 3.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional adalah sebuah langkah dimana seorang pembangun perangkat lunak menganalisis sumber daya yang akan digunakan perangkat lunak yang dibangun.

Analisis non fungsional tidak hanya menganalisis siapa saja yang akan menggunakan aplikasi tetapi juga menganalisis perangkat keras dan perangkat lunak, sehingga dapat menentukan kompabilitas aplikasi yang dibangun terhadap sumber daya yang ada. Setelah melakukan analisis non-fungsional, maka dilanjutkan kelangkah berikutnya, yaitu menentukan kebutuhan non-fungsional sistem yang akan dibangun untuk disesuaikan dengan fakta yang ada. Apabila terjadi ketidak cocokan antara fakta dan kebutuhan, maka perlu adanya penyesuaian fakta terhadap kebutuhan yang ada. Apabila kebutuhan tidak dipenuhi maka sistem yang akan dibangun tidak akan berjalan baik sesuai dengan yang diharapkan.

### 3.3.1 Analisis Pengguna

Analisis pengguna merupakan analisis terhadap pengguna yang akan menggunakan sistem yang telah dibangun. Perangkat keras dan perangkat lunak yang ada tidak akan berguna apabila tidak ada pengguna yang mengoperasikannya. Adapun pengguna yang terlibat dalam membangun aplikasi pemesanan menu makanan berbasis Android di Café Rumpoet ini terbagi menjadi 2, yaitu analisis pengguna yang ada di Cafe Rumpoet (lihat table 3.1) dan analisis pengguna yang akan menggunakan sistem (dapat dilihat pada Table 3.2).

### 1. Analisis Pengguna di Café Rumpoet

Terdapat 4 pengguna dalam sistem yang sedang berjalan di Café Rumpoet saat ini

a) Bagian dapur/koki, menerima pesanan menu makanan yang diberikan oleh pelayan. Mengecek menu yang akan disajikan, menyampaikan produksi makanan, membuat *taking order*, mengolah bahan makanan mejadi makanan siap saji.
b) Kasir, melakukan perhitungan total pembayaran dari jumlah pesanan pelanggan dengan memberikan nota/tagihan pembayaran kepada pelanggan. Membuat laporan transaksi harian dan bulanan, kemudian menyerahkan kepada *manager* dalam hal ini.
c) *Manager*, mengendalikan operasional maupun bidang administrasi, melakukan pemantauan seluruh proses kinerja mulai dari pelayan, petugas dapur, dan kasir. Mengelola serta merekap laporan-laporan transaksi pesanan menu makanan harian dan bulanan.

**Tabel 3.5 Analisis Pengguna di Café Rumpoet**

| Jabatan | Tanggung Jawab | Tingkat Pendidikan | Tingkat Keterampilan yang dimiliki | Pengalaman menggunakan komputer |
|---|---|---|---|---|
| Petugas Dapur / koki | Mengecek menu yang akan disajikan, menyampaikan produksi makanan, membuat *taking order*, mengolah bahan makanan mejadi makanan siap saji. | SMA/SMK – Diploma III | Menguasai komputer, mengerti cara penggunaan *web*, dan mengakses internet | Mengoperasikan komputer dan mengakses internet |
| Kasir | Memberi harga pada bill tamu, menerima uang pembayaran dari tamu, membuat laporan hasil | SMA/SMK – Diploma III | Menguasai komputer, mengerti cara penggunaan *web*, dan mengakses internet | Mengoperasikan komputer dan mengakses internet |

| | | | | |
|---|---|---|---|---|
| | penjualan | | | |
| *Manager* / admin | Mengendalikan operasional maupun bidang administrasi | Diploma III – StrataI | Menguasai komputer, mengerti cara penggunaan *web*, dan mengakses internet | Mengoperasikan komputer dan mengakses internet |

**2. Analisis pengguna yang menggunakan sistem**

Selain melakukan analisis pengguna yang ada di Cafe Rumpoet terhadap sistem yang sedang berjalan, dilakukan pula analisis pengguna yang akan menggunakan sistem, berikut kebutuhan *user* yang dibutuhkan :

a) Petugas dapur/koki, mengelola pesanan menu makanan yang telah dikirim oleh pelayan dan tersimpan di *database server*/admin. Kemudian menampilkan pesanan menu makanan yang diinputkan pelayan tersebut di bagian dapur.
b) Petugas kasir, melakukan perhitungan total pembayaran sesuai dengan data pesanan menu makanan yang dipesan pelanggan, kemudian memberikan struk dari jumlah total pembayaran.
c) Admin dalam hal ini *manager*, mengelola laporan transaksi harian dan bulanan, pengelolaan menu makanan, stok/persediaan menu makanan. Mengelola data *user* dalam hal ini pelayan, bagian dapur, dan bagian kasir.

**Tabel 3.6 Analisis pengguna yang akan menggunakan sistem**

| Pengguna sistem | Hak akses | Tingkat Keterampilan yang harus dimiliki | Pengalaman yang harus dimiliki | Jenis pelatihan yang akan diberikan |
|---|---|---|---|---|
| Petugas Dapur/koki | Melakukan *update* status pesanan dan data stok | Mengerti serta dapat menggunakan komputer | Mengoperasikan komputer | Cara menggunakan aplikasi |
| Kasir | Melakukan *input* pembayaran, mencetak nota,dan | Mengerti serta dapat menggunakan komputer | Mengoperasikan komputer | Cara menggunakanaplikasi |

| | | | | |
|---|---|---|---|---|
| | laporan | | | |
| *Manager* / admin | Mengelola data *user* , menu, meja, kategori. Mencetak laporan. | Mengerti serta dapat menggunakan komputer | Mengoperasikan komputer | Cara menggunakan aplikasi |

Kebutuhan pengguna untuk sistem ini minimal dapat mengoperasikan komputer. Berdasarkan data di atas, dapat disimpulkan bahwa setiap pengguna dapat mengakses dan menggunakan sistem yang akan dibangun.

### 3.3.2 Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras digunakan untuk mempermudah proses perancangan dan implementasi. Berdasarkan hasil pengamatan yang ada, spesifikasi perangkat keras yang dimiliki oleh Cafe Rumpoet adalah sebagai berikut:

1. *Processor* dengan kecepatan 1.6 GHz +*Motherboard*
2. Kapasitas *Harddisk* 80 GB
3. RAM 1024 MBVGA *Card on board*
4. *Monitor*
5. *Lan Card* 10/100Mbps *onboard*
6. *Mouse*
7. *Keyboard*

Sedangkan perangkat keras yang dibutuhkan untuk menjalankan sistem ini, digunakan perangkat keras sebagai berikut:

1. *Mobile device* Android minimal platform 4.0.3
2. *Processor* dengan kecepatan minimal 1.6 GHz +*Motherboard*
3. Kapasitas *Hard Disk* minimal 80 GB
4. RAM minimal 1024 MB
5. VGA *Card on board*
6. *Monitor*
7. Hub, USB *Wireless*
8. *Mouse*

9. *Keyboard*

Setelah melihat data tersebut, maka diperoleh kesimpulan bahwa Café Rumpoet sudah mencukupi untuk membangun sistem ini, dan hanya membutuhkan *mobile device* Android versi 4.0.3 serta USB *wireless access point* sebagai penghubung aplikasi android dengan aplikasi *web base*.

### 3.3.3 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan sehari-hari di Cafe Rumpoet sebagai berikut:

1. Sistem operasi Windows XP
2. *Web browser* menggunakan Mozilla Firefox dan Internet Explorer.

Sedangkan perangkat lunak yang dibutuhkan untuk membangun sistem ini digunakan perangkat lunak pendukung sebagai berikut:

1. Notepad ++, sebagai perangkat lunak yang digunakan untuk membangun aplikasi *web base* pada bagian kasir dan dapur.
2. *Java Development Kit*, untuk kompilasi kode-kode program pada aplikasi *mobile*.
3. *Java Runtime Edition*, sebagai platform untuk menjalankan sistem pada aplikasi *mobile*.
4. Android Studio, sebagai perangkat lunak yang digunakan untuk membangun aplikasi *mobile*.
5. Android SDK, untuk pengembang aplikasi android.
6. *Android Virtual Device*, sebagai *emulator* untuk menjalankan sistem android.

Kemudian Perangkat lunak yang diperlukan untuk menjalankan sistem ini adalah sebagai berikut:

1. *Wamp server*,aplikasi untuk membuat *server local*.
2. *Web browser*, untuk menampilkan aplikasi pada *web base*. Penulis rekomendasikan Chrome, Firefox, atau Internet Explorer versi terbaru.

## 3.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik sesuai kebutuhan.

### 3.4.1 *Use Case Diagram* pada Sistem Android

Diagram *Use Case* merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang akan menggambarkan bagaimana seseorang atau aktor akan menggunakan dan memanfaatkan sistem. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan sistem yang akan dibangun. *Use Case* terdiri dari tiga bagian yaitu definisi aktor, definisi *Use Case*, dan Skenario *Use Case*.

#### 3.4.1.1 Definisi Aktor pada Sistem Android

Pada tabel 3.7 berisi daftar aktor dan deskripsi peran mereka masing-masing dalam aplikasi ini.

**Tabel 3.7 Definisi aktor dalam sistem android**

| Kode | Aktor | Deskripsi |
|---|---|---|
| A-01 | Pelayan | Merupakan pengguna pada aplikasi android yang mempunyai hak akses *login*, *input* pesanan, *update* pesanan, dan lihat pesanan. |
| A-02 | Pelanggan | Merupakan pengguna pada aplikasi android yang mempunyai hak akses *input* pesanan, *update* pesanan, dan lihat pesanan. |

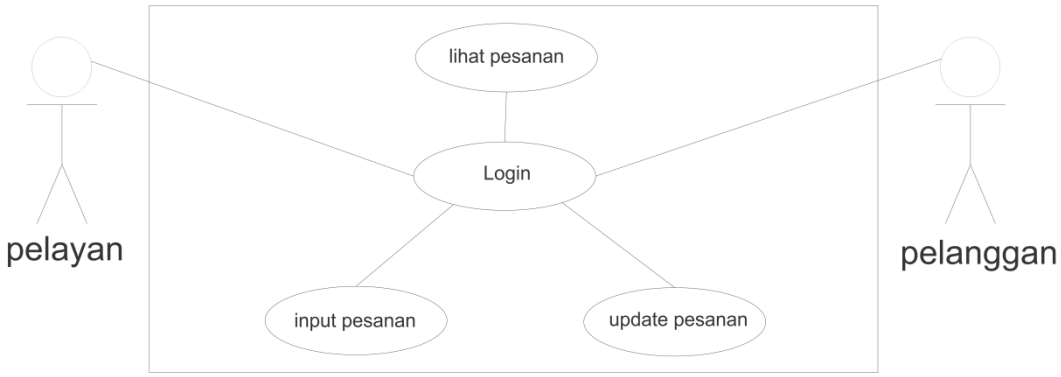
#### 3.4.1.2 Definisi *Use Case* pada Sistem Android

Berikut adalah deskripsi pendefinisian *Use Case* pada sistem android di Cafe Rumpoet, dapat dilihat pada tabel 3.8 berikut:

**Tabel 3.8 Definisi *Use Case* pada sistem android**

| No | Use Case | Deskripsi |
|---|---|---|
| 1 | *Login* | Merupakan proses otentikasi pengguna. |
| 2 | *Input* pesanan | Merupakan proses memasukkan data pesanan kedalam basis data dari aplikasi android. |
| 3 | *Update* Pesanan | Merupakan proses menambah data pesanan dari pesanan yang sudah ada ke dalam basis data dari aplikasi android. |
| 4 | Lihat Pesanan | Merupakan proses melihat pesanan yang sudah dipesan. |

*Use Case Diagram* pada sistem android di Cafe Rumpoet dengan aktor utama yaitu pelanggan yang berinterkasi dengan aplikasi android. Berikut ini adalah *use case diagram* aplikasi pemesanan menu makanan pada sistem android di Cafe Rumpoet yang dapat dilihat pada Gambar 3.5.




**Gambar 3.5 *Use case diagram* pada sistem android**

### 3.4.1.3 Skenario *Use Case* pada Sistem Android

Berdasarkan *use case* diagram di atas maka di buatlah skenario dari tiap proses yang ada pada *use case* diagram tersebut. Pada tabel 3.9 menggambarkan skenario pada *use case login*.

**Tabel 3.9 Skenario *Use case login***

| Identifikasi | |
|---|---|
| Nama *Use Case* | *Login* meja |
| Deskripsi | Proses otentikasi. |
| Aktor | Pelanggan |
| Kondisi awal | Pengguna berada di halaman *login* |

| Skenario Utama | |
|---|---|
| Aksi *actor user* | Respon Sistem |
| 1. Menjalankan aplikasi | 2. Menampilkan halaman *login* |
| 3. Memilih meja | 4. Mengecek valid tidaknya data masukan |
| | 5. *Login* berhasil. Masuk ke halaman utama. |
| Skenario Alternatif | |
| | 5b. *Login* berhasil. Masuk ke halaman utama halaman login |

Pada tabel 3.10 menggambarkan skenario pada *use case input* pesanan.

**Tabel 3.10 Skenario *Use case Input* Pesanan**

| Identifikasi | |
|---|---|
| Nama *Use Case* | *Input* Pesanan |
| Deskripsi | Proses untuk meng-*input* data menu yang dipesan konsumen. |
| Aktor | Pelanggan |
| Kondisi awal | Pelanggan berada dihalaman menu utama. |
| Skenario Utama | |
| Aksi *actor user* | Respon Sistem |
| 1. Memilih kategori pesanan | 2. Menampilkan halaman kategori pesanan |
| 3. Pilih menu dan jumlah yang dipesanan | 4. Menampilkan form menu pesanan |
| 5. Pilih menu "pesanan" dan klik kirim pesanan | 6. Mengirimkan data pesanan ke *database*. Kemudian data stok menu berkurang sesuai jumlah masukan |
| | 7. Menampilkan konfirmasi |
| Skenario Alternatif | |
| | 7b. data tidak lengkap maka menampilkan pesan "data belum lengkap, Mohon periksa kembali." |

Pada tabel 3.11 menggambarkan skenario *update* pesanan yang dilakukan oleh pelayan pada *use case update* pesanan.

**Tabel 3.11 Skenario *Use case update* pesanan pada aplikasi**

| Identifikasi | |
|---|---|
| Nama *Use Case* | *Update* pesanan |

| Deskripsi | Proses untuk meng-*update* pesanan |
|---|---|
| Aktor | Pelanggan |
| Kondisi awal | berada dihalaman menu utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Memilih kategori pesanan | 2. Menampilkan halaman kategori pesanan |
| 3. Pilih menu dan jumlah yang dipesanan | 4. Menampilkan form menu pesanan |
| 5. Pilih menu pesanan dan klik kirim pesanan | 6. Mengirimkan data pesanan ke *database*. Kemudian data stok menu berkurang sesuai jumlah masukan |
| | 7. Menampilkan konfirmasi berhasil |
| Skenario Alternatif | |
| | 7b. data tidak lengkap maka menampilkan pesan "data belum lengkap, Mohon periksa kembali." |

Pada tabel 3.12 menggambarkan skenario *update* pesanan yang dilakukan oleh pelayan pada *use case update* pesanan.

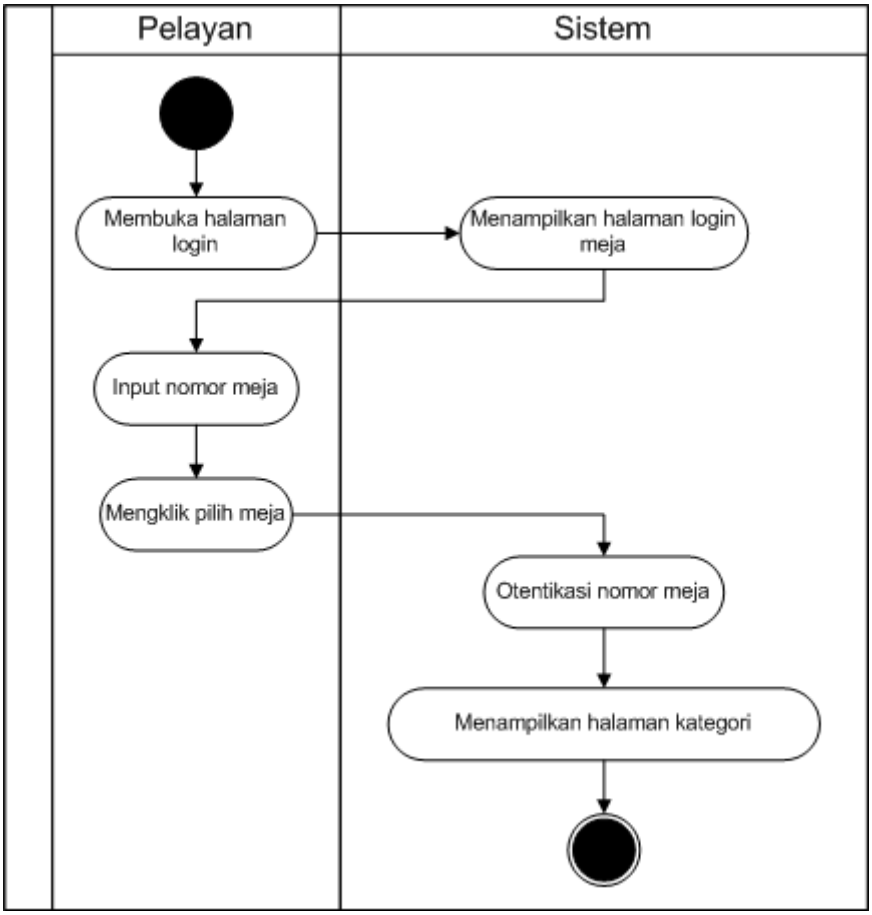
**Tabel 3.12 Skenario *Use case* lihat pesanan pada aplikasi**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Lihat Pesanan |
| Deskripsi | Proses untuk melihat pesanan |
| Aktor | Pelanggan |
| Kondisi awal | berada dihalaman menu utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Memilih menu "status pesanan" | 2. Menampilkan halaman status pesanan |

### 3.4.2 *Activity Diagram* pada Sistem Android

#### 3.4.2.1 *Login* pada Sistem Android

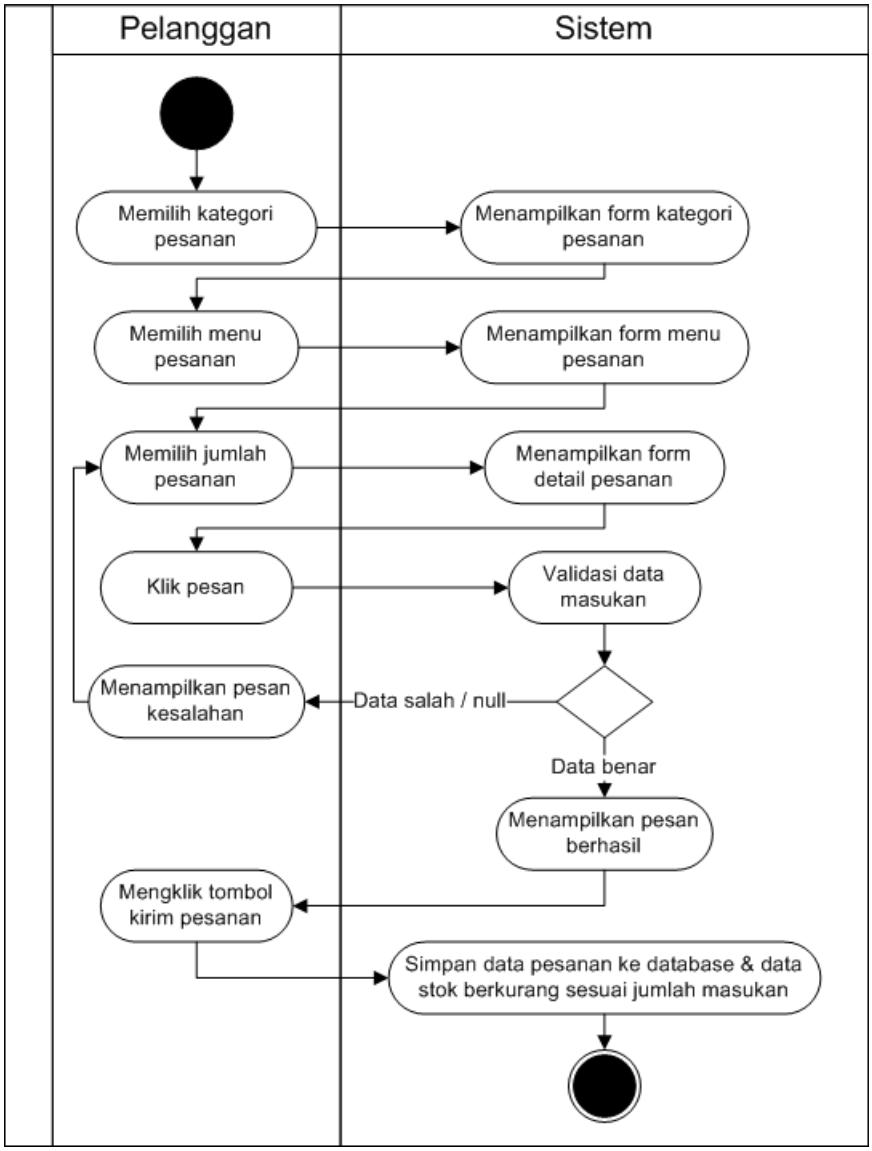
Aktivitas *login* merupakan aktivitas yang dilakukan oleh pelayan untuk proses otentikasi. Seluruh aktivitas *login* terlihat pada gambar 3.6




**Gambar 3.6** ***Activity diagram login***

### 3.4.2.2 *Input* Pesanan pada Sistem Android

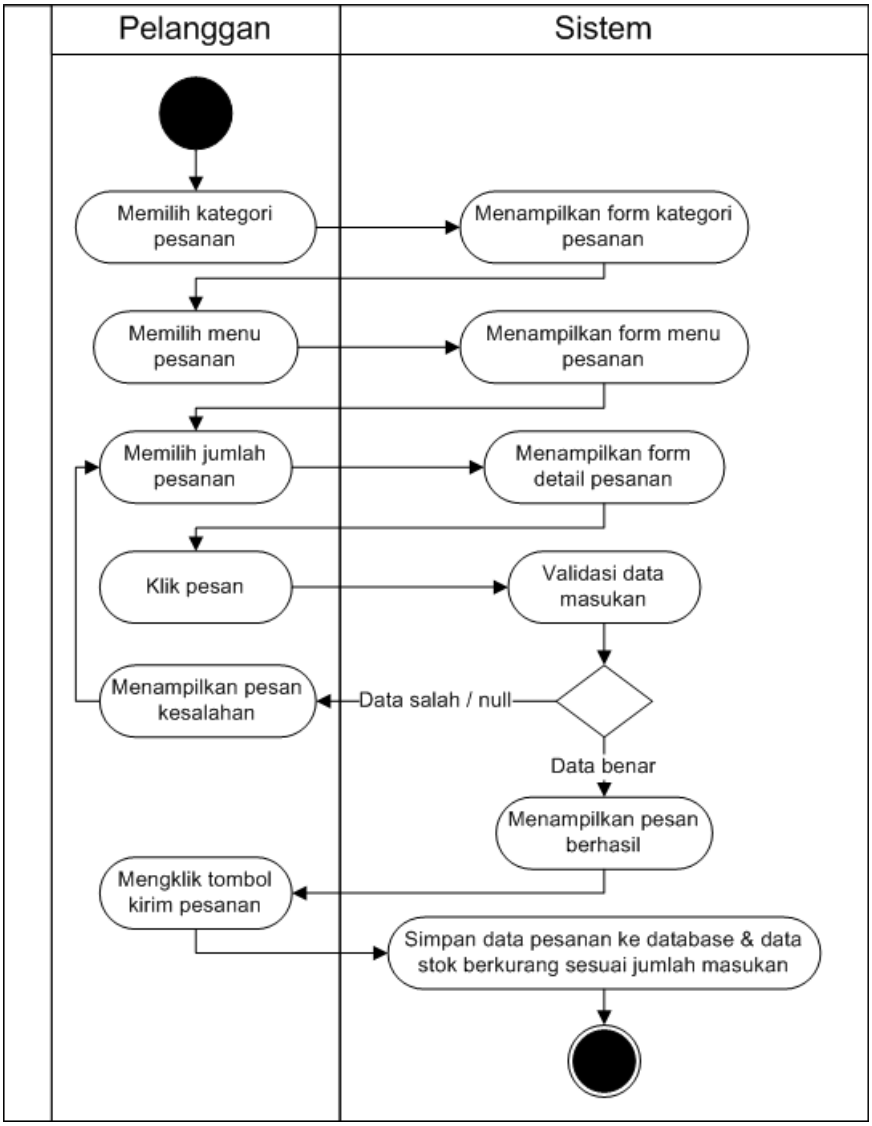
Aktivitas *Input* pesanan merupakan aktivitas yang dilakukan oleh pelanggan untuk meng-*input* data pesana. Seluruh aktivitas *input* pesanan terlihat pada *activity diagram* gambar 3.7




**Gambar 3.7 *Activity diagram input* pesanan**

### *3.4.2.3 Update* Pesanan pada Sistem Android

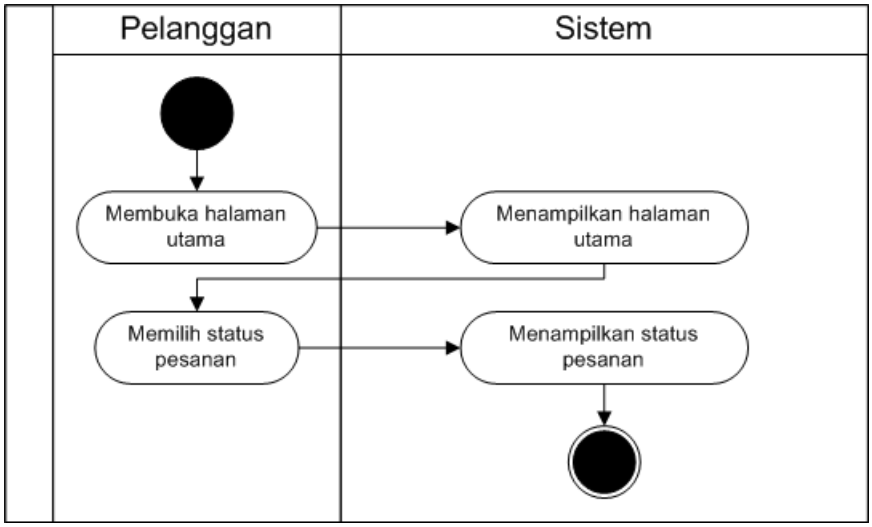
Aktivitas *update* pesanan merupakan aktivitas yang dilakukan oleh pelanggan untuk melakukan tambahan pesanan. Seluruh aktivitas *update* pesanan terlihat pada *activity diagram* terlihat pada *activity diagram* gambar 3.8




**Gambar 3.8 *Activity diagram update* pesanan**

### 3.4.2.4 Lihat Pesanan pada Sistem Android

Aktivitas lihat pesanan merupakan aktivitas yang dilakukan oleh pelanggan untuk melihat pesanan. Seluruh aktivitas lihat pesanan terlihat pada *activity diagram* pada gambar 3.9.

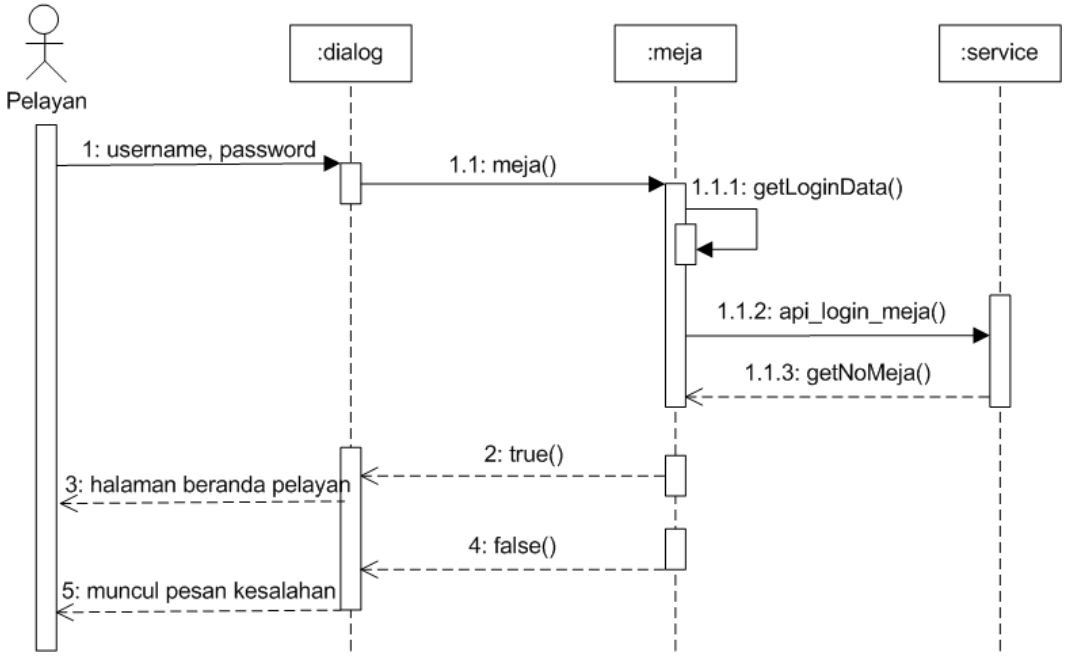



**Gambar 3.9 *Activity diagram* lihat pesanan**

### 3.4.3 *Sequence Diagram* pada Sistem Android

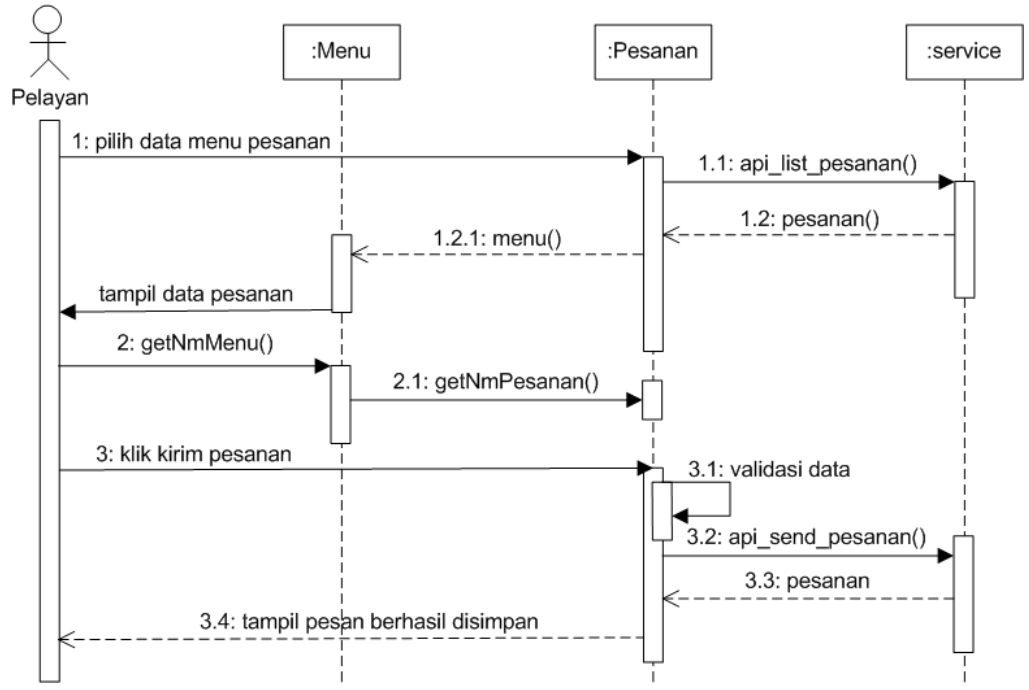
Merupakan bentuk dari *interaction diagram*, menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa *message* yang di gambarkan terhadap waktu. Masing–masing objek termasuk *actor* memiliki *line vertical message* digambarkan garis berpanah dari satu objek ke objek lainnya, dan *activation bar* menunjukan lamanya eksekusi sebuah proses. Sebuah *sequence diagram*, secara khusus menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan – pesan yang melewati objek–objek ini di dalam *use case*. Berikut ini adalah *sequence diagram* dari sistem yang akan dibangun.

### 3.4.3.1 *Sequence diagram login*



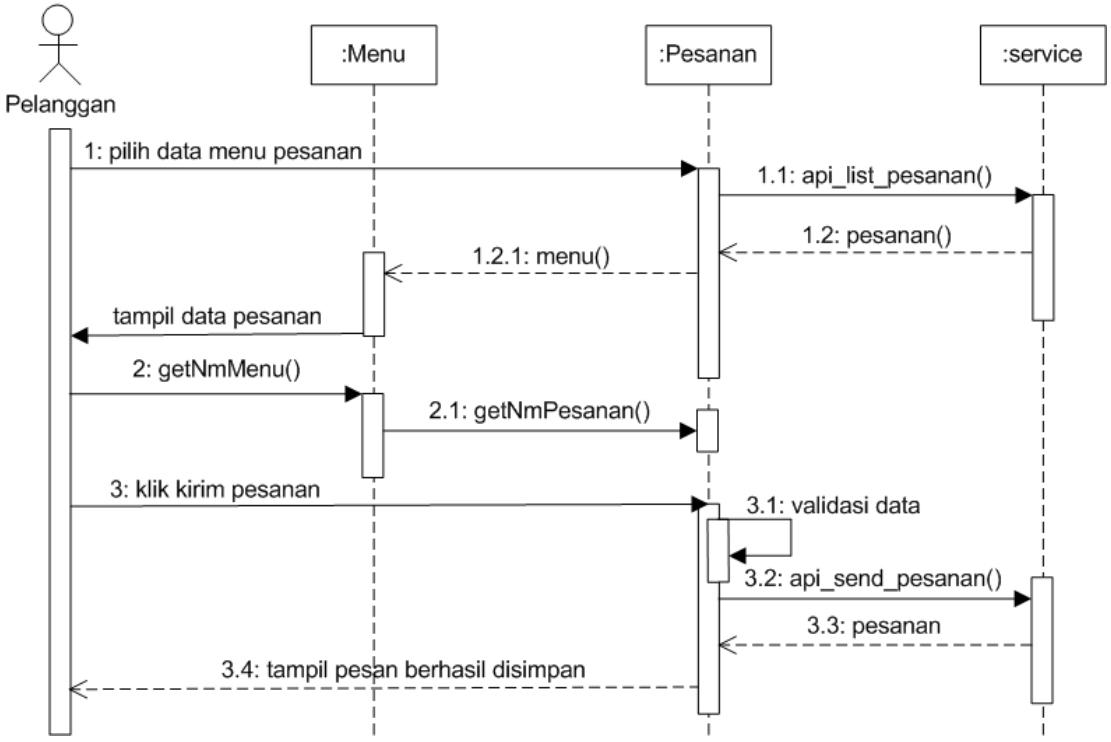

**Gambar 3.10 *Sequence diagram login***

### 3.4.3.2 *Sequence diagram Input* pesanan



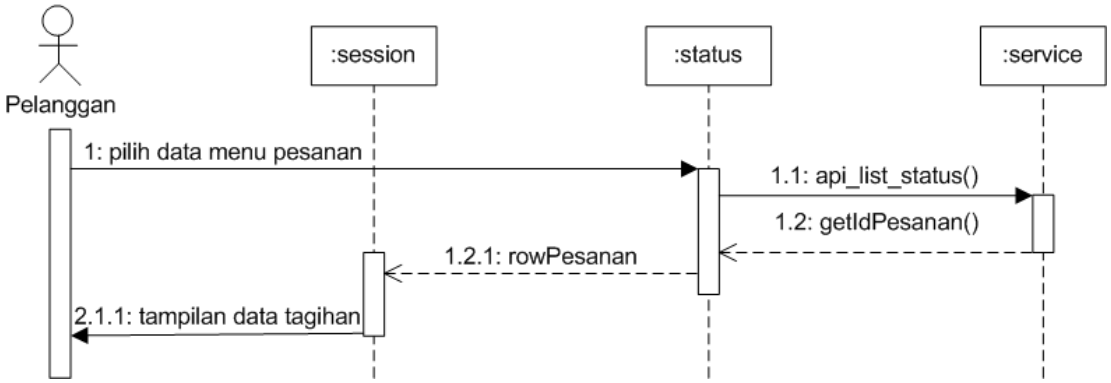

**Gambar 3.11 *Sequence diagram Input* pesanan**

### 3.4.3.3 *Sequence diagram Update* pesanan



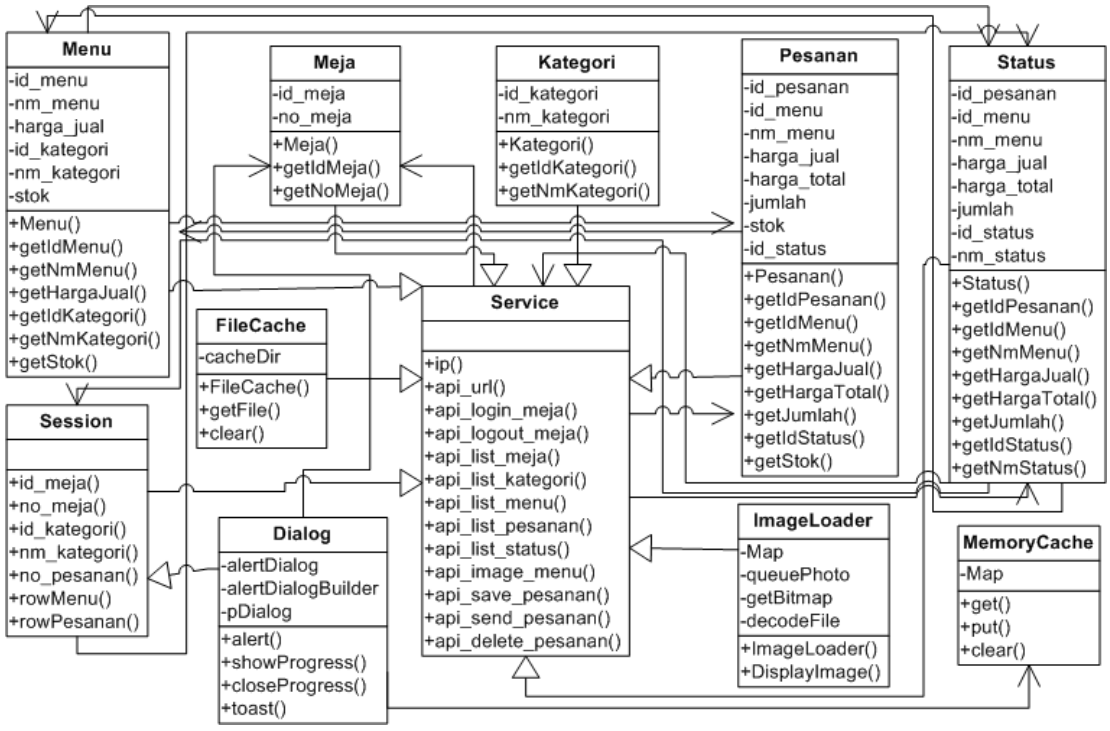

**Gambar 3.12 *Sequence diagram Update* pesanan**

### 3.4.3.4 *Sequence diagram* lihat pesanan




**Gambar 3.13 *Sequence diagram* lihat pesanan**

### 3.4.4 *Class Diagram* pada Aplikasi Android




**Gambar 3.14 *Class Diagram* pada Aplikasi Android**

### 3.4.5 *Use Case Diagram* pada *Web Base*

Diagram *Use Case* merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang akan menggambarkan bagaimana seseorang atau aktor akan menggunakan dan memanfaatkan sistem. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan sistem yang akan dibangun. *Use Case* terdiri dari tiga bagian yaitu definisi aktor, definisi *Use Case*, dan Skenario *Use Case*.

#### 3.4.5.1 Definisi Aktor

Berikut adalah deskripsi pendefinisian aktor yang berperan pada sistem aplikasi pemesanan menu makanan restoran berbasis android platform di Cafe Rumpoet, dapat dilihat pada tabel 3.13 berikut:

**Tabel 3.13 Definisi aktor dalam sistem *web base***

| Kode | Aktor | Deskripsi |
|---|---|---|
| B-01 | *Manager*/Admin | Merupakan pengguna pada aplikasi *web* yang mempunyai hak akses untuk tambah, edit, hapus data menu, data meja, data kategori,data pengguna dan cetak laporan. |
| B-02 | Dapur | Merupakan Pengguna pada aplikasi *web* yang akan menampilkan data pesanan makanan, *update* status pesananan menu, dan *update* stok. |
| B-03 | Kasir | Merupakan pengguna pada aplikasi *web* yang akan menampilkan data pesanan, total harga yang harus dibayar, mencetak nota, dan laporan. |

### 3.4.5.2 Definisi *Use Case*

Berikut adalah deskripsi pendefinisian *Use Case* pada *web service* di Cafe Rumpoet, dapat dilihat pada tabel 3.14 berikut:

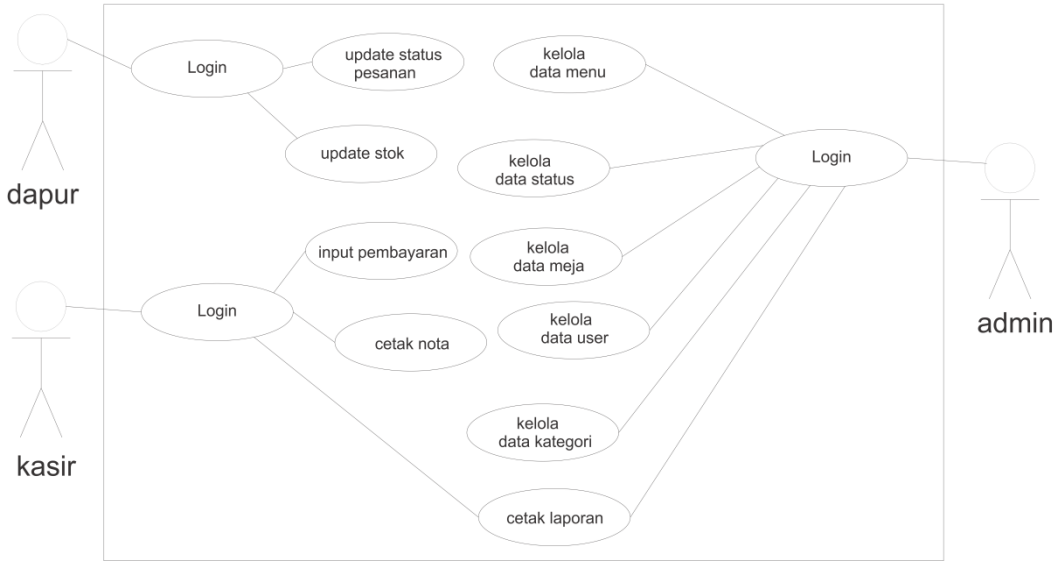
**Tabel 3.14 Definisi *Use Case***

| No | Use Case | Deskripsi |
|---|---|---|
| 1 | Login | Merupakan proses pengecekan hak akses siapa saja yang berhak mengakses proses pengelolaan data, seperti admin, kepala dapur, atau kasir. *Login* wajib untuk fungsi- fungsi yang |

| | | |
|---|---|---|
| | | berkaitan dengan akses pengubahan kebasis data, oleh karena itu fungsi yang melakukan perubahan basis data harus mengecek validasi *user* yang mengakses fungsi-fungsi ini. |
| 2 | Kelola Data Menu | Merupakan proses menambah, mengedit, dan menghapus data menu ke dalam basis data. |
| 3 | Kelola Data Meja | Merupakan proses menambah, mengedit, dan menghapus data meja ke dalam basis data. |
| 4 | Kelola Data Kategori | Merupakan proses menambah, mengedit, dan menghapus data kategori ke dalam basis data. |
| 5 | Kelola Data Status | Merupakan proses menambah, mengedit, dan menghapus data status ke dalam basis data. |
| 6 | Kelola Data *User* | Merupakan proses menambah data *user* ke dalam basis data. |
| 7 | *Update* Status Pesanan | Merupakan proses pengubahan status pesanan dari awal pesan dengan status “antri” sampai semua pesanan selesai. |
| 8 | *Update* stok menu | Merupakan proses pengubahan jumlah stok data menu. |
| 9 | *Input* pembayaran | Merupakan proses penginputan data pembayaran dari data pesanan yang telah selesai di proses. |
| 10 | Cetak Nota | Merupakan proses mencetak nota setelah proses *input* pembayaran. |
| 11 | Cetak Laporan | Merupakan proses mencetak laporan baik harian maupun bulanan. |

*Use Case Diagram* pada sistem aplikasi pemesanan menu makanan berbasis android *platform* di Cafe Rumpoet terdiri dari tiga aktor yang berperan yaitu, Admin, bagian Dapur, dan bagian Kasir. Aktor pertama yaitu Admin hanya dapat berinterkasi dengan aplikasi *web base* dengan akses *login* sebagai admin. Aktor petugas dapur hanya dapat berinteraksi dengan aplikasi *web base* dengan akses *login* sebagai dapur. Aktor petugas kasir hanya dapat berinteraksi dengan

aplikasi *web base* dengan akses *login* sebagai kasir. Berikut ini adalah *use case diagram* aplikasi pemesanan menu makanan di Cafe Rumpoet.




**Gambar 3.15 *Use case diagram* pada aplikasi *Web Base***

### 3.4.5.3 Skenario *Use Case* pada bagian Admin

Skenario *Use Case* pada bagian Admin merupakan deskripsi urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor (admin) terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor (admin). Skenario *use case* pada bagian admin terdiri dari skenario *login* admin, tambah data menu pesanan, edit menu pesanan, hapus menu pesanan, tambah meja, edit meja, hapus meja, tambah kategori, edit kategori, hapus kategori, laporan menu, laporan transaksi bulanan, laporan transaksi harian, *login* dapur, lihat pesanan, *update* status menu pesanan, *login* kasir, lihat tagihan, dan cetak tagihan.

Berikut ini penjelasan dari masing-masing skenario *use case* pada bagian admin yang terlihat pada tabel 3.15 menggambarkan skenario pada *use case login* admin.

**Tabel 3.15 Skenario *use case login* admin**

| Identifikasi | |
|---|---|
| Nama *Use Case* | *login* |
| Deskripsi | Proses otentifikasi admin untuk masuk pada sistem |
| Aktor | Admin |
| Kondisi awal | tampil halaman *login* |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Memasukan *user name*, *password* dan mengklik tombol *login* | 2. Mengecek valid tidaknya data masukan |
| | 3. *Login* berhasil. Masuk ke halaman beranda aplikasi admin. |
| Skenario Alternatif | |
| | 3b. Jika *login* gagal karena data tidak valid, maka sistem menampilkan pesan kesalahan. "Nama pengguna atau *password* salah." |

Pada tabel 3.16 menggambarkan skenario tambah menu pesanan pada *use case* tambah data menu.

**Tabel 3.16 Skenario *use case* tambah data menu**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Tambah Data Menu |
| Deskripsi | Proses untuk menambah data menu makanan |
| Aktor | Admin |
| Kondisi awal | berada dihalaman data menu |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data menu | 2. Menampilkan data menu |
| 3. Memilih tambah data menu | 4. Menampilkan *form* tambah menu pesanan |
| 5. Memasukkan data sesuai kolom isian yang tersedia | |
| 6. Mengklik simpan data | 7. Mengecek valid tidaknya data masukan |

| | |
|---|---|
| | 8. Menyimpan data menu ke dalam *database*. |
| | 9. Menampilkan pesan data menu berhasil disimpan |
| Skenario Alternatif | |
| | 9b. data tidak valid, maka sistem menampilkan pesan kesalahan, "semua *field* harus diisi dengan benar." |

Pada tabel 3.17 menggambarkan skenario hapus menu pesanan yang dilakukan oleh admin pada *use case* hapus data menu.

**Tabel 3.17 Skenario *use case* hapus data menu**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Hapus Data Menu |
| Deskripsi | Proses untuk menghapus data menu dalam *database* |
| Aktor | Admin |
| Kondisi awal | berada dihalaman data menu |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data menu | 2. Menampilkan data menu |
| 3. Mencari data menu yang akan di hapus | |
| 4. Memilih hapus | 5. Menampilkan pesan Anda yakin? |
| 6. Mengklik ya hapus | 7. Menghapus data menu dari *database* |
| Skenario Alternatif | |
| | 5b. Jika "batal" menghapus, maka sistem mengembalikan tampilan ke halaman utama data menu. |

Pada 3.18 berikut menggambarkan skenario untuk mengedit menu pesanan yang dilakukan oleh admin pada *use case* edit data menu.

**Tabel 3.18 Skenario *use case* edit menu pesanan**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Edit Data Menu |
| Deskripsi | Proses untuk mengedit data menu dalam *database* |

| Aktor | Admin |
|---|---|
| Kondisi awal | berada di halaman menu |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data menu | 2. Menampilkan data menu |
| 3. Memilih data menu yang akan di edit | |
| 4. Mengklik edit | 5. Menampilkan *form* edit menu pesanan |
| 6. Mengedit data menu | |
| 7. Mengklik simpan data | 8. Melakukan proses validasi |
| | 9. Menyimpan data yang telah diedit dalam *database* |
| | 10. Menampilkan pesan bahwa data menu berhasil disimpan |
| Skenario Alternatif | |
| | 10b. data tidak valid, sistem menampilkan pesan kesalahan "semua field harus diisi dengan benar". |

Pada tabel 3.19 menggambarkan skenario tambah meja pada *use case* tambah data meja.

**Tabel 3.19 Skenario *use case* tambah data meja**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Tambah Data Meja |
| Deskripsi | Proses untuk menambah data meja |
| Aktor | Admin |
| Kondisi awal | berada dihalaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data meja | 2. Menampilkan data meja |
| 3. Memilih tambah data meja | 4. Menampilkan form tambah meja |
| 5. Memasukkan data sesuai kolom isian yang tersedia | |
| 6. Mengklik simpan data | 7. Mengecek valid tidaknya data masukan |

| | |
|---|---|
| | 8. Menyimpan data meja ke dalam *database*. |
| | 9. Menampilkan pesan data meja berhasil disimpan |
| Skenario Alternatif | |
| | 9b. data tidak valid, maka sistem menampilkan pesan kesalahan, "semua *field* harus diisi dengan benar." |

Pada tabel 3.20 menggambarkan skenario hapus meja yang dilakukan oleh admin pada *use case* hapus data meja.

**Tabel 3.20 Skenario *use case* hapus data meja**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Hapus Data Meja |
| Deskripsi | Proses untuk menghapus data meja dalam *database* |
| Aktor | Admin |
| Kondisi awal | berada dihalaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data meja | 2. Menampilkan data meja |
| 3. Mencari data meja yang akan di hapus | |
| 4. Memilih hapus | 5. Menampilkan pesan Anda yakin? |
| 6. Mengklik ya hapus | 7. Menghapus data meja dari *database* |
| Skenario Alternatif | |
| | 5b. Jika "batal" menghapus, maka sistem mengembalikan tampilan ke halaman utama data meja. |

Pada tabel 3.21 menggambarkan skenario untuk mengedit meja yang dilakukan oleh admin pada *use case* edit data meja.

**Tabel 3.21 Skenario *use case* edit meja**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Edit Data meja |
| Deskripsi | Proses untuk mengedit data meja dalam *database* |

| Aktor | Admin |
|---|---|
| Kondisi awal | berada di halaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data meja | 2. Menampilkan data meja |
| 3. Memilih data meja yang akan di edit | |
| 4. Mengklik edit | 5. Menampilkan form edit meja pesanan |
| 6. Mengedit data meja | |
| 7. Mengklik simpan data | 8. Melakukan proses validasi |
| | 9. Menyimpan data yang telah diedit dalam *database* |
| | 10. Menampilkan pesan bahwa data meja berhasil disimpan |
| Skenario Alternatif | |
| | 10b. data tidak valid, sistem menampilkan pesan kesalahan “semua field harus diisi dengan benar”. |

Pada tabel 3.22 menggambarkan skenario tambah kategori menu makanan pada *use case* tambah data kategori.

**Tabel 3.22 Skenario *use case* tambah kategori**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Tambah Data Kategori |
| Deskripsi | Proses untuk menambah data kategori |
| Aktor | Admin |
| Kondisi awal | berada dihalaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data kategori | 2. Menampilkan data kategori |
| 3. Memilih tambah data kategori | 4. Menampilkan form tambah kategori |
| 5. Memasukkan data sesuai kolom isian yang tersedia | |
| 6. Mengklik simpan data | 7. Mengecek valid tidaknya data masukan |
| | 8. Menyimpan data |

| | |
|---|---|
| | kategori ke dalam *database*. |
| | 9. Menampilkan pesan data kategori berhasil disimpan |
| Skenario Alternatif | |
| | 9b. data tidak valid, maka sistem menampilkan pesan kesalahan, “semua field harus diisi dengan benar.” |

Pada tabel 3.23 menggambarkan skenario hapus kategori menu yang dilakukan oleh admin pada *use case* hapus data kategori.

**Tabel 3.23 Skenario *use case* hapus kategori**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Hapus data kategori |
| Deskripsi | Proses untuk menghapus data kategori dalam *database* |
| Aktor | Admin |
| Kondisi awal | berada dihalaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data kategori | 2. Menampilkan data kategori |
| 3. Mencari data kategori yang akan di hapus | |
| 4. Memilih hapus | 5. Menampilkan pesan Anda yakin? |
| 6. Mengklik ya hapus | 7. Menghapus data kategori dari *database* |
| Skenario Alternatif | |
| | 5b. Jika “batal” menghapus, maka sistem mengembalikan tampilan ke halaman utama data kategori. |

Pada tabel 3.24 menggambarkan skenario edit kategori makanan yang dilakukan oleh admin pada *use case* edit data kategori.

**Tabel 3.24 Skenario *use case* edit data kategori**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Edit Data kategori |

| Deskripsi | Proses untuk mengedit data kategori dalam *database* |
|---|---|
| Aktor | Admin |
| Kondisi awal | berada di halaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data kategori | 2. Menampilkan data kategori |
| 3. Memilih data kategori yang akan di edit | |
| 4. Mengklik edit | 5. Menampilkan form edit kategori pesanan |
| 6. Mengedit data kategori | |
| 7. Mengklik simpan data | 8. Melakukan proses validasi |
| | 9. Menyimpan data yang telah diedit dalam *database* |
| | 10. Menampilkan pesan bahwa data kategori berhasil disimpan |
| Skenario Alternatif | |
| | 10b. data tidak valid, sistem menampilkan pesan kesalahan “semua field harus diisi dengan benar”. |

Pada tabel 3.25 menggambarkan skenario tambah data *user* pada *use case* tambah data *user* .

**Tabel 3.25 Skenario *use case* tambah data *user***

| Identifikasi | |
|---|---|
| Nama *Use Case* | Tambah Data *user* |
| Deskripsi | Proses untuk menambah data *user* |
| Aktor | Admin |
| Kondisi awal | berada dihalaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data *user* | 2. Menampilkan data *user* |
| 3. Memilih tambah data *user* | 4. Menampilkan *form* tambah *user* |
| 5. Memasukkan data sesuai kolom isian | |

| | |
|---|---|
| yang tersedia | |
| 6. Mengklik simpan data | 7. Mengecek valid tidaknya data masukan |
| | 8. Menyimpan data *user* ke dalam *database*. |
| | 9. Menampilkan pesan data *user* berhasil disimpan |
| Skenario Alternatif | |
| | 9b. data tidak valid, maka sistem menampilkan pesan kesalahan, "semua *field* harus diisi dengan benar." |

Pada tabel 3.26 menggambarkan skenario hapus data *user* yang dilakukan oleh admin pada *use case* hapus data *user*.

**Tabel 3.26 Skenario *use case* hapus data *user***

| Identifikasi | |
|---|---|
| Nama *Use Case* | Hapus Data *user* |
| Deskripsi | Proses untuk menghapus data *user* dalam *database* |
| Aktor | Admin |
| Kondisi awal | berada dihalaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data *user* | 2. Menampilkan data *user* |
| 3. Mencari data *user* yang akan di hapus | |
| 4. Memilih hapus | 5. Menampilkan pesan Anda yakin? |
| 6. Mengklik ya hapus | 7. Menghapus data *user* dari *database* |
| Skenario Alternatif | |
| | 5b. Jika "batal" menghapus, maka sistem mengembalikan tampilan ke halaman utama data *user*. |

Pada 3.27 berikut menggambarkan skenario untuk mengedit data *user* yang dilakukan oleh admin pada *use case* edit data *user*.

**Tabel 3.27 Skenario *use case* edit data *user***

| Identifikasi |
|---|

| Nama *Use Case* | Edit data *user* |
|---|---|
| Deskripsi | Proses untuk mengedit data *user* dalam *database* |
| Aktor | Admin |
| Kondisi awal | berada di halaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data *user* | 2. Menampilkan data *user* |
| 3. Memilih data *user* yang akan di edit | |
| 4. Mengklik edit | 5. Menampilkan *form* edit *user* |
| 6. Mengedit data *user* | |
| 7. Mengklik simpan data | 8. Melakukan proses validasi |
| | 9. Menyimpan data yang telah diedit dalam *database* |
| | 10. Menampilkan pesan bahwa data *user* berhasil disimpan |
| Skenario Alternatif | |
| | 10b. data tidak valid, sistem menampilkan pesan kesalahan "semua *field* harus diisi dengan benar". |

Pada tabel 3.28 menggambarkan skenario tambah data status pada *use case* tambah data status.

**Tabel 3.28 Skenario *use case* tambah data status**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Tambah data status |
| Deskripsi | Proses untuk menambah data status |
| Aktor | Admin |
| Kondisi awal | berada dihalaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data status | 2. Menampilkan data status |
| 3. Memilih tambah data status | 4. Menampilkan form tambah status |
| 5. Memasukkan data sesuai kolom isian yang tersedia | |

| | |
|---|---|
| 6. Mengklik simpan data | 7. Mengecek valid tidaknya data masukan |
| | 8. Menyimpan data status ke dalam *database*. |
| | 9. Menampilkan pesan data status berhasil disimpan |
| Skenario Alternatif | |
| | 9b. data tidak valid, maka sistem menampilkan pesan kesalahan, "semua *field* harus diisi dengan benar." |

Pada tabel 3.29 menggambarkan skenario hapus data status yang dilakukan oleh admin pada *use case* hapus data status.

**Tabel 3.29 Skenario *use case* hapus data status**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Hapus data status |
| Deskripsi | Proses untuk menghapus data status dalam *database* |
| Aktor | Admin |
| Kondisi awal | berada dihalaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data status | 2. Menampilkan data status |
| 3. Mencari data status yang akan di hapus | |
| 4. Memilih hapus | 5. Menampilkan pesan Anda yakin? |
| 6. Mengklik ya hapus | 7. Menghapus data status dari *database* |
| Skenario Alternatif | |
| | 5b. Jika "batal" menghapus, maka sistem mengembalikan tampilan ke halaman utama data status. |

Pada 3.30 berikut menggambarkan skenario untuk mengedit data status yang dilakukan oleh admin pada *use case* edit data status.

**Tabel 3.30 Skenario *use case* edit data status**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Edit data status |

| Deskripsi | Proses untuk mengedit data status dalam *database* |
|---|---|
| Aktor | Admin |
| Kondisi awal | berada di halaman utama |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman data status | 2. Menampilkan data status |
| 3. Memilih data status yang akan di edit | |
| 4. Mengklik edit | 5. Menampilkan *form* edit status |
| 6. Mengedit data status | |
| 7. Mengklik simpan data | 8. Melakukan proses validasi |
| | 9. Menyimpan data yang telah diedit dalam *database* |
| | 10. Menampilkan pesan bahwa data status berhasil disimpan |
| Skenario Alternatif | |
| | 10b. Data tidak valid, sistem menampilkan pesan kesalahan "semua *field* harus diisi dengan benar". |

Pada table 3.31 menggambarkan skenario untuk cetak laporan transaksi tahunan yang dilakukan oleh admin pada *use case* cetak laporan transaksi tahunan.

**Tabel 3.31 Skenario *use case* laporan transaksi tahunan**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Cetak Laporan Transaksi Tahunan |
| Deskripsi | Proses untuk Cetak Laporan Transaksi Tahunan dari *database* |
| Aktor | Admin |
| Kondisi awal | berada di halaman laporan |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka menu laporan transaksi | 2. Menampilkan halaman laporan transaksi tahunan |
| 3. Memilih tahun | |

| | |
|---|---|
| 4. Mengklik *search* | 5. Melakukan proses pencarian berdasarkan tahun |
| | 6. Menampilkan data yang dicari dari *database* |
| 7. Mengklik cetak laporan | 8. Menampilkan laporan transaksi tahunan |
| Skenario Alternatif | |
| | 6b. Data tidak ditemukan, sistem menampilkan laporan kosong". |

Pada table 3.32 menggambarkan skenario untuk cetak laporan transaksi bulanan yang dilakukan oleh admin pada *use case* cetak laporan transaksi bulanan.

**Tabel 3.32 Skenario *use case* laporan transaksi bulanan**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Cetak Laporan Transaksi Bulanan |
| Deskripsi | Proses untuk Cetak Laporan Transaksi Bulanan dari *database* |
| Aktor | Admin |
| Kondisi awal | berada di halaman laporan |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka menu laporan transaksi | 2. Menampilkan halaman laporan transaksi bulanan |
| 3. Memilih bulanan | |
| 4. Mengklik *search* | 5. Melakukan proses pencarian berdasarkan bulan |
| | 6. Menampilkan data yang dicari dari *database* |
| 7. Mengklik cetak laporan | 8. Menampilkan laporan transaksi bulanan |
| Skenario Alternatif | |
| | 6b. Data tidak ditemukan, sistem menampilkan laporan kosong". |

Pada tabel 3.33 menggambarkan skenario untuk cetak laporan transaksi harian yang dilakukan oleh admin pada *use case* cetak laporan transaksi harian.

**Tabel 3.33 Skenario *use case* cetak laporan transaksi harian**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Cetak Laporan Transaksi Harian |

| Deskripsi | Proses untuk Cetak Laporan Transaksi Harian dari *database* |
|---|---|
| Aktor | Admin |
| Kondisi awal | berada di halaman laporan |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka menu laporan transaksi | 2. Menampilkan halaman laporan transaksi harian |
| 3. Memilih harian | |
| 4. Mengklik *search* | 5. Melakukan proses pencarian berdasarkan hari |
| | 6. Menampilkan data yang dicari dari *database* |
| 7. Mengklik cetak laporan | 8. Menampilkan laporan transaksi harian |
| Skenario Alternatif | |
| | 6b. Data tidak ditemukan, sistem menampilkan laporan kosong". |

### 3.4.5.4 Skenario *Use Case* pada bagian Dapur

Pada tabel 3.34 menggambarkan skenario *usecase login* dapur.

**Tabel 3.34 Skenario *use case login* dapur**

| Identifikasi | |
|---|---|
| Nama *Use Case* | *login* |
| Deskripsi | Proses otentifikasi dapur untuk masuk pada sistem |
| Aktor | Dapur |
| Kondisi awal | Tampil halaman *login* |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Memasukan *user name*, *password* dan mengklik tombol *login* | 2. Mengecek valid tidaknya data masukan |
| | 3. *Login* berhasil. Masuk ke halaman beranda aplikasi dapur. |
| Skenario Alternatif | |
| | 3b. *Login* gagal karena data tidak valid, maka sistem menampilkan pesan kesalahan. "Nama |

| | |
|---|---|
| | pengguna atau *password* salah.” |

Pada tabel 2.35 menggambarkan skenario untuk meng-*update* data stok menu yang dilakukan oleh bagian dapur.

**Tabel 3.35 Skenario *use case update* data stok menu**

| Identifikasi | |
|---|---|
| Nama *Use Case* | *Update* data stok menu |
| Deskripsi | Proses untuk meng-*update* data stok menu dalam *database* |
| Aktor | Dapur |
| Kondisi awal | Berada di halaman utama data stok |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Memilih menu data stok | 2. Menampilkan halaman data stok |
| 3. Memilih data menu | |
| 4. Memilih *update* | 5. Menampilkan *form update* data stok menu |
| 6. Meng-*update* jumlah data menu, sesuai persediaan | |
| 7. Mengklik simpan | 8. Melakukan proses validasi |
| | 9. Menyimpan data stok yang telah di-*update* dalam *database*. |
| | 10. Menampilkan pesan bahwa data stok menu berhasil di-*update* |
| Skenario Alternatif | |
| | 9b. Menampilkan pesan kesalahan bahwa data stok tidak berhasil di-*update* |

Pada tabel 3.36 menggambarkan skenario untuk meng-*update* status menu pesanan yang dilakukan oleh bagian dapur.

**Tabel 3.36 Skenario *use case update* status pesanan**

| Identifikasi | |
|---|---|
| Nama *Use Case* | *Update* status data pesanan |
| Deskripsi | Proses untuk meng-*update* status pesanan dalam *database* |
| Aktor | Dapur |
| Kondisi awal | berada di halaman *update* status pesanan |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Memilih menu data pesanan | 2. Menampilkan halaman data pesanan |
| 3. Memilih data pesanan | |
| 4. Mengklik *update* | 5. Menampilkan *form update* data pesanan, dengan jumlah total masing- masing pesanan |
| 6. Meng-*update* pesanan yang telah selesai diproses lalu dikirim | |
| 7. Mengklik kirim | 8. Melakukan proses *update* status pesanan dari tiap menu pesanan sesuai jumlah pesanan yang selesai |
| 9. Mengklik simpan | 10. Melakukan proses *update* pesanan dalam *database*, mengubah status pesanan menjadi sudah dibuat, |
| Skenario Alternatif | |
| | 10b. Jika belum selesai sesuai jumlah pesanan maka statusnya masih proses |

### 3.4.5.5 Skenario *Use Case* pada bagian Kasir

Pada tabel 3.37 menggambarkan skenario *usecase login* kasir.

**Tabel 3.37 Skenario *use case login* kasir**

| Identifikasi | |
|---|---|
| Nama *Use Case* | *login* |
| Deskripsi | Proses otentifikasi kasir untuk masuk pada sistem |
| Aktor | Kasir |
| Kondisi awal | Tampil halaman *login* |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Memasukan *user* | 2. Mengecek valid tidaknya |

| | |
|---|---|
| *name*, *password* dan mengklik tombol *login* | data masukan |
| | 3. *Login* berhasil. Masuk ke halaman beranda aplikasi kasir. |
| Skenario Alternatif | |
| | 3b. *Login* gagal karena data tidak valid, maka sistem menampilkan pesan kesalahan. "Nama pengguna atau *password* salah." |

Pada tabel 3.38 menggambarkan skenario untuk *input* pembayaran yang dilakukan oleh kasir pada *use case input* pembayaran.

**Tabel 3.38 Skenario *use case* cetak *input* pembayaran**

| Identifikasi | |
|---|---|
| Nama *Use Case* | *Input* pembayaran |
| Deskripsi | Proses untuk input pembayaran dari *database* |
| Aktor | Kasir |
| Kondisi awal | berada di halaman data tagihan |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman utama | 2. Menampilkan halaman utama |
| 3. Memilih data tagihan | 4. Menampilkan daftar tagihan |
| 5. Pilih meja | 6. Menampilkan detail pesanan, dengan jumlah pesanan |
| 7. Meng-*input* jumlah pesanan yang telah selesai dan jumlah yang harus dibayar | |
| 8. Mengklik simpan dan cetak nota | 9. Menyimpan data pembayaran dalam *database*, mengubah status pemesanan menjadi sudah bayar |
| Skenario Alternatif | |
| 7b. *Input* jumlah bayar kurang | 9b. Total bayar kurang, tampil pesan"pembayaran kurang" |

Pada tabel 3.39 menggambarkan skenario untuk cetak tagihan yang dilakukan oleh kasir pada *use case* cetak tagihan.

**Tabel 3.39 Skenario *use case* cetak nota**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Cetak Nota |
| Deskripsi | Proses untuk Cetak Nota dari *database* |
| Aktor | Kasir |
| Kondisi awal | Berada dihalaman data nota |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka halaman utama | 2. Menampilkan halaman utama |
| 3. Memilih data tagihan | 4. Menampilkan data tagihan |
| 5. Memilih id pesanan/meja | |
| 6. Mengklik bayar dengan status belum bayar | 7. Menampilkan data pesanan berdasarkan jumlah pesanan dengan |
| 8. Memasukan jumlah menu yang selesai dan jumlah bayar sesuai total bayar | 9. Mengubah data status “belum bayar” menjadi “sudah bayar” |
| | 10. Menampilkan halaman data nota dengan status sudah bayar |
| 11. Mengklik cetak | 12. Menampilkan halaman data nota yang siap dicetak |
| Skenario Alternatif | |
| 7b. *Input* jumlah bayar kurang dari total bayar | 9b. Total bayar kurang, tampil pesan”pembayaran kurang” |

Pada table 3.40 menggambarkan skenario untuk cetak laporan transaksi tahunan yang dilakukan oleh kasir pada *use case* cetak laporan transaksi tahunan.

**Tabel 3.40 Skenario *use case* laporan transaksi tahunan**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Cetak Laporan Transaksi Tahunan |
| Deskripsi | Proses untuk Cetak Laporan Transaksi Tahunan dari *database* |
| Aktor | Kasir |

| Kondisi awal | berada di halaman laporan |
|---|---|
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka menu laporan transaksi | 2. Menampilkan halaman laporan transaksi tahunan |
| 3. Memilih tahun | |
| 4. Mengklik search | 5. Melakukan proses pencarian berdasarkan tahun |
| | 6. Menampilkan data yang dicari dari *database* |
| 7. Mengklik cetak laporan | 8. Menampilkan laporan transaksi tahunan |
| Skenario Alternatif | |
| | 6b. Data tidak ditemukan, sistem menampilkan laporan kosong". |

Pada table 3.41 menggambarkan skenario untuk cetak laporan transaksi bulanan yang dilakukan oleh kasir pada *use case* cetak laporan transaksi bulanan.

**Tabel 3.41 Skenario *use case* laporan transaksi bulanan**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Cetak Laporan Transaksi Bulanan |
| Deskripsi | Proses untuk Cetak Laporan Transaksi Bulanan dari *database* |
| Aktor | Kasir |
| Kondisi awal | berada di halaman laporan |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka menu laporan transaksi | 2. Menampilkan halaman laporan transaksi bulanan |
| 3. Memilih bulanan | |
| 4. Mengklik *search* | 5. Melakukan proses pencarian berdasarkan bulan |
| | 6. Menampilkan data yang dicari dari *database* |
| 7. Mengklik cetak laporan | 8. Menampilkan laporan transaksi bulanan |
| Skenario Alternatif | |
| | 6b. Data tidak ditemukan, sistem menampilkan laporan kosong". |

Pada tabel 3.42 menggambarkan skenario untuk cetak laporan transaksi harian yang dilakukan oleh kasir pada *use case* cetak laporan transaksi harian.

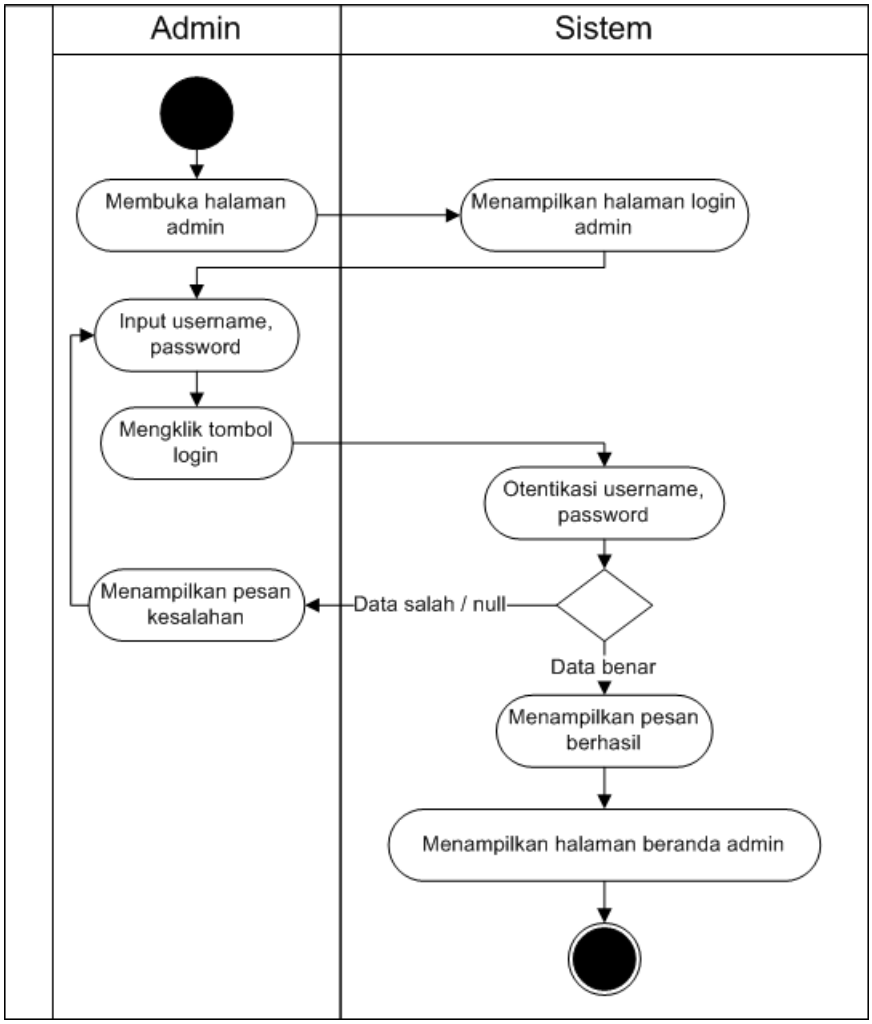
**Tabel 3.42 Skenario *use case* cetak laporan transaksi harian**

| Identifikasi | |
|---|---|
| Nama *Use Case* | Cetak Laporan Transaksi Harian |
| Deskripsi | Proses untuk Cetak Laporan Transaksi Harian dari *database* |
| Aktor | Kasir |
| Kondisi awal | berada di halaman laporan |
| Skenario Utama | |
| Aksi aktor | Respon Sistem |
| 1. Membuka menu laporan transaksi | 2. Menampilkan halaman laporan transaksi harian |
| 3. Memilih harian | |
| 4. Mengklik *search* | 5. Melakukan proses pencarian berdasarkan hari |
| | 6. Menampilkan data yang dicari dari *database* |
| 7. Mengklik cetak laporan | 8. Menampilkan laporan transaksi harian |
| Skenario Alternatif | |
| | 6b. Data tidak ditemukan, sistem menampilkan laporan kosong". |

### 3.4.6 *Activity Diagram* pada Aplikasi Admin

#### 3.4.6.1 *Login* Admin

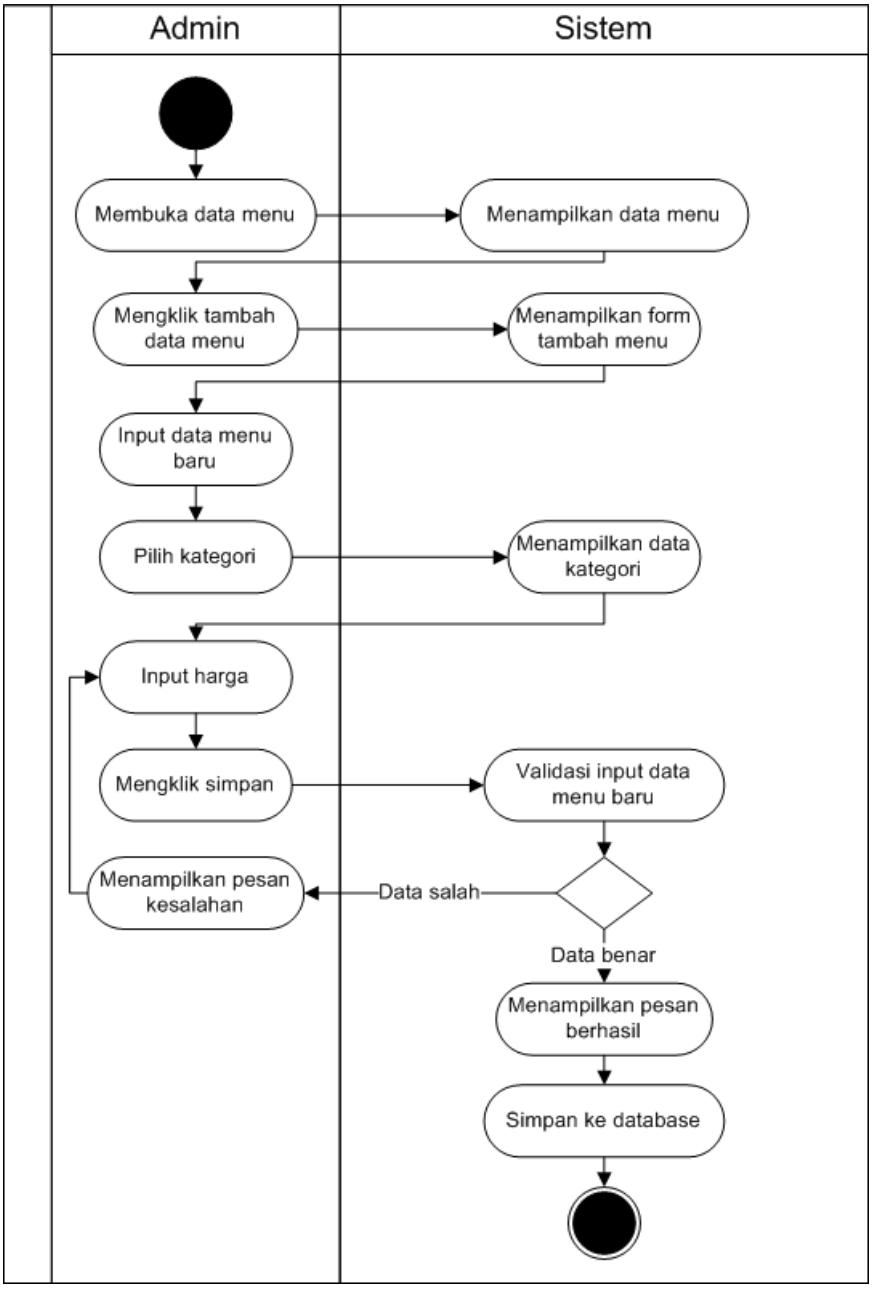
Aktivitas *login* merupakan aktivitas yang dilakukan admin untuk mendapatkan akses mengelola halaman admin. Seluruh aktivitas *login* terlihat pada *activity diagram* gambar 3.16.




**Gambar 3.16** ***Activity diagram login***

### 3.4.6.2 Tambah Data Menu

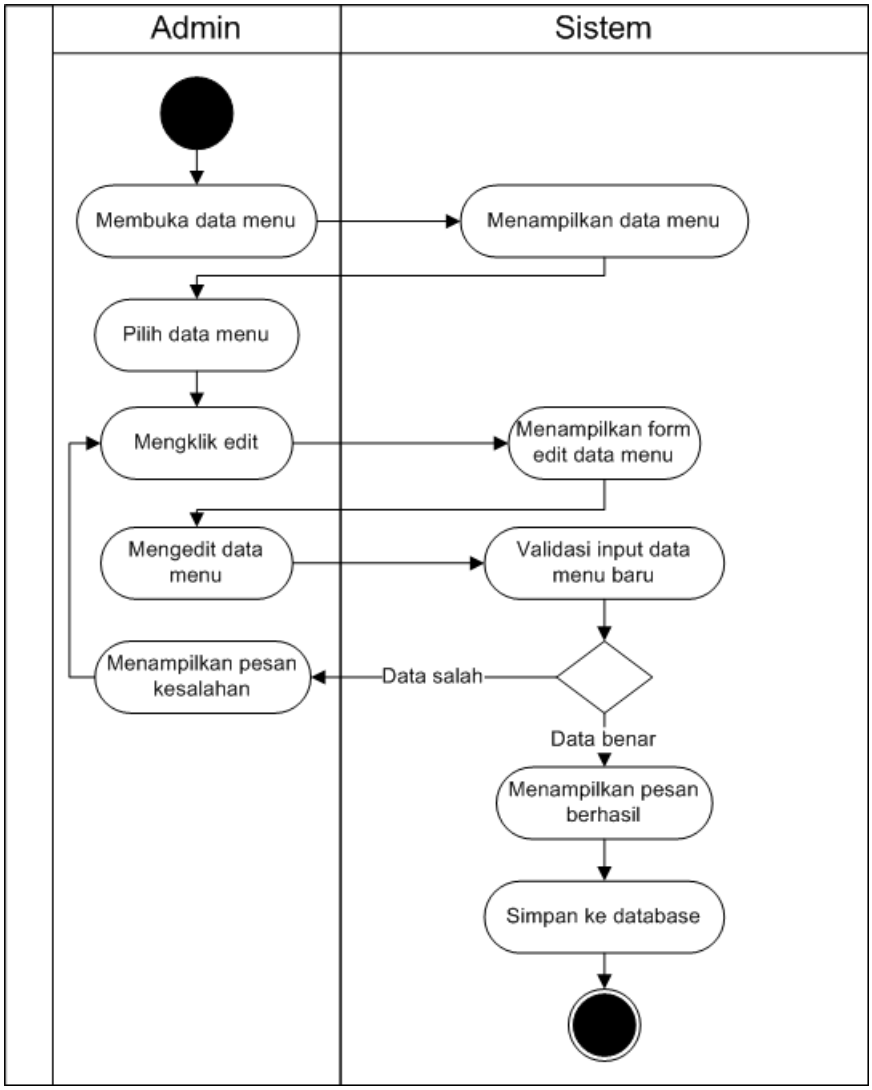
Aktivitas tambah data menu merupakan aktivitas yang dilakukan oleh admin untuk menambahkan data menu baru. Seluruh aktivitas tambah menu pesanan terlihat pada *activity diagram* gambar 3.17.




**Gambar 3.17 *Activity diagram* tambah data menu**

### 3.4.6.3 Edit Data Menu

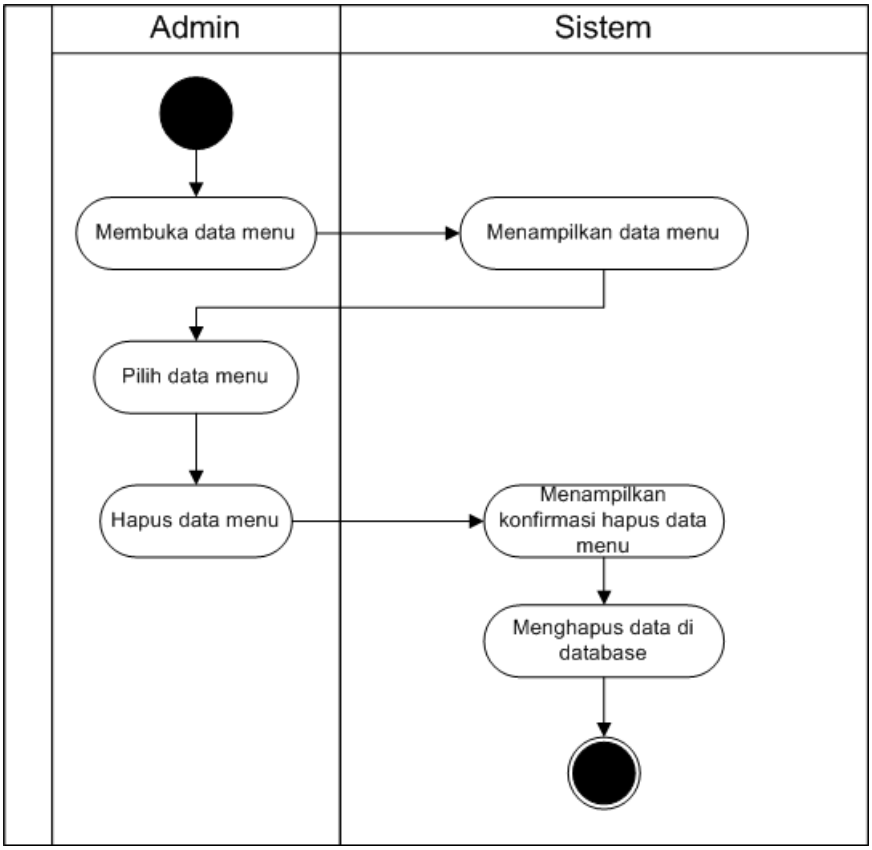
Aktivitas edit data menu merupakan aktivitas yang dilakukan oleh admin untuk mengedit data menu yang sudah ada dalam *database*. Seluruh aktivitas edit data menu terlihat pada *activity diagram* gambar 3.18.




**Gambar 3.18 *Activity diagram* edit data menu**

### 3.4.6.4 Hapus Data Menu

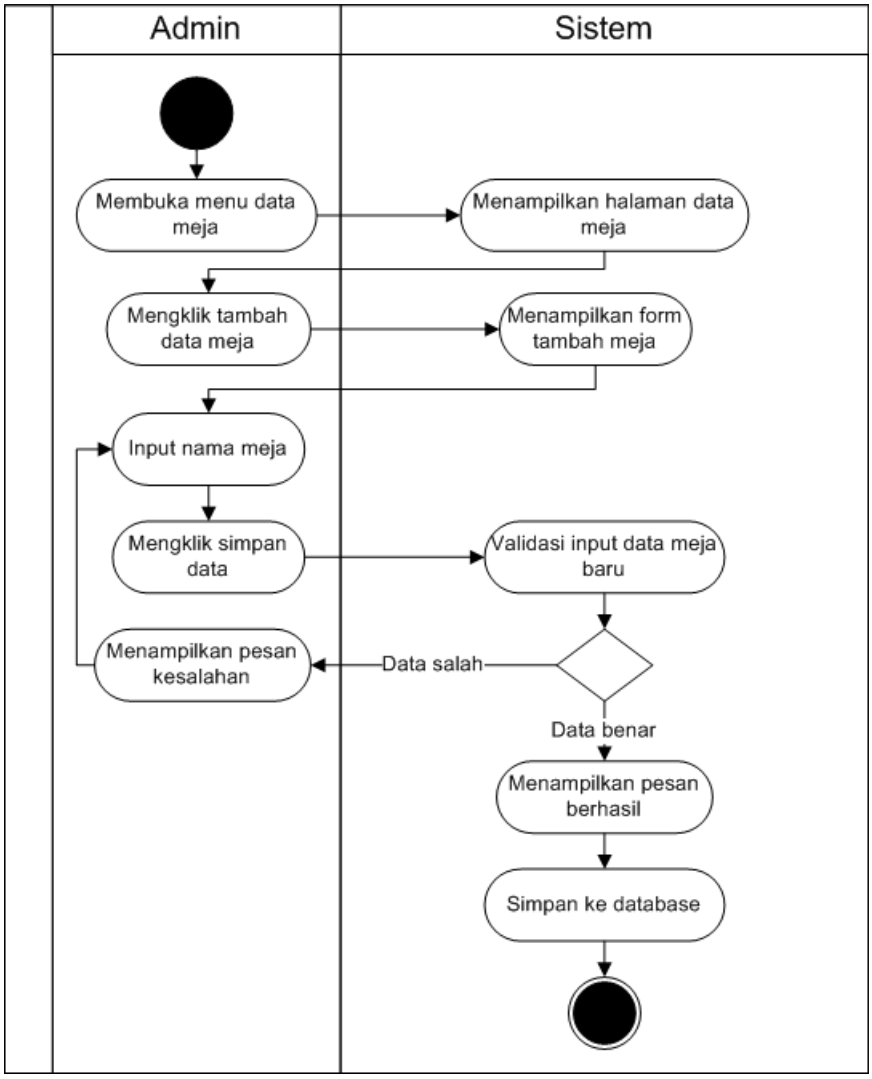
Aktivitas hapus data menu merupakan aktivitas yang dilakukan oleh admin untuk menghapus data menu yang ada dalam *database*. Seluruh aktivitas hapus data menu terlihat pada *activity diagram* gambar 3.19.




**Gambar 3.19 *Activity diagram* hapus menu pesanan**

### 3.4.6.5 Tambah Data Meja

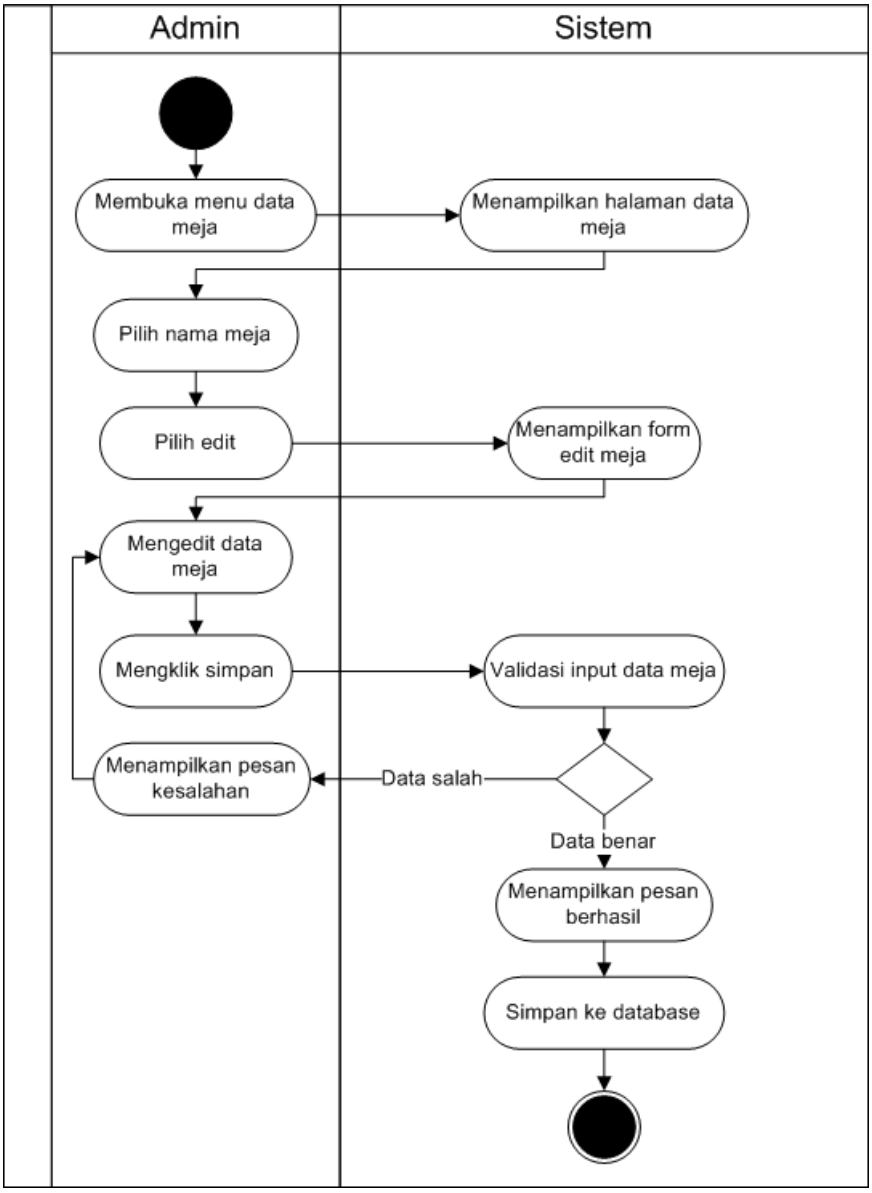
Aktivitas tambah meja merupakan aktivitas yang dilakukan oleh admin untuk menambahkan data meja baru. Seluruh aktivitas tambah meja terlihat pada *activity diagram* gambar 3.20.




**Gambar 3.20 *Activity diagram* tambah meja**

### 3.4.6.6 Edit Data Meja

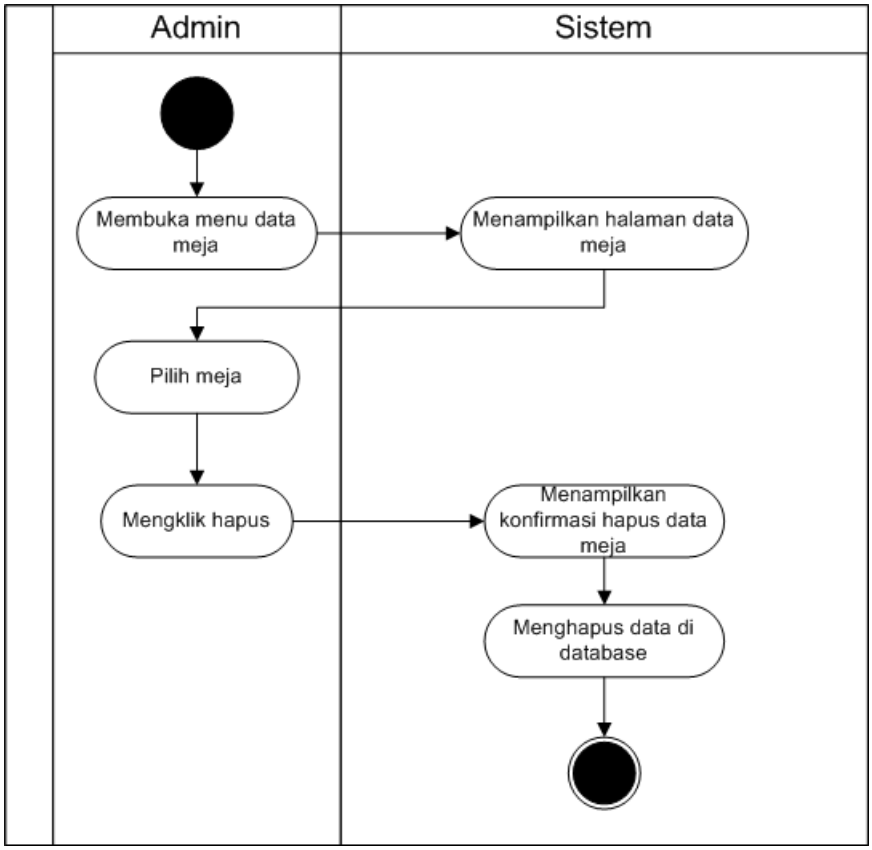
Aktivitas edit meja merupakan aktivitas yang dilakukan oleh admin untuk mengedit data meja yang sudah ada dalam *database*. Seluruh aktivitas edit meja terlihat pada *activity diagram* gambar 3.21.




**Gambar 3.21 *Activity diagram* edit meja**

### 3.4.6.7 Hapus Data Meja

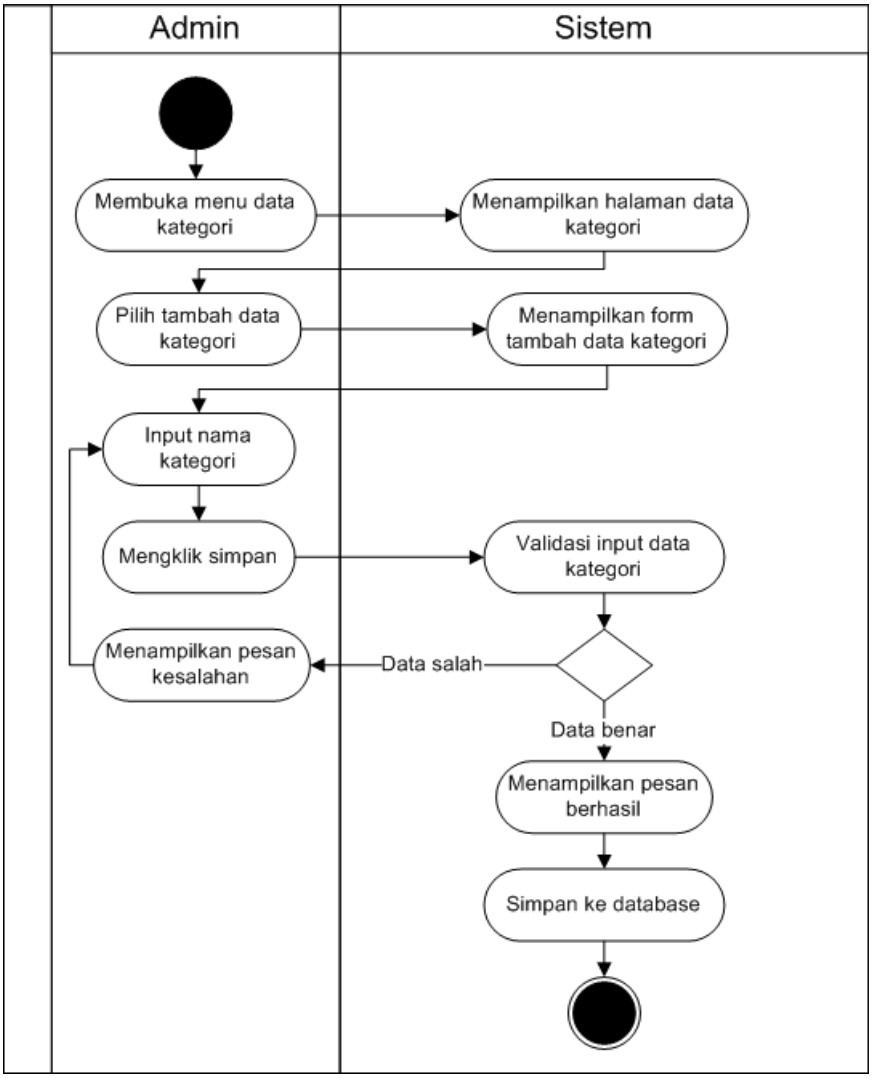
Aktivitas hapus meja merupakan aktivitas yang dilakukan oleh admin untuk menghapus meja yang ada dalam *database*. Seluruh aktivitas hapus meja terlihat pada *activity diagram* gambar 3.22.




**Gambar 3.22 *Activity diagram* hapus meja**

### 3.4.6.8 Tambah Data Kategori

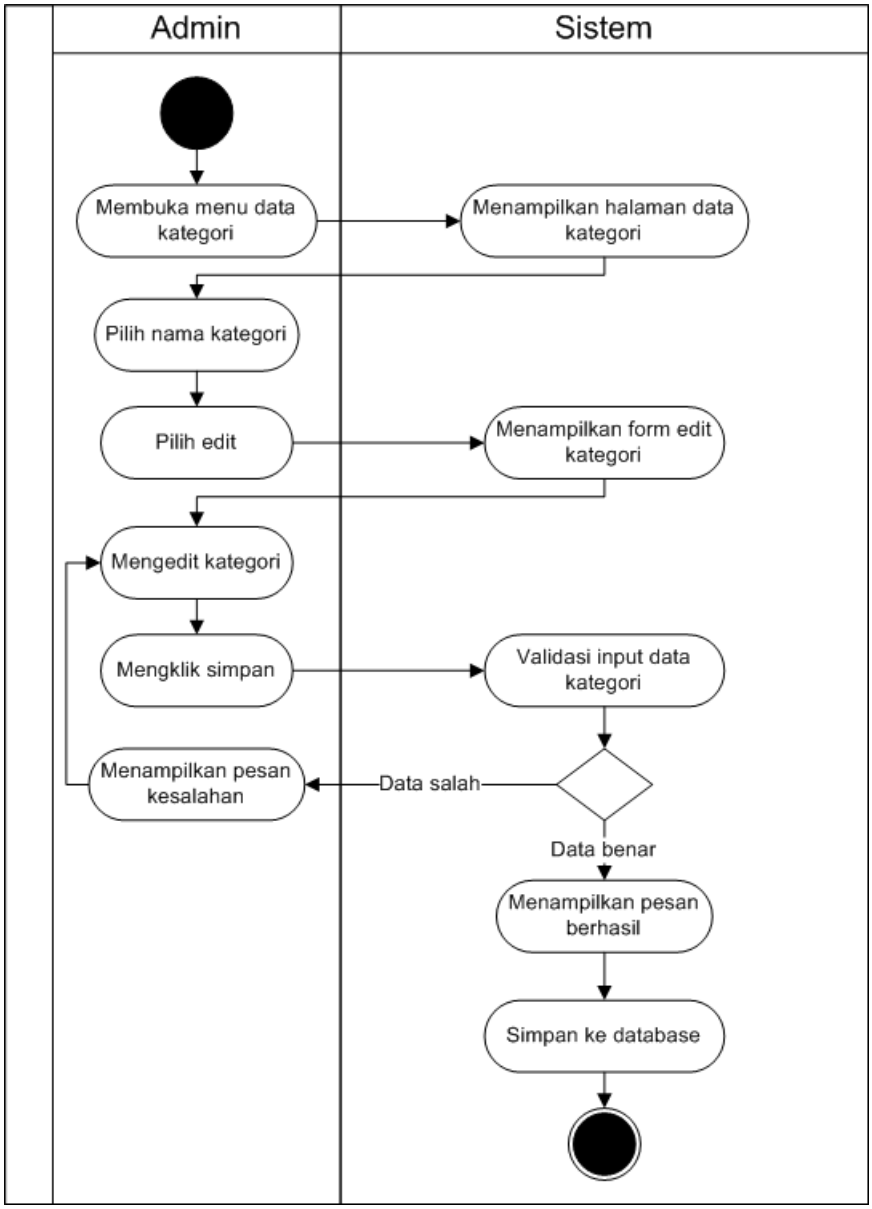
Aktivitas tambah kategori merupakan aktivitas yang dilakukan oleh admin untuk menambahkan kategori menu makanan baru. Seluruh aktivitas tambah kategori terlihat pada *activity diagram* gambar 3.23.




**Gambar 3.23 *Activity diagram* tambah kategori**

### 3.4.6.9 Edit Data Kategori

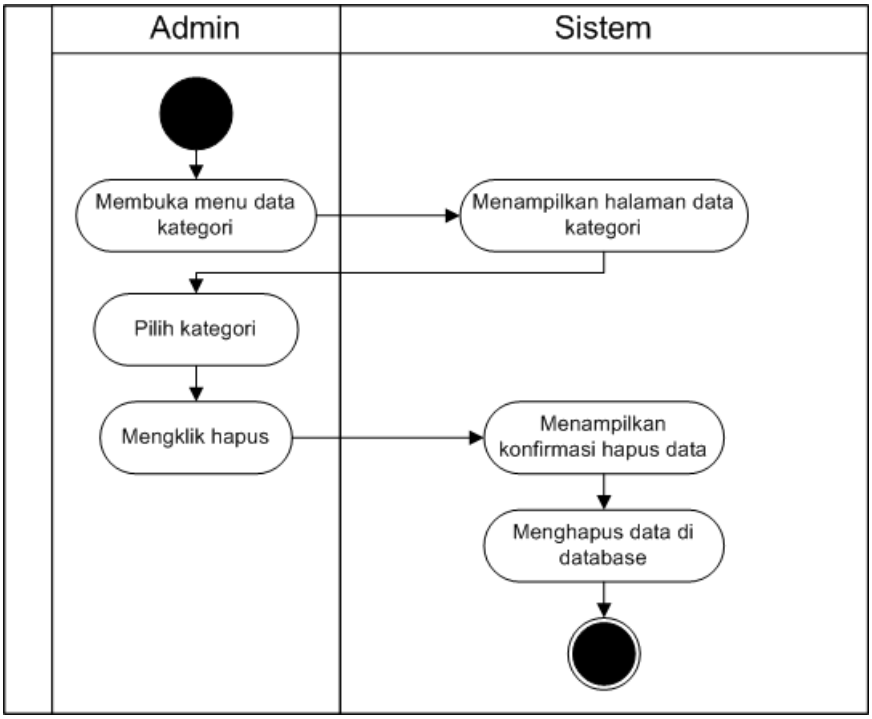
Aktivitas edit kategori merupakan aktivitas yang dilakukan oleh admin untuk mengedit kategori yang sudah ada dalam *database*. Seluruh aktivitas edit kategori terlihat pada *activity diagram* gambar 3.24.




**Gambar 3.24 *Activity diagram* edit kategori**

### 3.4.6.10 Hapus Data Kategori

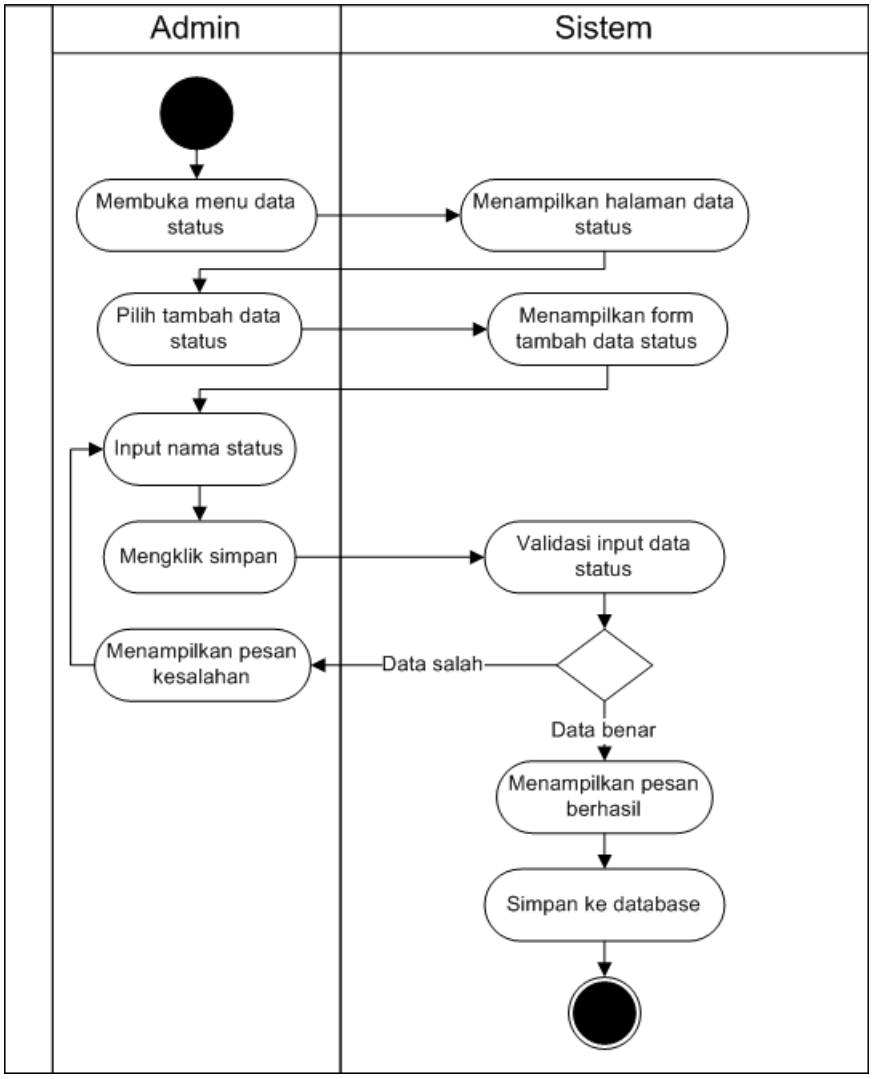
Aktivitas hapus kategori merupakan aktivitas yang dilakukan oleh admin untuk menghapus kategori yang ada dalam *database*. Seluruh aktivitas hapus kategori terlihat pada *activity diagram* gambar 3.25.




**Gambar 3.25 *Activity diagram* hapus kategori**

### 3.4.6.11 Tambah Data Status

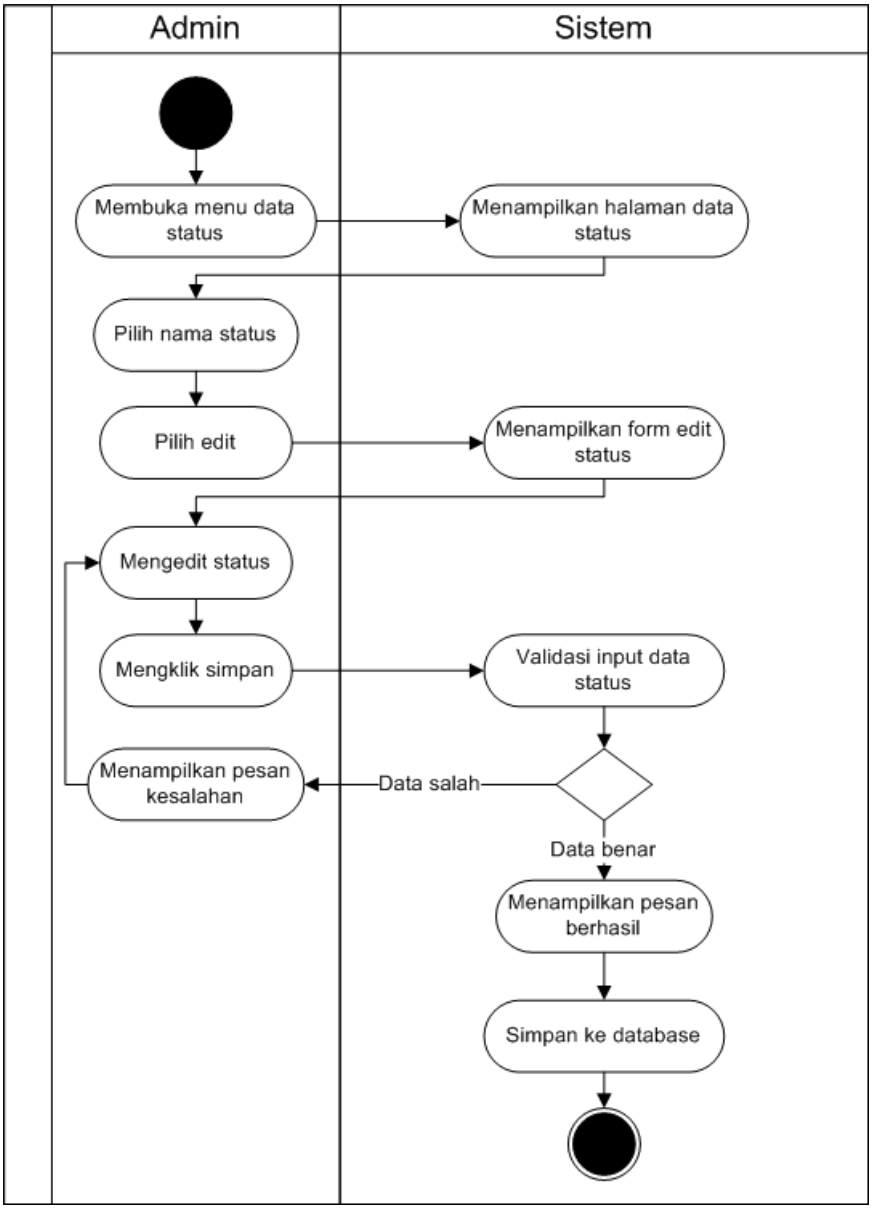
Aktivitas tambah status merupakan aktivitas yang dilakukan oleh admin untuk menambahkan status menu makanan baru. Seluruh aktivitas tambah status terlihat pada *activity diagram* gambar 3.26.




**Gambar 3.26 *Activity diagram* tambah status**

### 3.4.6.12 Edit Data Status

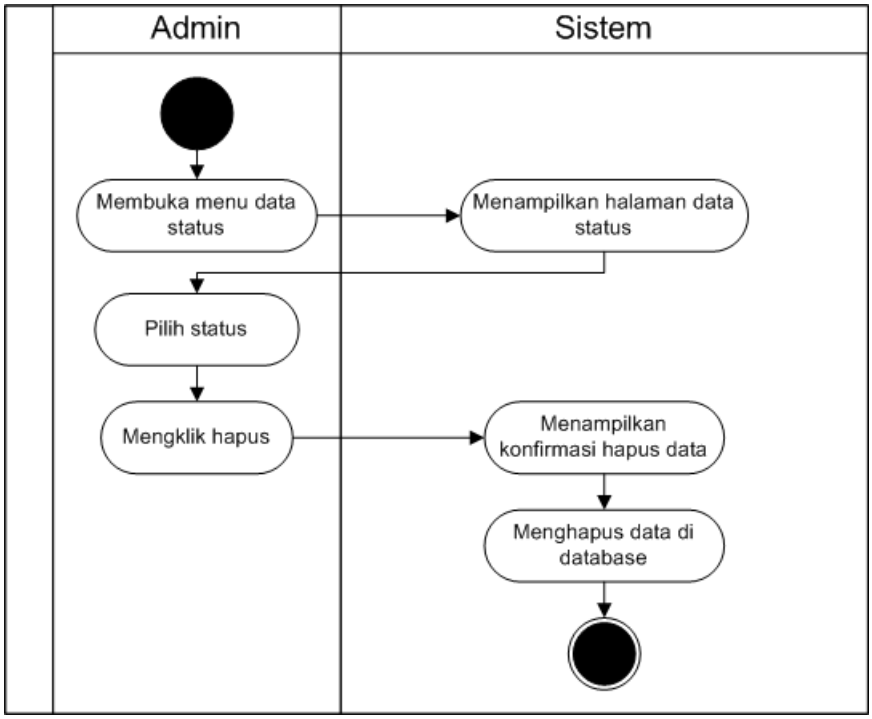
Aktivitas edit status merupakan aktivitas yang dilakukan oleh admin untuk mengedit status yang sudah ada dalam *database*. Seluruh aktivitas edit status terlihat pada *activity diagram* gambar 3.27.




**Gambar 3.27 *Activity diagram* edit status**

### 3.4.6.13 Hapus Data Status

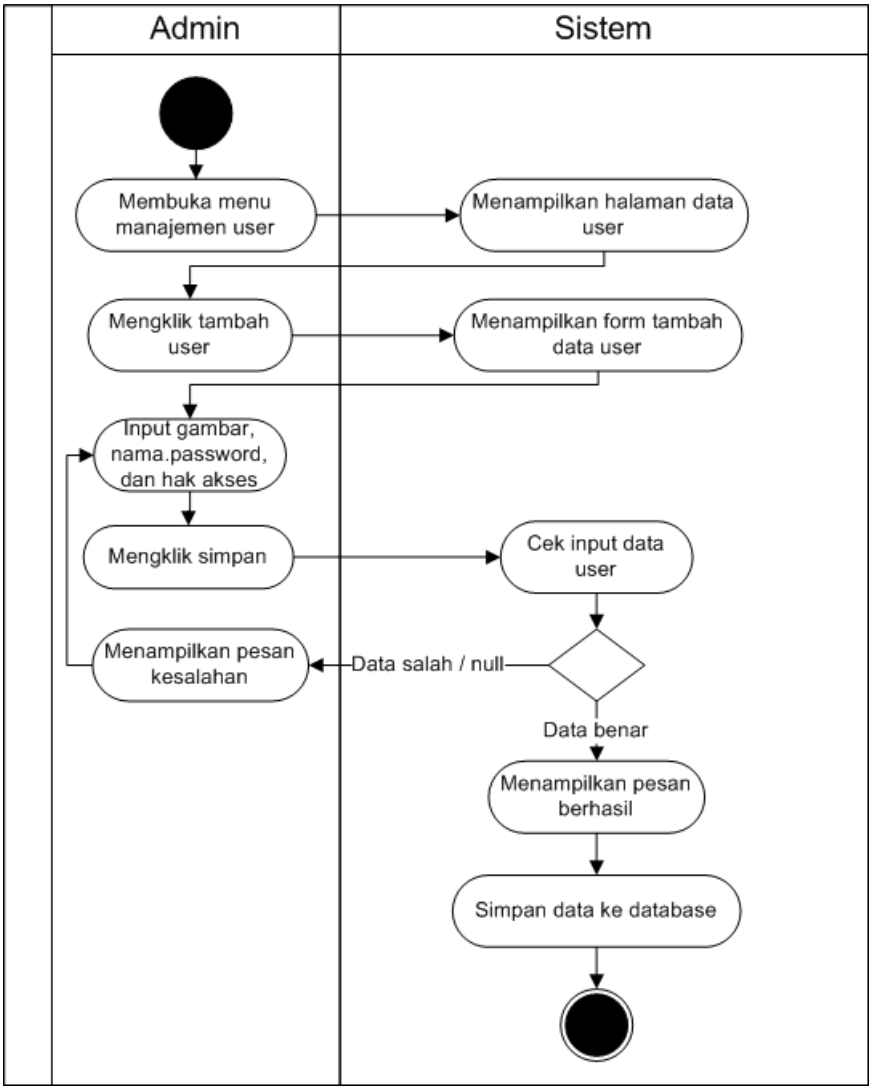
Aktivitas hapus status merupakan aktivitas yang dilakukan oleh admin untuk menghapus status yang ada dalam *database*. Seluruh aktivitas hapus status terlihat pada *activity diagram* gambar 3.28.




**Gambar 3.28 *Activity diagram* hapus status**

### 3.4.6.14 Tambah Data *User*

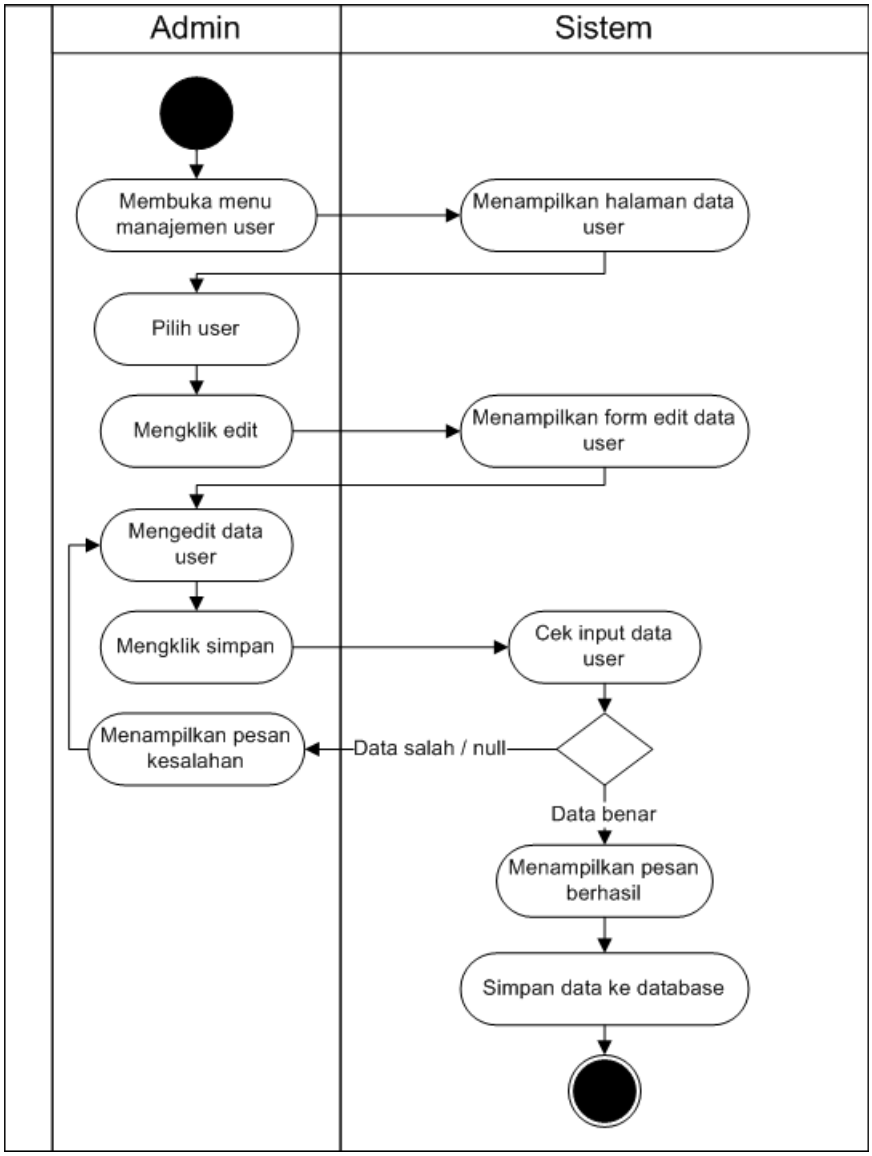
Aktivitas tambah data *user* merupakan aktivitas yang dilakukan oleh admin untuk menambahkan data *user* sebagai pengguna aplikasi ini. Seluruh aktivitas tambah kategori terlihat pada *activity diagram* gambar 3.29.




**Gambar 3.29 *Activity diagram* tambah data *user***

### 3.4.6.15 Edit Data *User*

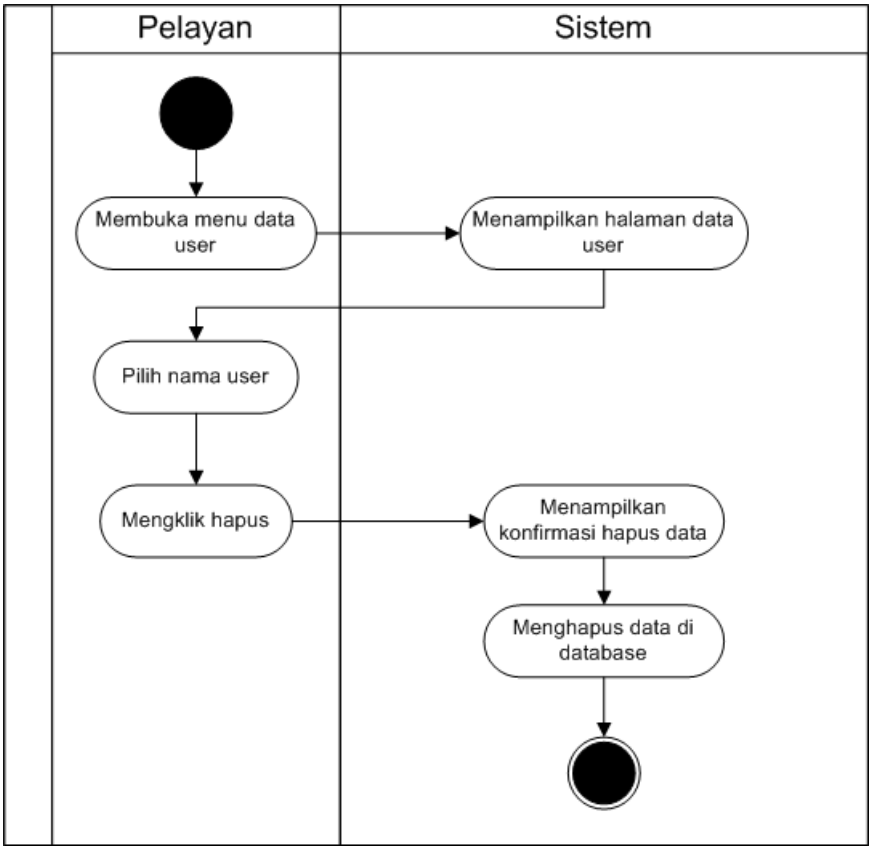
Aktivitas edit *user* merupakan aktivitas yang dilakukan oleh admin untuk mengedit *user* yang sudah ada dalam *database*. Seluruh aktivitas edit *user* terlihat pada *activity diagram* gambar 3.30.




**Gambar 3.30 *Activity diagram* edit data *user***

### 3.4.6.16 Hapus Data *User*

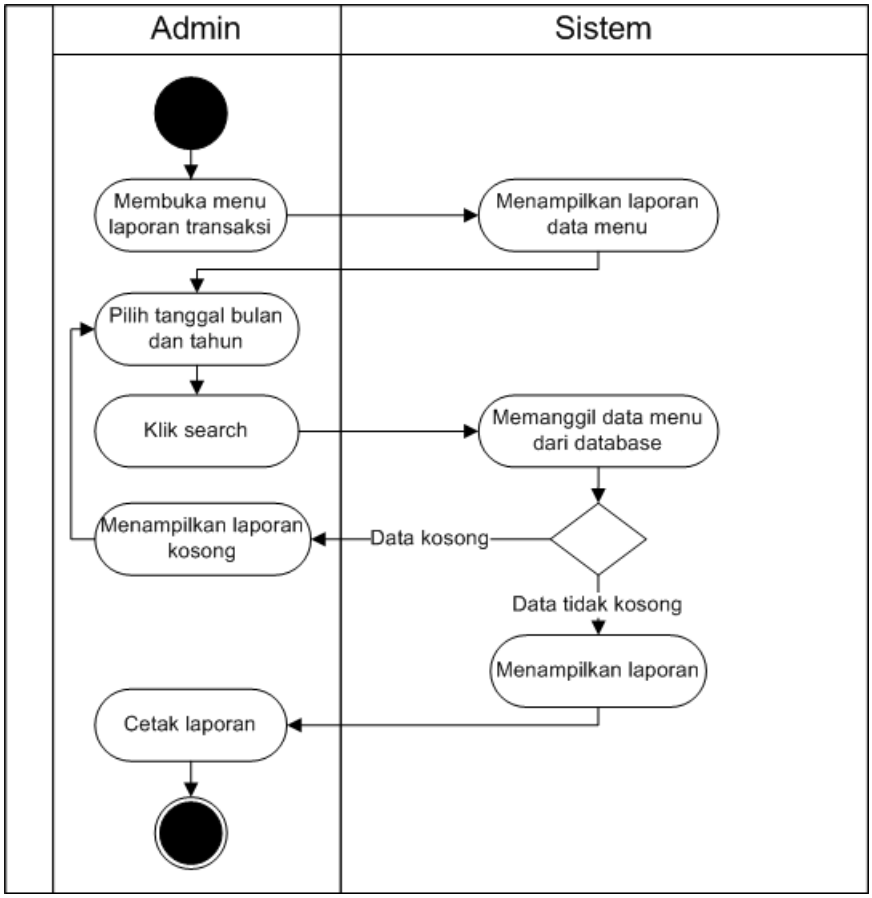
Aktivitas hapus *user* merupakan aktivitas yang dilakukan oleh admin untuk menghapus *user* yang ada dalam *database*. Seluruh aktivitas hapus *user* terlihat pada *activity diagram* gambar 3.31.




**Gambar 3.31 *Activity diagram* hapus data *user***

### 3.4.6.17 Cetak Laporan Transaksi Tahunan

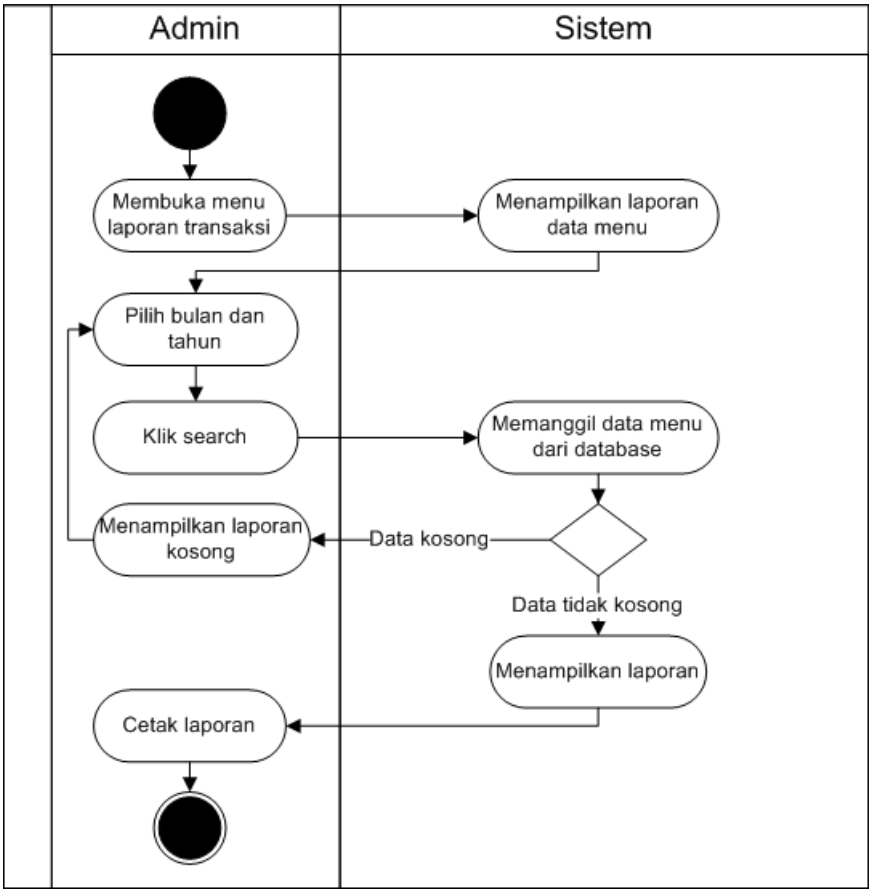
Aktivitas cetak laporan transaksi tahunan merupakan aktivitas yang dilakukan oleh admin untuk mencetak laporan transaksi tahunan yang ada dalam *database*. Seluruh aktivitas cetak laporan transaksi bulanan terlihat pada *activity diagram* gambar 3.32.




**Gambar 3.32 *Activity diagram* cetak laporan transaksi tahunan**

### 3.4.6.18 Cetak Laporan Transaksi Bulanan

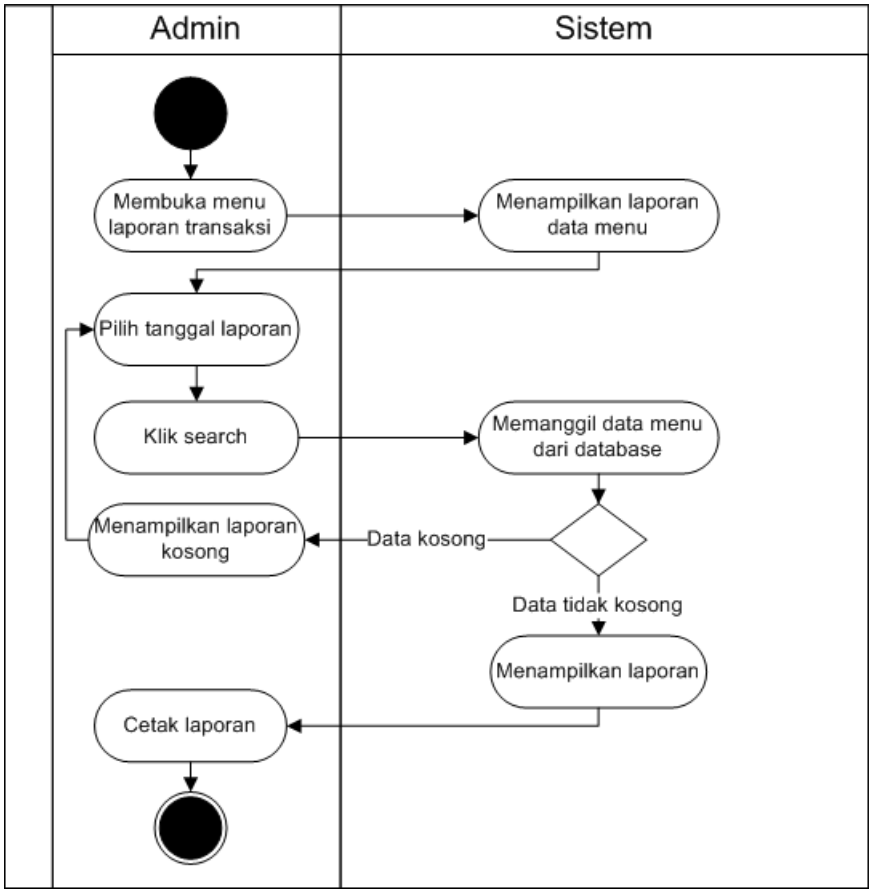
Aktivitas cetak laporan transaksi bulanan merupakan aktivitas yang dilakukan oleh admin untuk mencetak laporan transaksi bulanan yang ada dalam *database*. Seluruh aktivitas cetak laporan transaksi bulanan terlihat pada *activity diagram* gambar 3.33.




**Gambar 3.33 *Activity diagram* cetak laporan transaksi bulanan**

### 3.4.6.19 Cetak Laporan Transaksi Harian

Aktivitas cetak laporan transaksi harian merupakan aktivitas yang dilakukan oleh admin untuk mencetak laporan transaksi harian yang ada dalam *database*. Seluruh aktivitas cetak laporan transaksi harian terlihat pada *activity diagram* gambar 3.34.

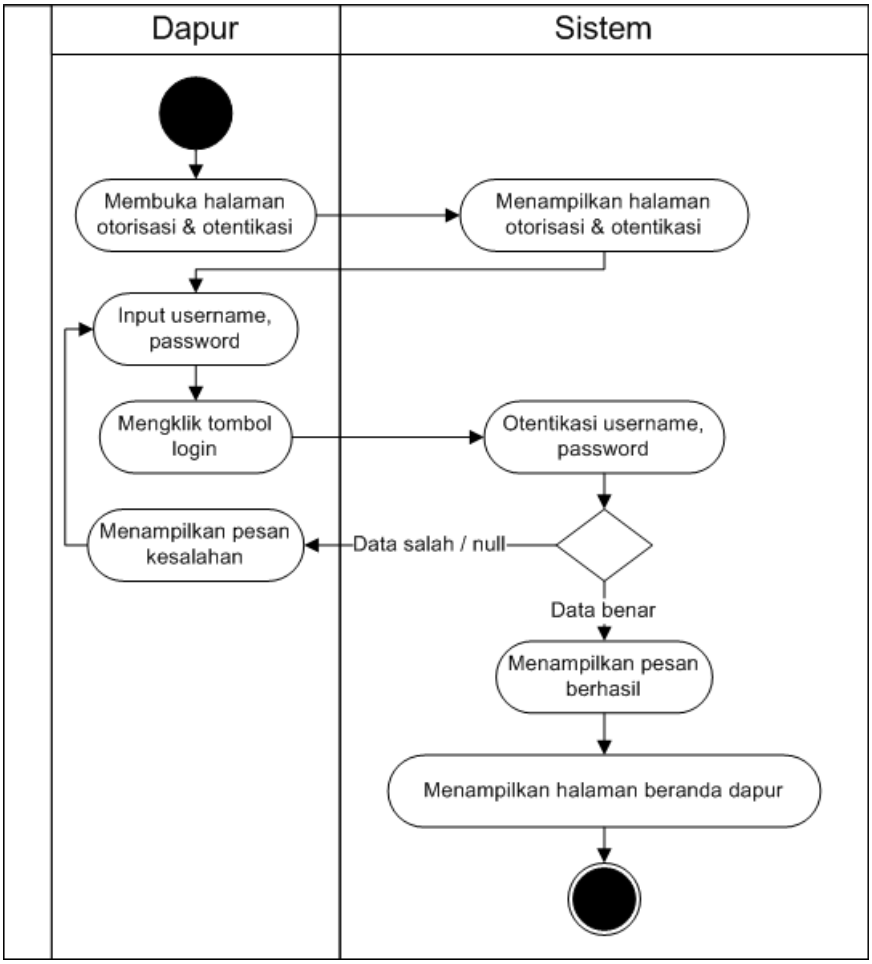



**Gambar 3.34 *Activity diagram* cetak laporan transaksi harian**

### 3.4.7 *Activity Diagram* pada Aplikasi Dapur

#### 3.4.7.1 *Login* Dapur

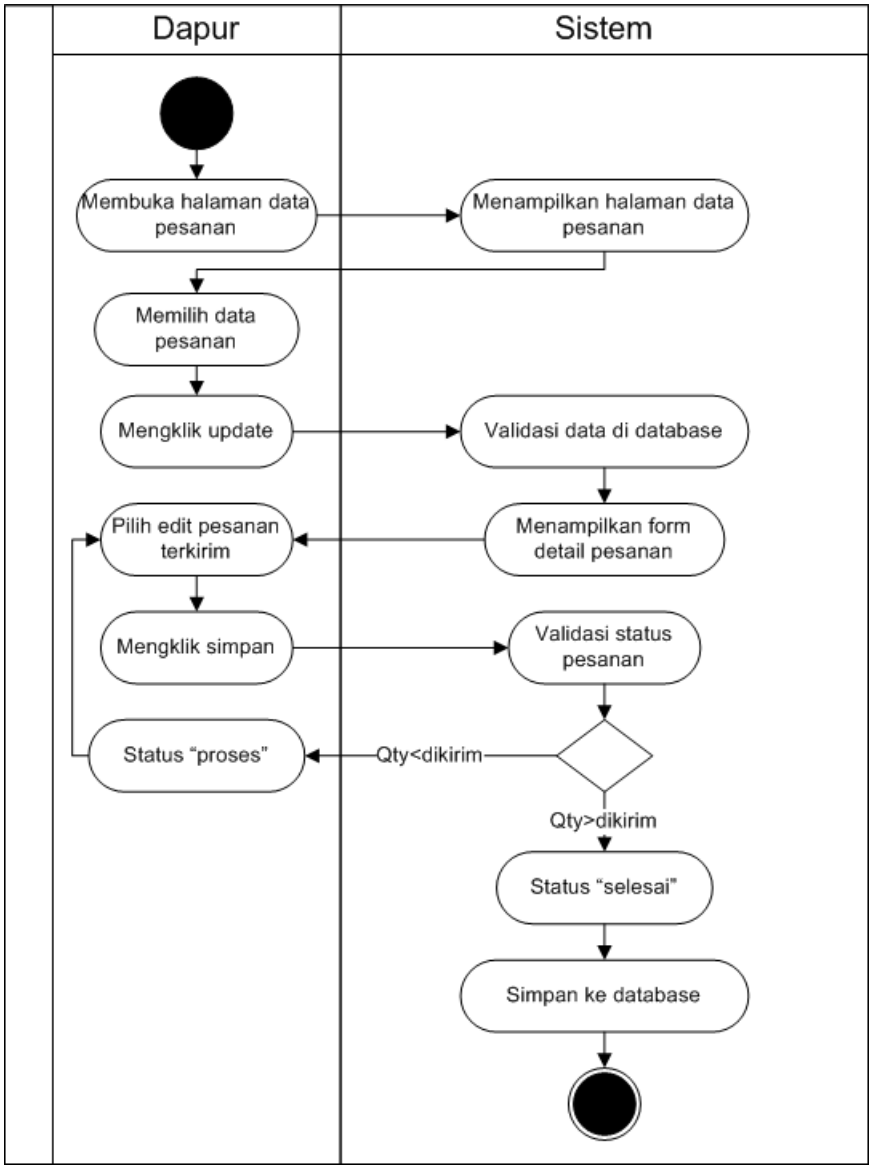
Aktivitas *login* merupakan aktivitas yang dilakukan bagian dapur untuk mandapatkan akses dalam halaman dapur. Seluruh aktivitas *login* terlihat pada *activity diagram* gambar 3.35.




**Gambar 3.35 *Activity diagram login* dapur**

### 3.4.7.2 *Update* status pesanan

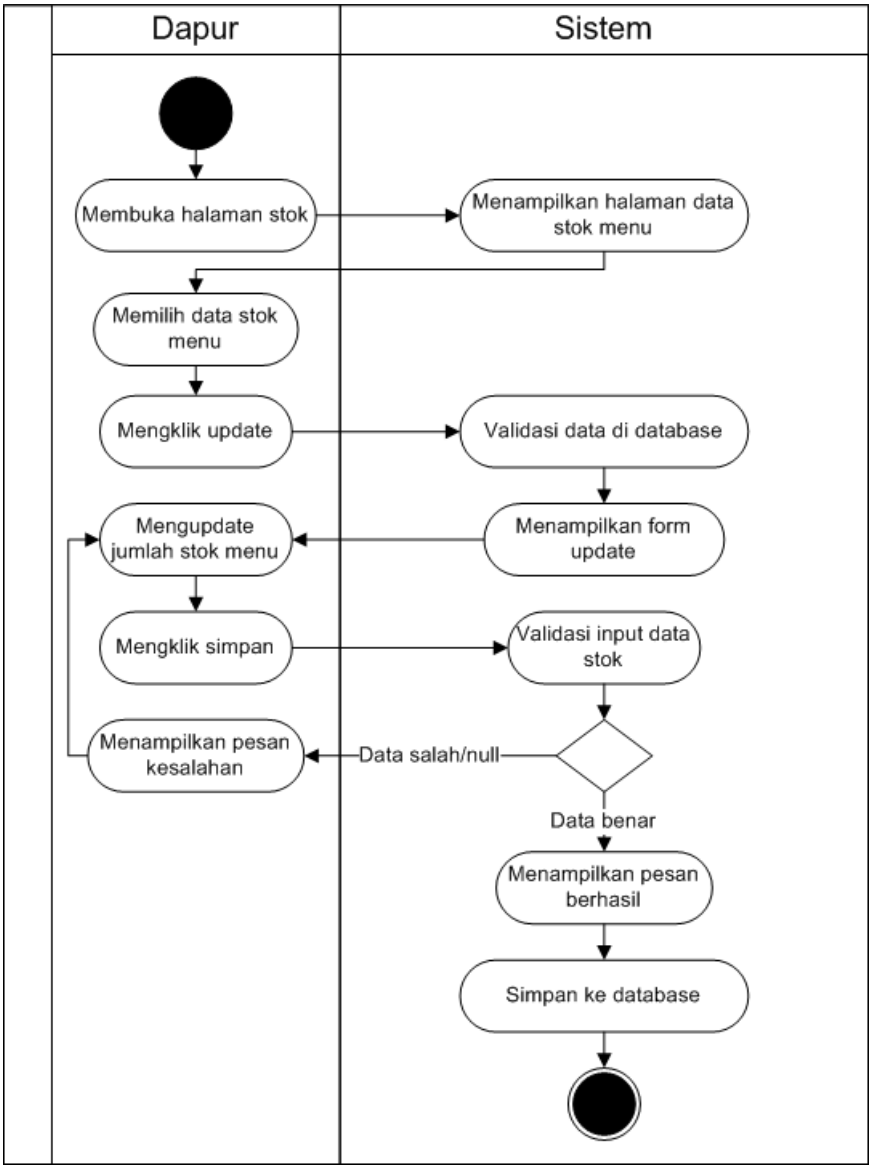
Aktivitas *update* status pesanan merupakan aktivitas yang dilakukan oleh bagian dapur untuk melakukan *update* pesanan yang dipesan oleh pelanggan. Seluruh aktivitas *update* status menu pesanan terlihat pada *activity diagram* gambar 3.36.




**Gambar 3.36 *Activity diagram update* status pesanan**

### 3.4.7.3 *Update* data stok menu

Aktivitas *update* stok data menu merupakan aktivitas yang dilakukan oleh bagian dapur untuk melakukan *update* stok data menu. Seluruh aktivitas *update* data stok menu terlihat pada *activity diagram* gambar 3.37.

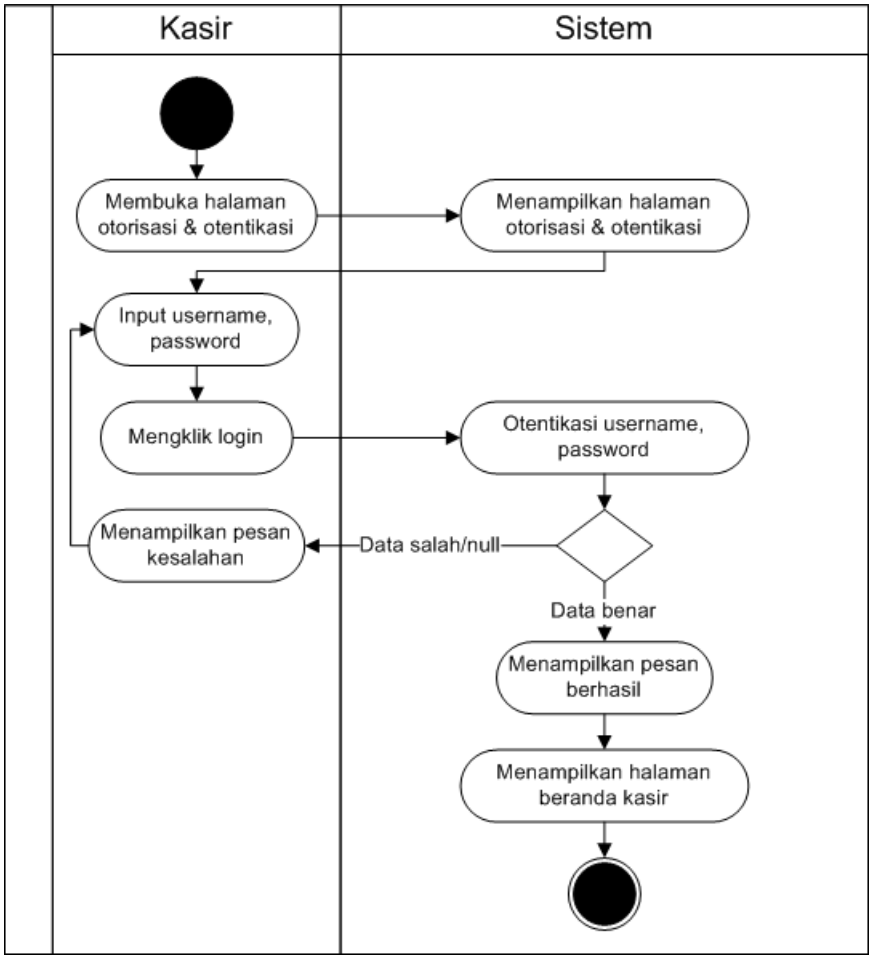



**Gambar 3.37** ***Activity diagram update*** **data stok menu**

### 3.4.8 *Activity Diagram* pada Aplikasi Kasir

#### 3.4.8.1 *Login* Kasir

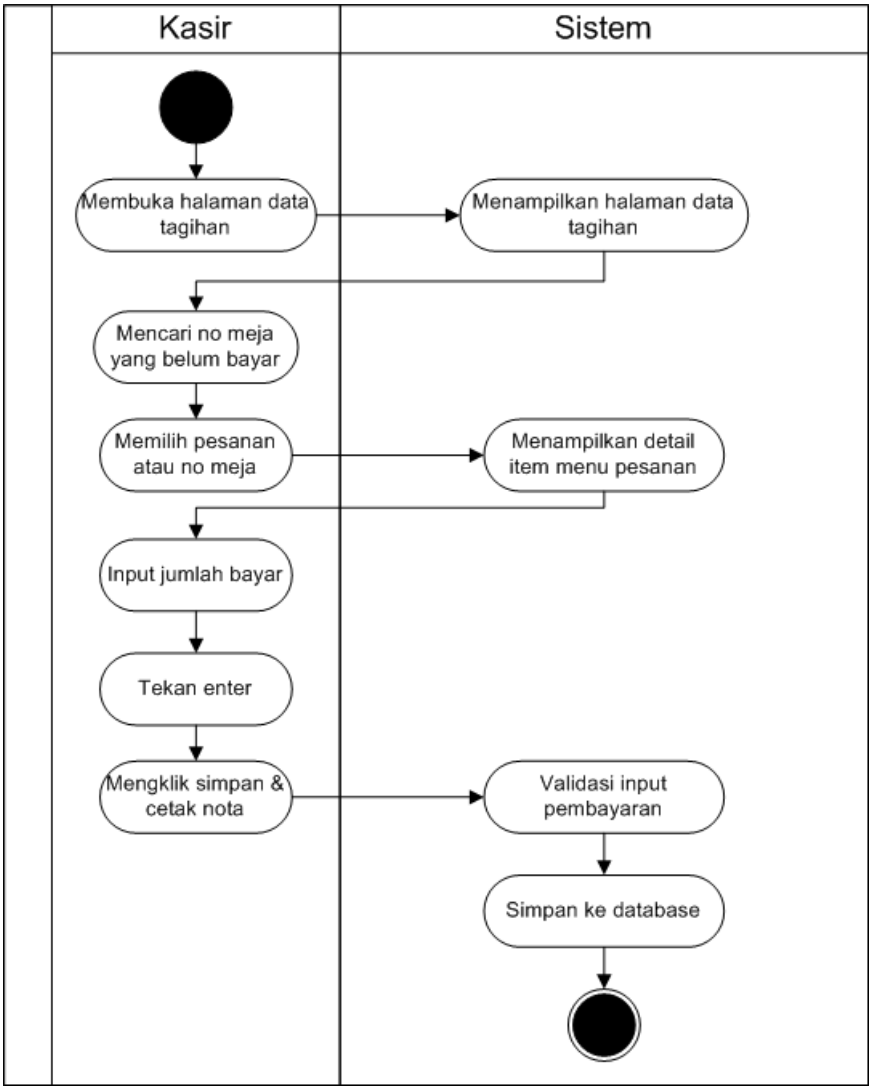
Aktivitas *login* merupakan aktivitas yang dilakukan bagian kasir untuk mandapatkan akses dalam halaman kasir. Seluruh aktivitas *login* kasir terlihat pada *activity diagram* gambar 3.38.




**Gambar 3.38 *Activity diagram login* kasir**

### 3.4.8.2 *Input* Pembayaran

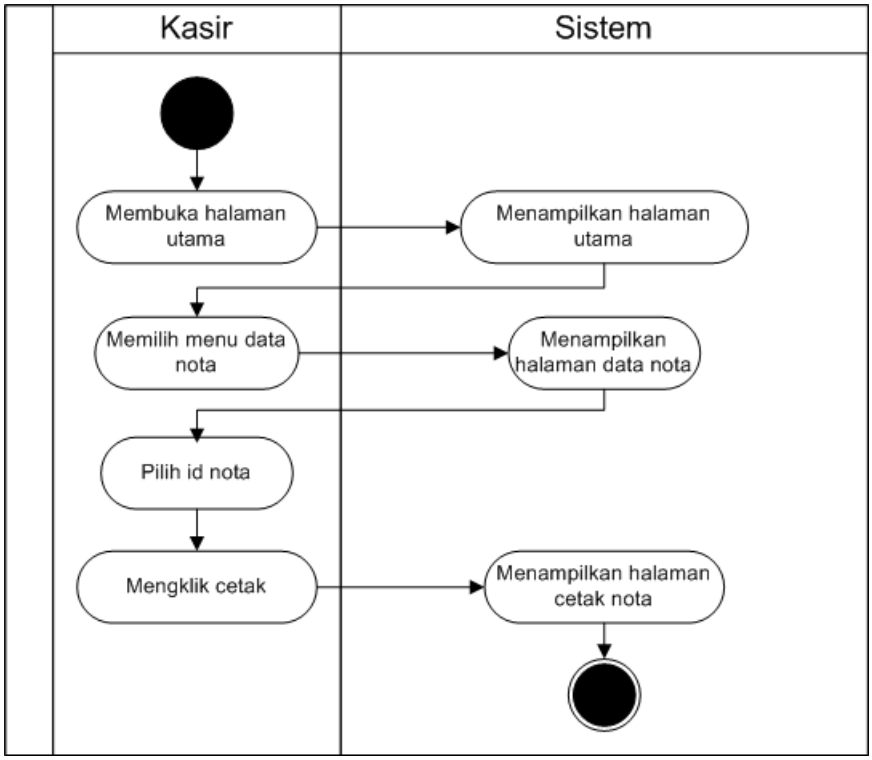
Aktivitas *input* pembayaran merupakan aktivitas yang dilakukan oleh kasir untuk melihat proses pembayaran. Seluruh aktivitas *input* pembayaran ini terlihat pada *activity diagram* gambar 3.39.




**Gambar 3.39 *Activity diagram input* pembayaran**

### 3.4.8.3 Cetak Nota

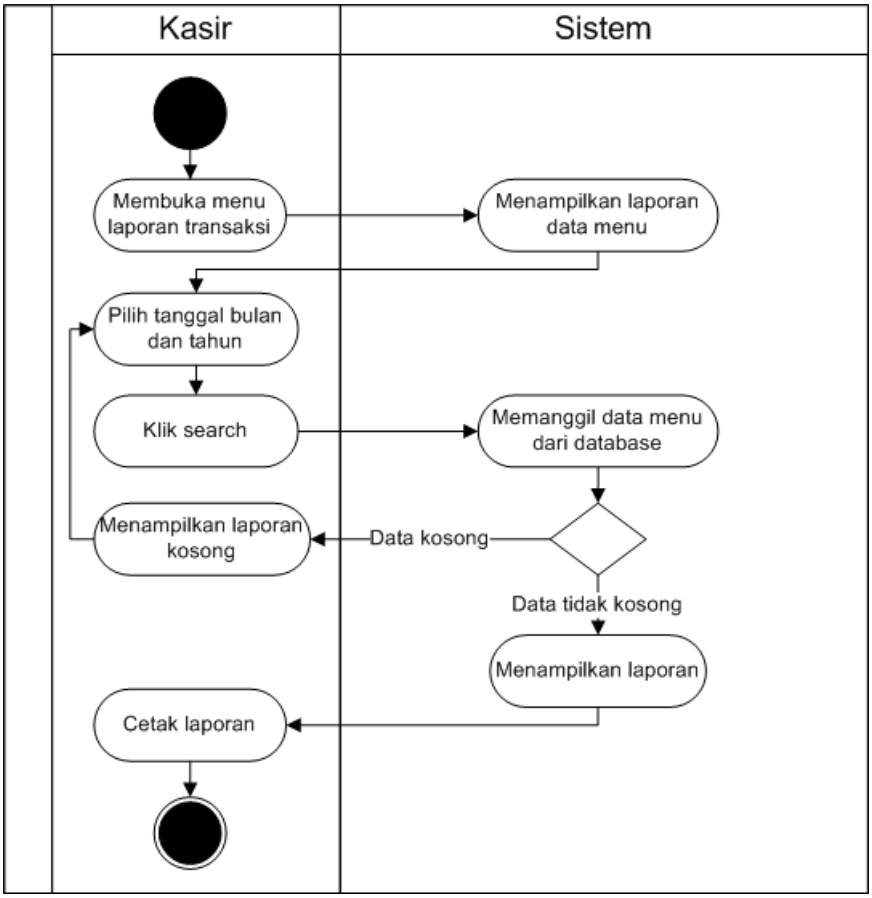
Aktivitas cetak nota merupakan aktivitas yang dilakukan oleh kasir untuk mencetak nota yang ada dalam *database*. Seluruh aktivitas cetak nota ini terlihat pada *activity diagram* gambar 3.40.




**Gambar 3.40 *Activity diagram* cetak nota**

### 3.4.8.4 Cetak Laporan Transaksi Tahunan

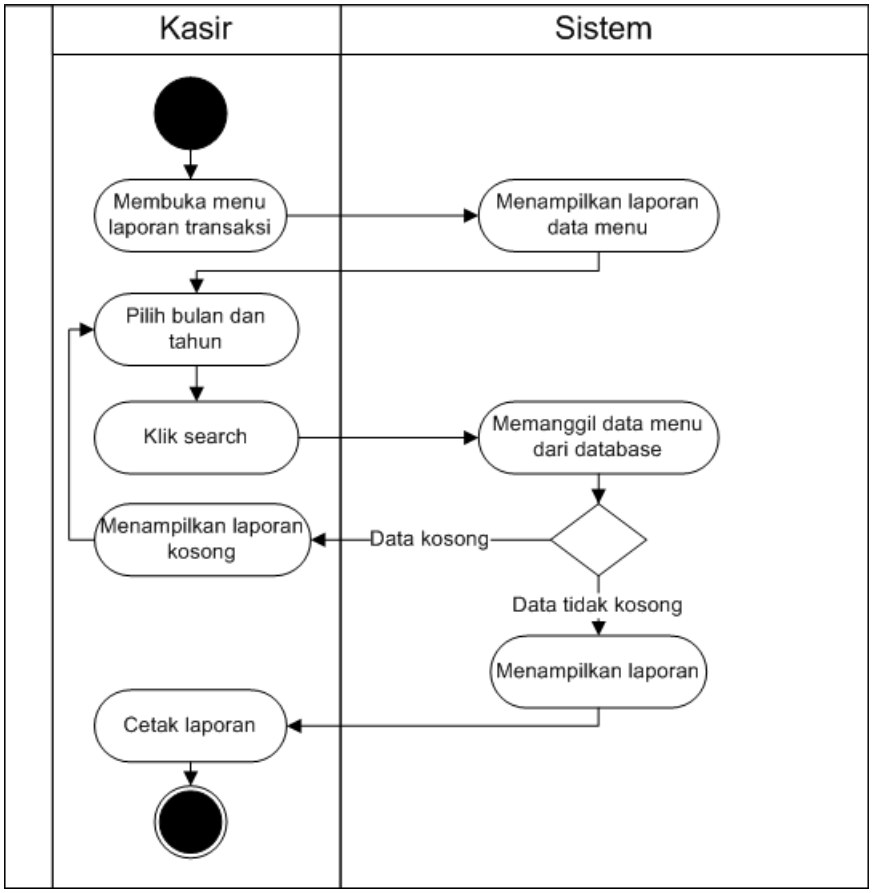
Aktivitas cetak laporan transaksi tahunan merupakan aktivitas yang dilakukan oleh kasir untuk mencetak laporan transaksi bulanan yang ada dalam *database*. Seluruh aktivitas cetak laporan transaksi tahunan terlihat pada *activity diagram* gambar 3.41.




**Gambar 3.41 *Activity diagram* cetak laporan transaksi tahunan**

### 3.4.8.5 Cetak Laporan Transaksi Bulanan

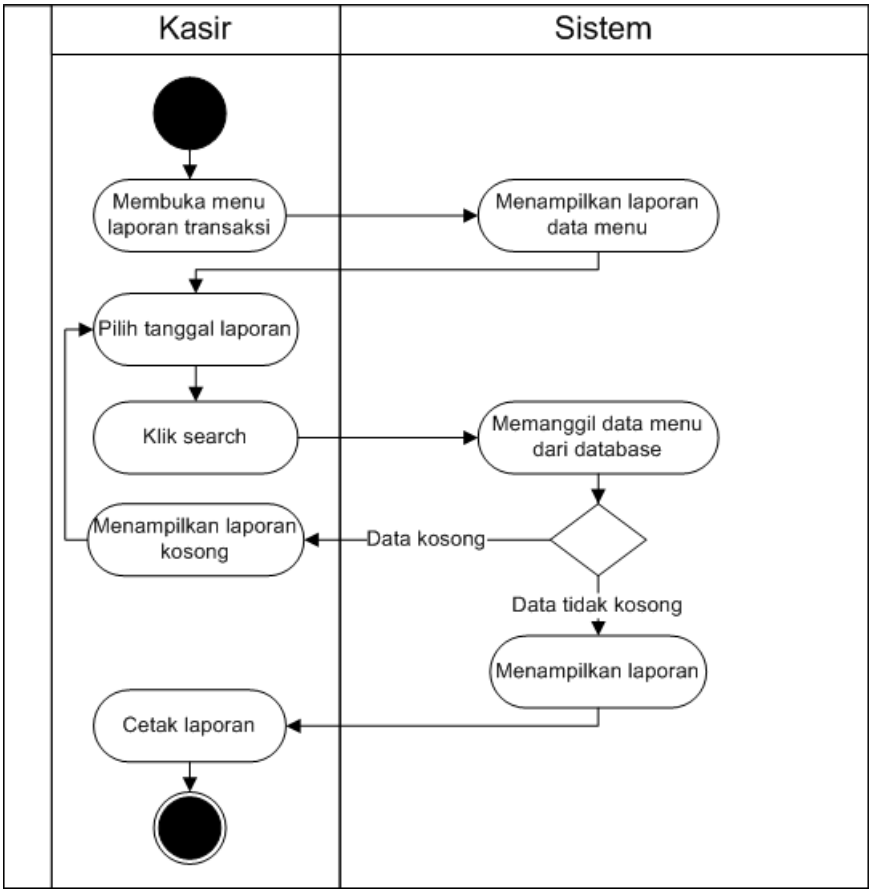
Aktivitas cetak laporan transaksi bulanan merupakan aktivitas yang dilakukan oleh kasir untuk mencetak laporan transaksi bulanan yang ada dalam *database*. Seluruh aktivitas cetak laporan transaksi bulanan terlihat pada *activity diagram* gambar 3.42.




**Gambar 3.42 *Activity diagram* cetak laporan transaksi bulanan**

### 3.4.8.6 Cetak Laporan Transaksi Harian

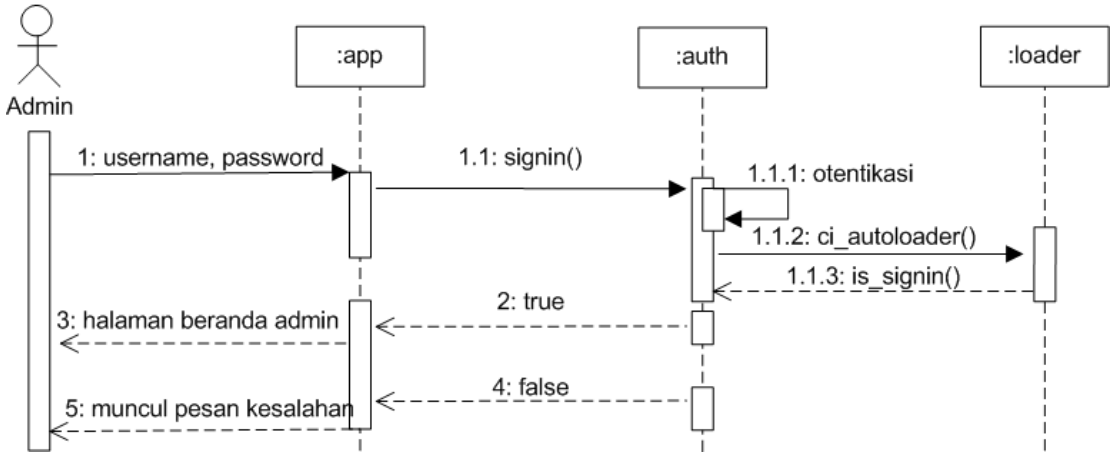
Aktivitas cetak laporan transaksi harian merupakan aktivitas yang dilakukan oleh kasir untuk mencetak laporan transaksi harian yang ada dalam *database*. Seluruh aktivitas cetak laporan transaksi harian terlihat pada *activity diagram* gambar 3.43.




**Gambar 3.43 *Activity diagram* cetak laporan transaksi harian**

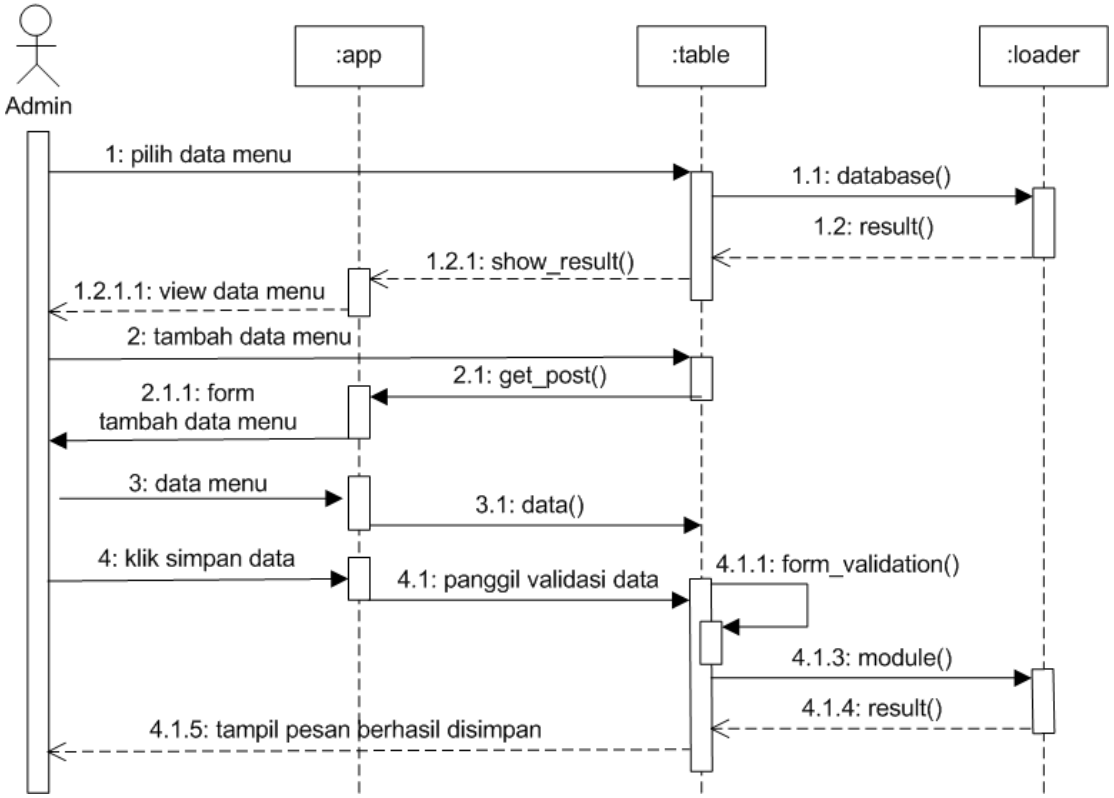
### 3.4.9 *Sequence diagram* pada Aplikasi Admin

#### 3.4.9.1 *Login* Admin



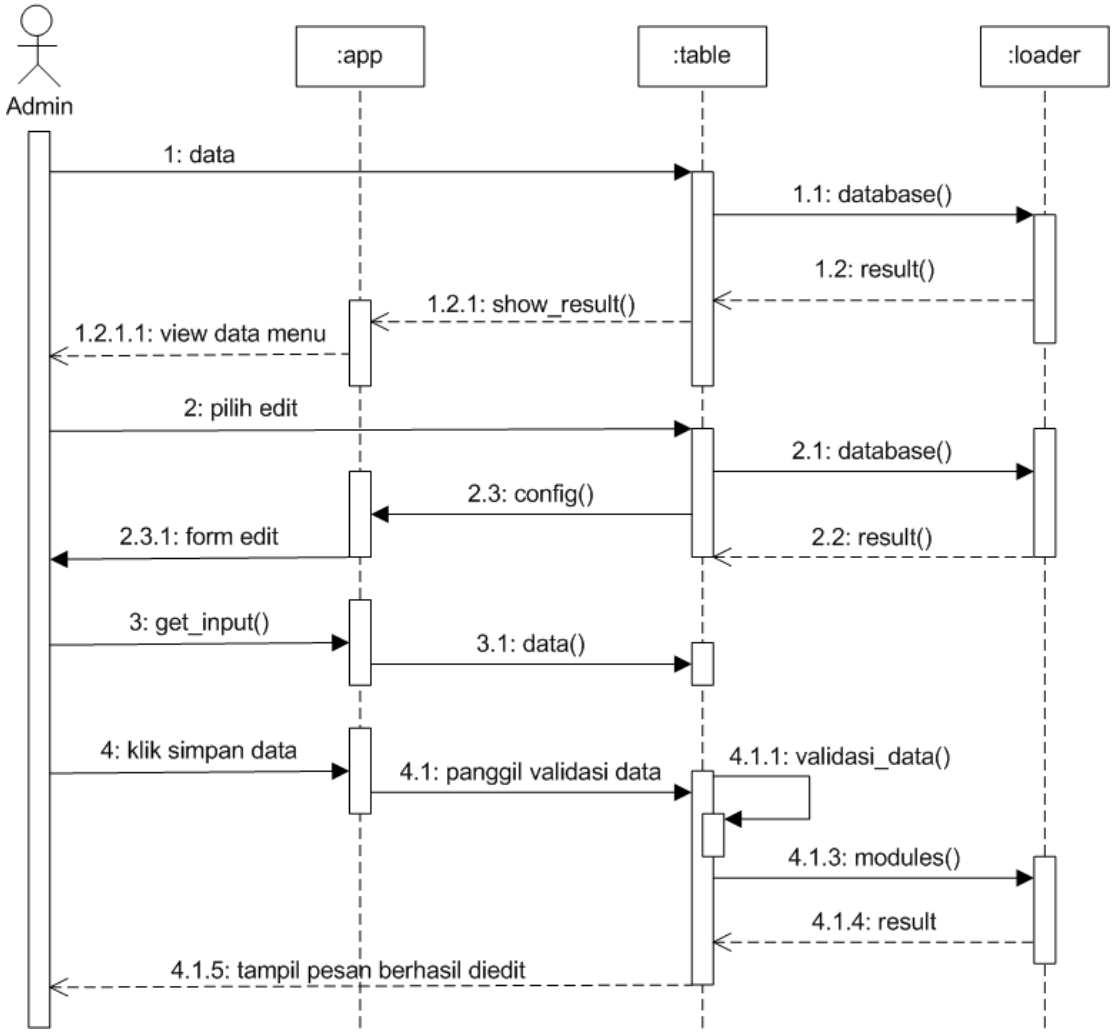

**Gambar 3.44 *Sequence diagram login* admin**

### 3.4.9.2 Tambah Data Menu



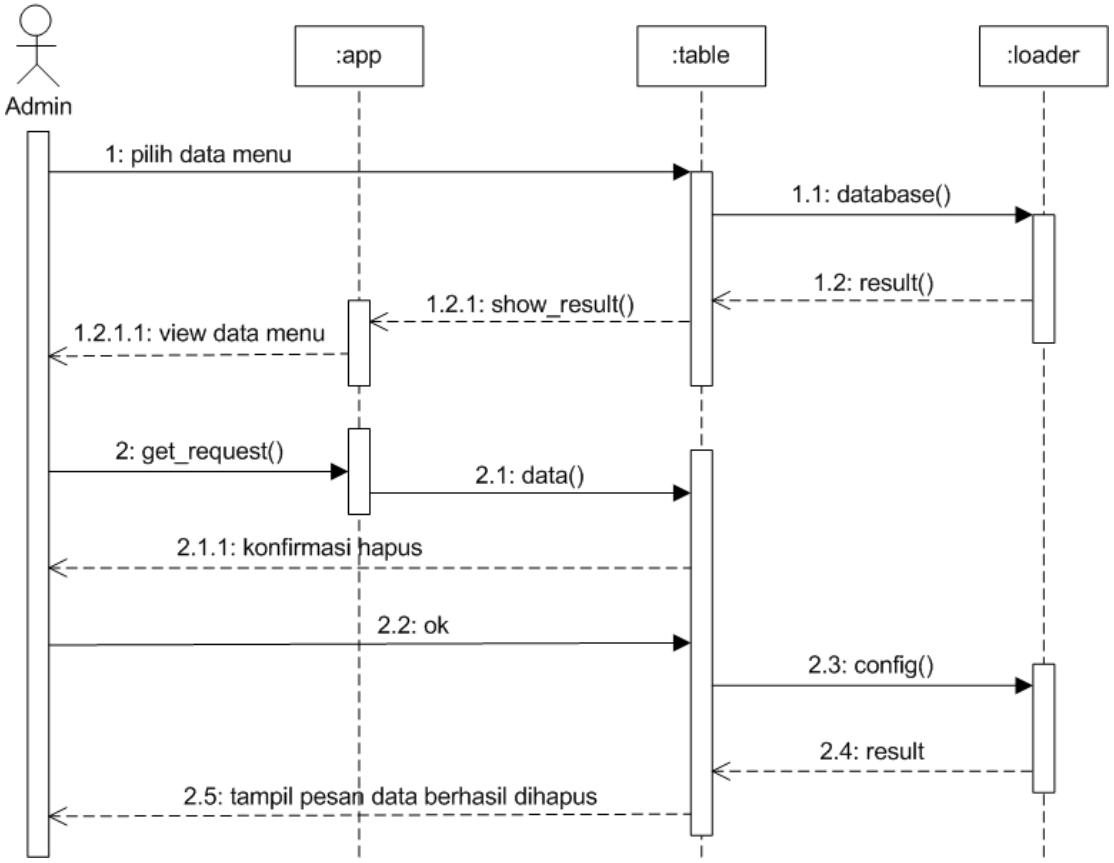

**Gambar 3.45 *Sequence diagram* tambah data menu**

### 3.4.9.3 Edit Data Menu



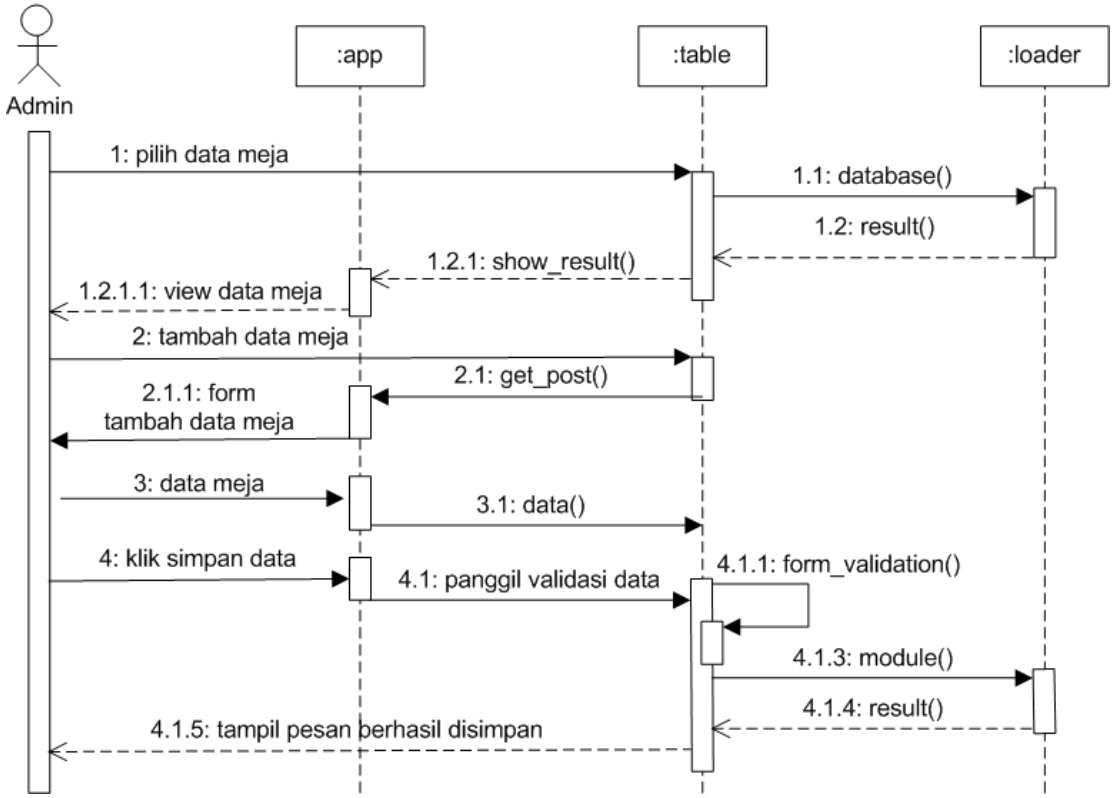

**Gambar 3.46 *Sequence diagram* edit data menu**

#### 3.4.9.4 Hapus Data Menu



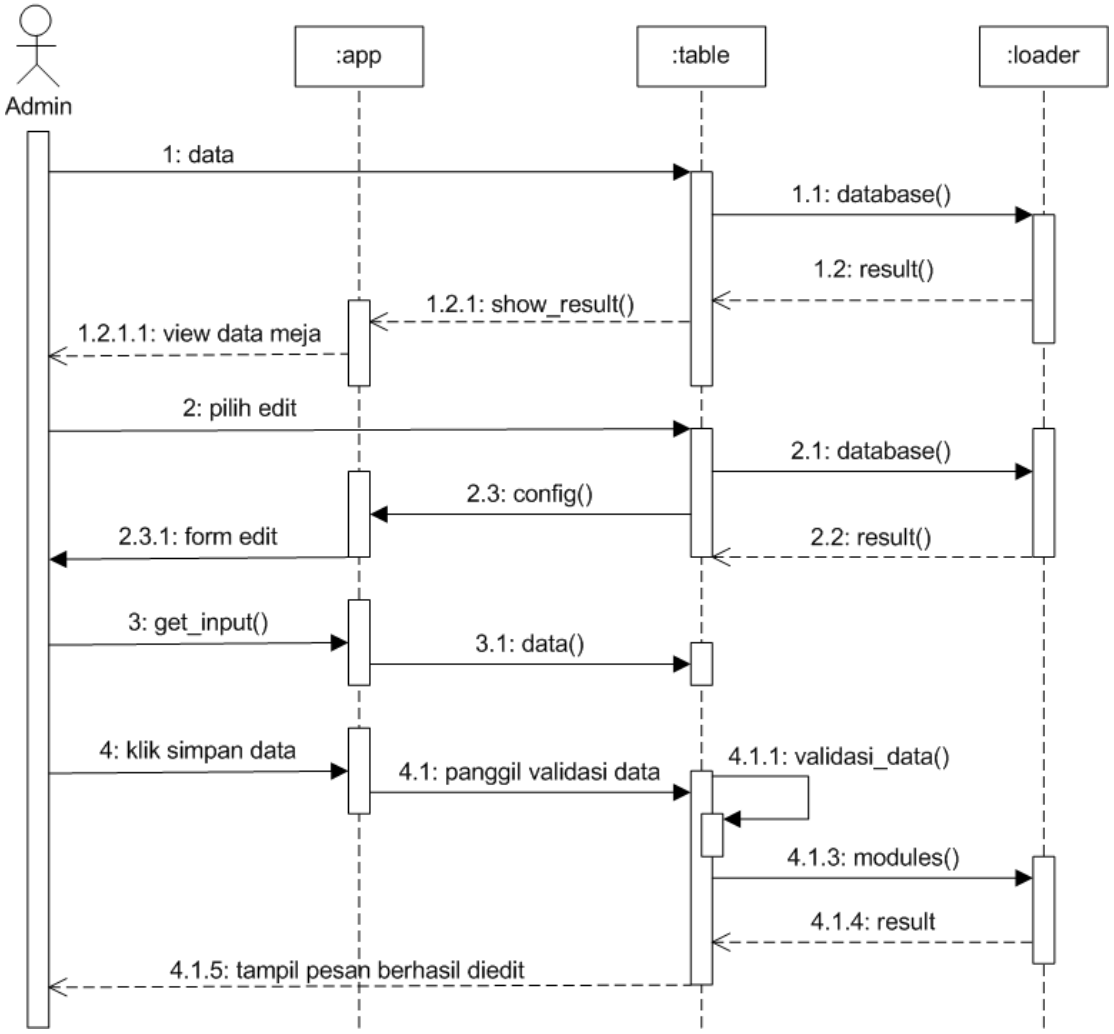

**Gambar 3.47 *Sequence diagram* hapus data menu**

### 3.4.9.5 Tambah Data Meja



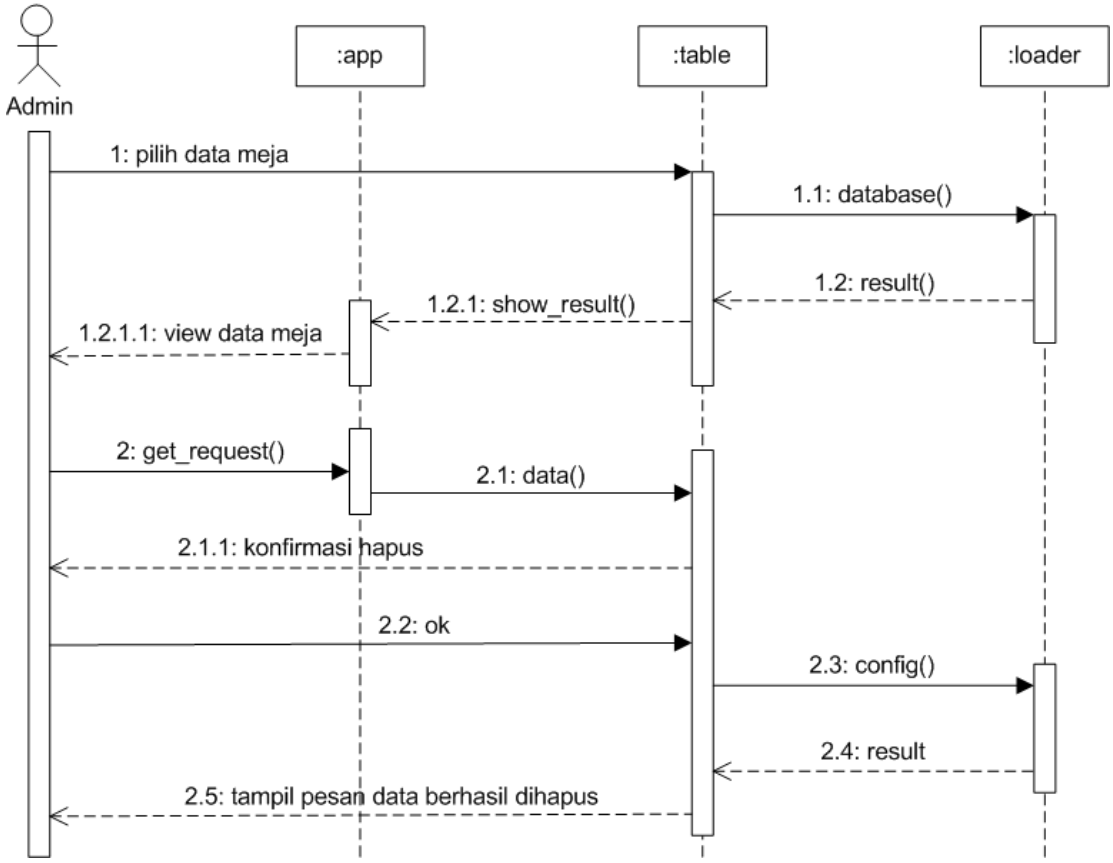

**Gambar 3.48 *Sequence diagram* tambah data meja**

### 3.4.9.6 Edit Data Meja



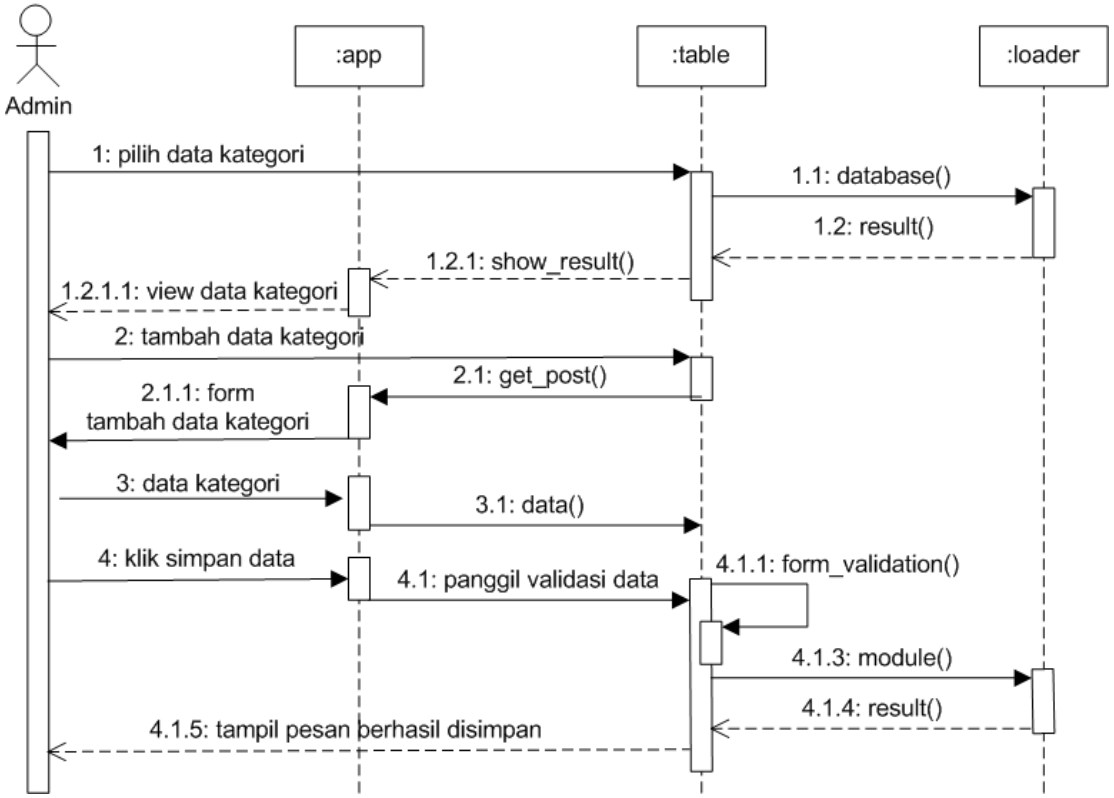

**Gambar 3.49 *Sequence diagram* edit data meja**

### 3.4.9.7 Hapus Data Meja



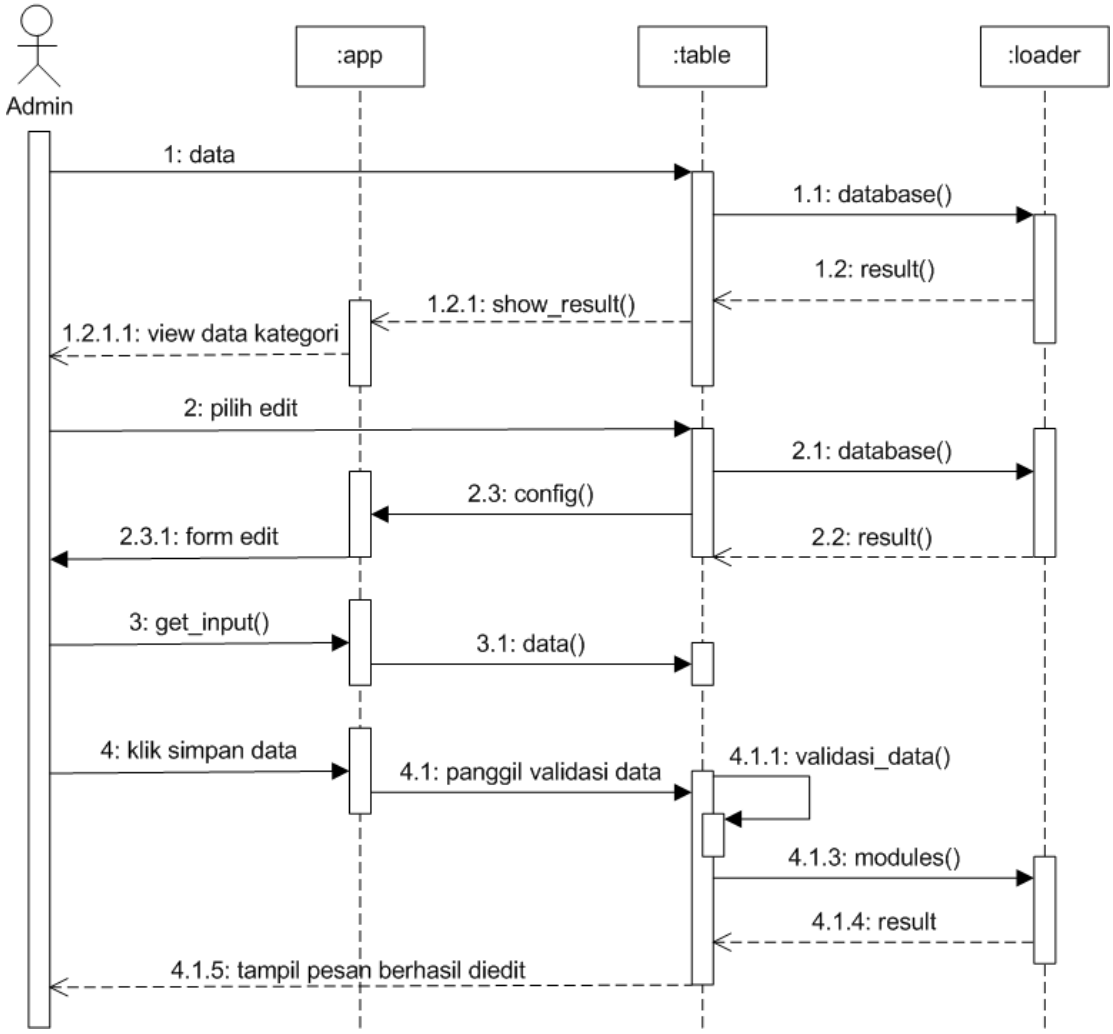

**Gambar 3.50 *Sequence diagram* hapus data meja**

### 3.4.9.8 Tambah Kategori



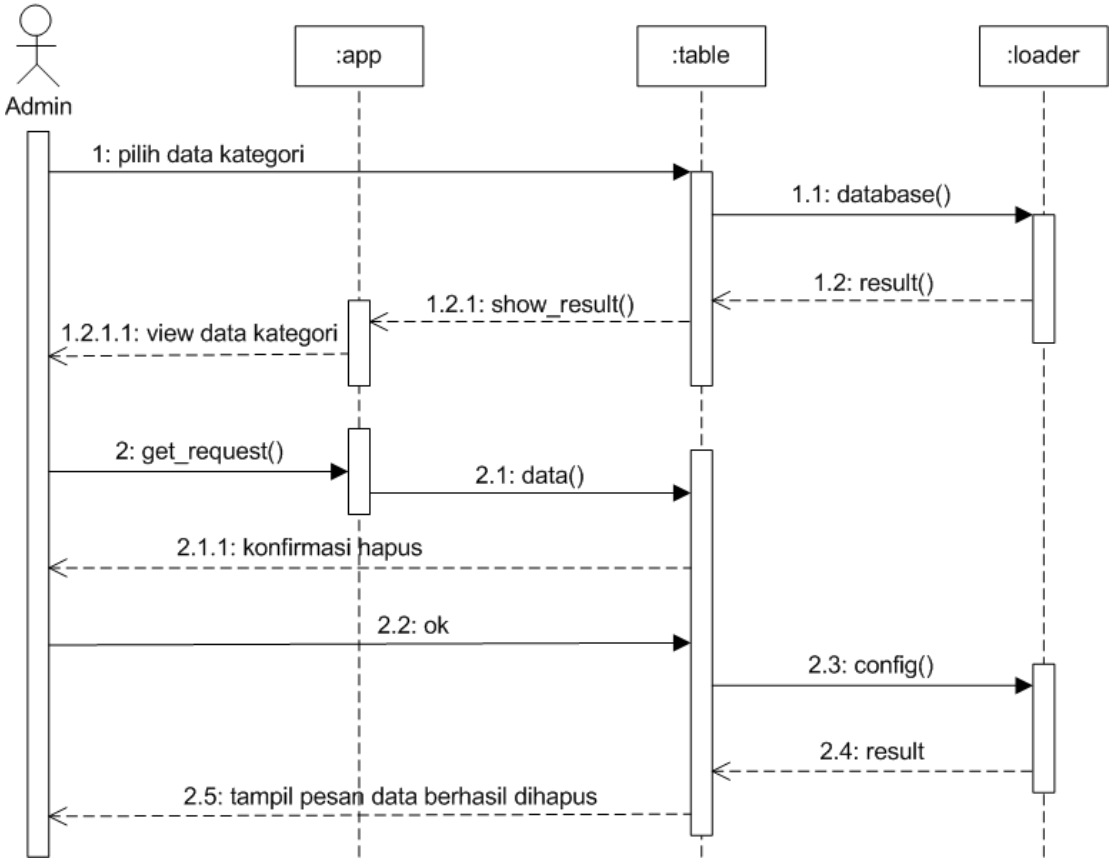

**Gambar 3.51 *Sequence diagram* tambah kategori**

### 3.4.9.9 Edit Kategori



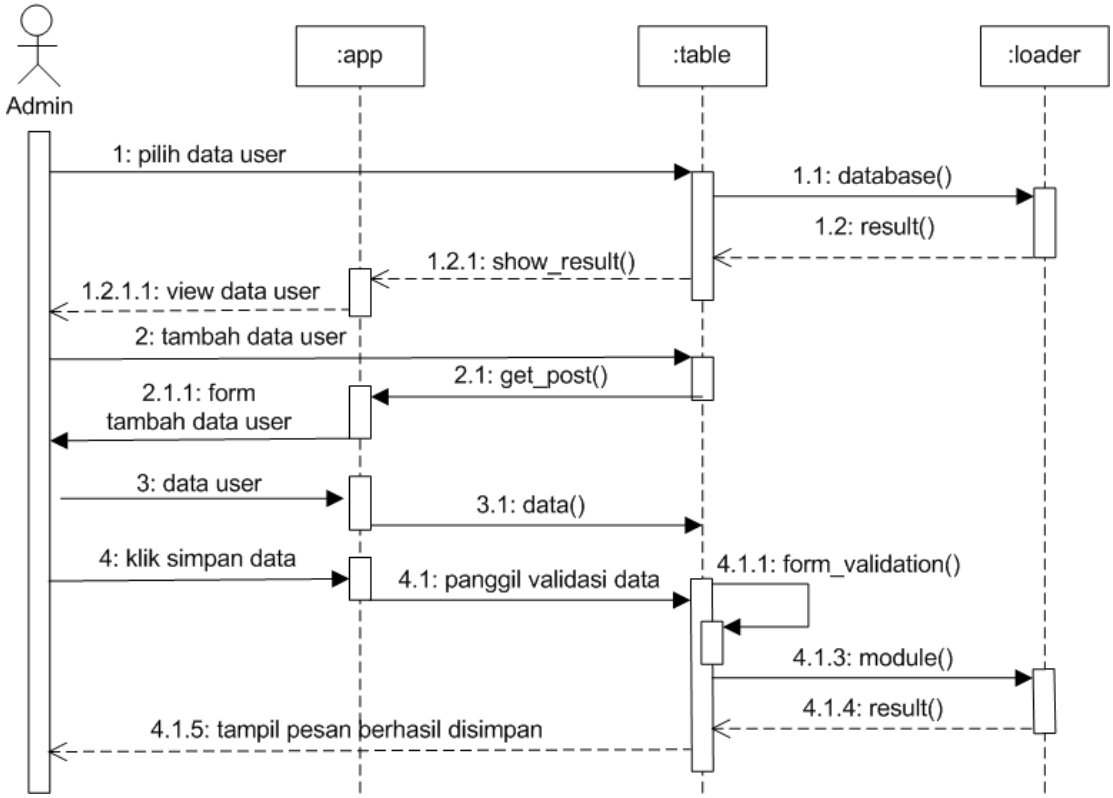

**Gambar 3.52 *Sequence diagram* edit kategori**

### 3.4.9.10 Hapus Kategori



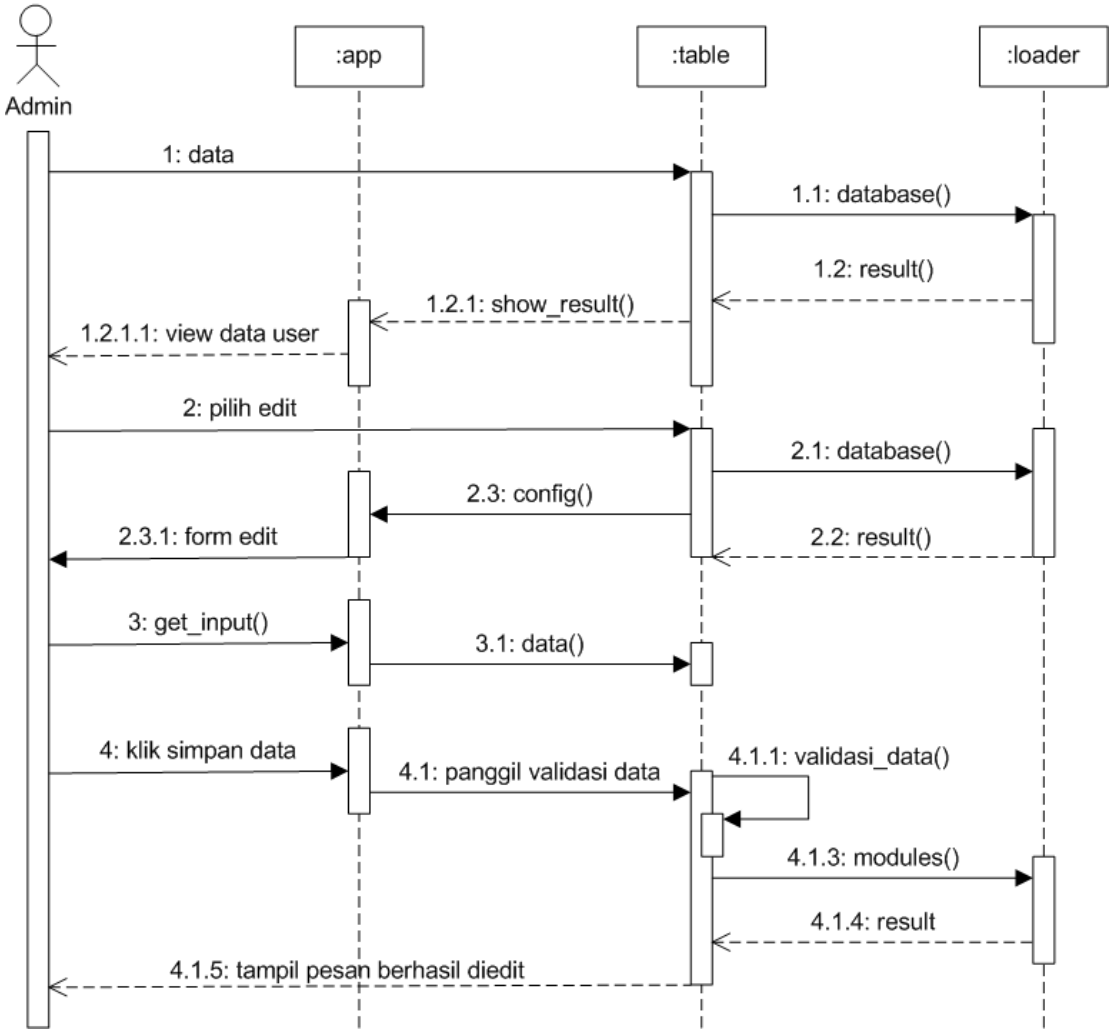

**Gambar 3.53 *Sequence diagram* hapus kategori**

### 3.4.9.11 Tambah Data *User*



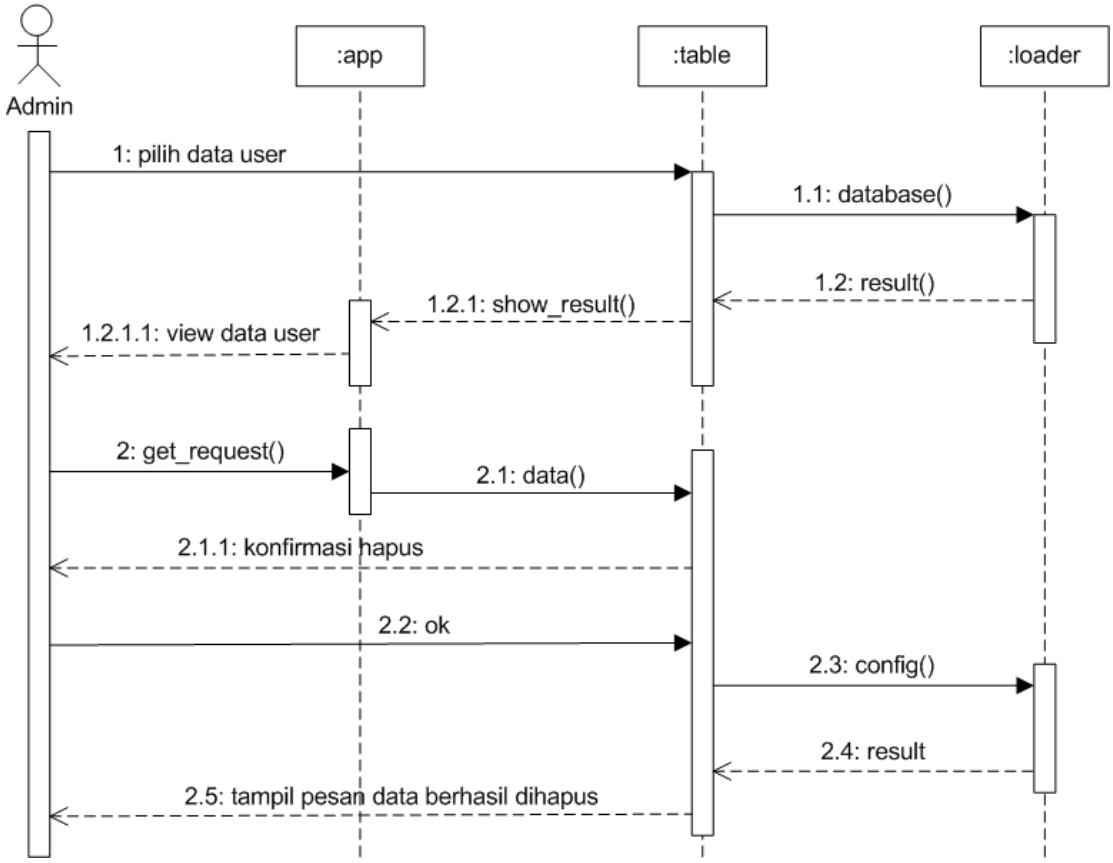

**Gambar 3.54 *Sequence diagram* tambah data *user***

### 3.4.9.12 Edit Data *User*



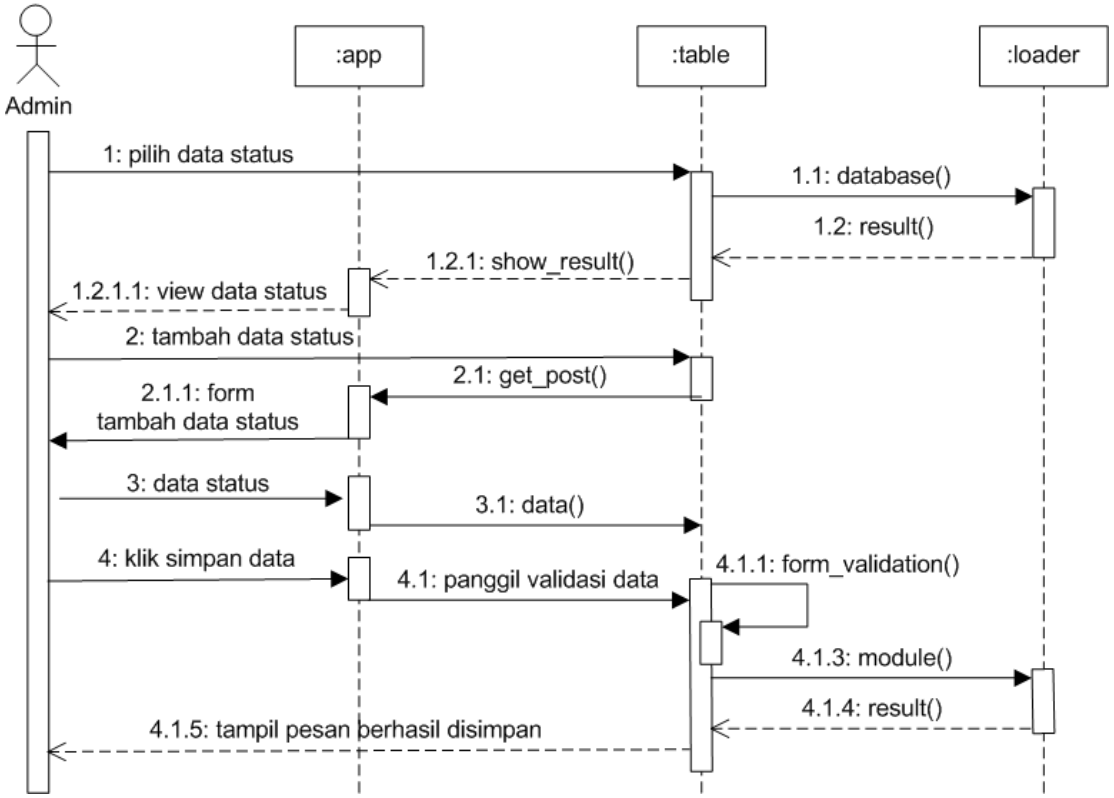

**Gambar 3.55 *Sequence diagram* edit data *user***

### 3.4.9.13 Hapus Data *User*



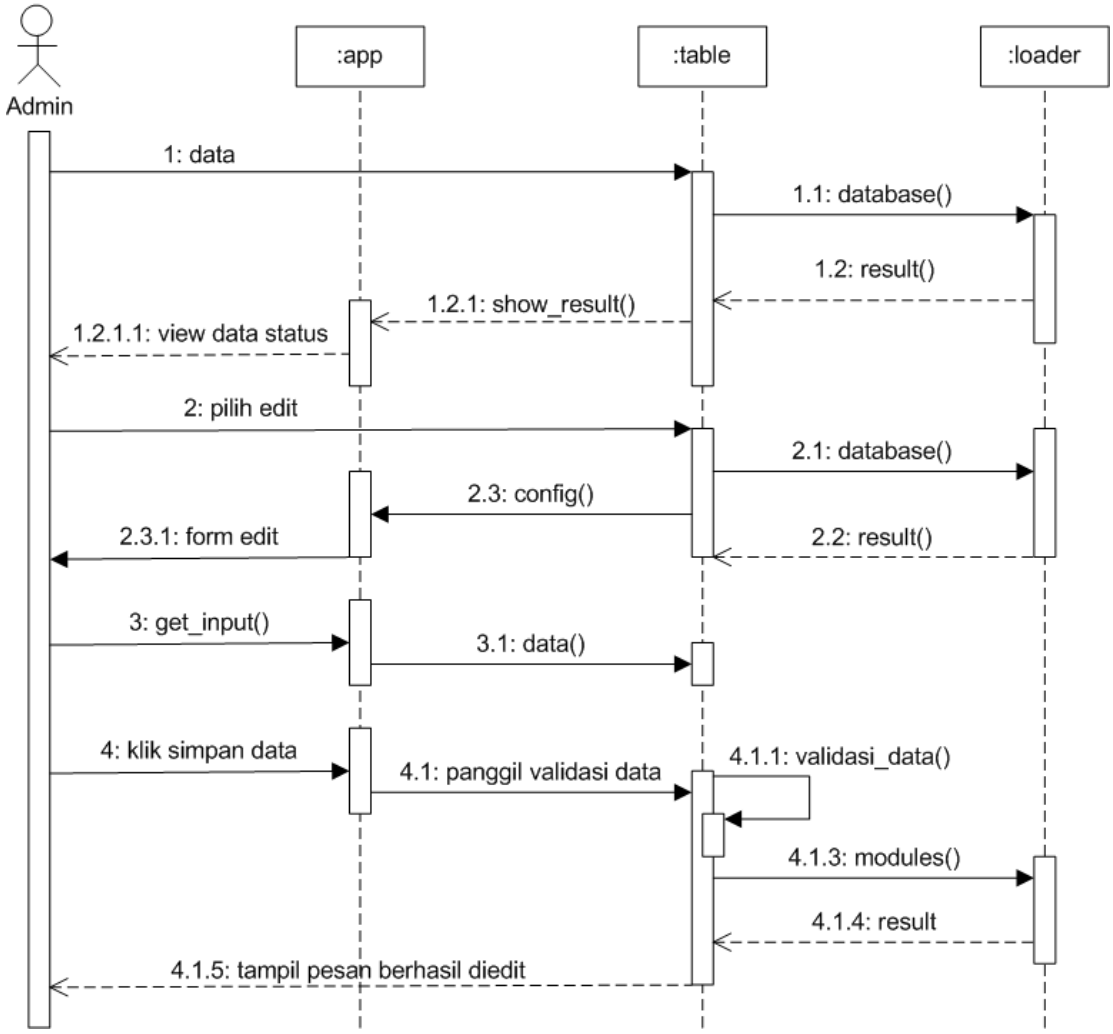

**Gambar 3.56 *Sequence diagram* hapus data *user***

### 3.4.9.14 Tambah Data Status



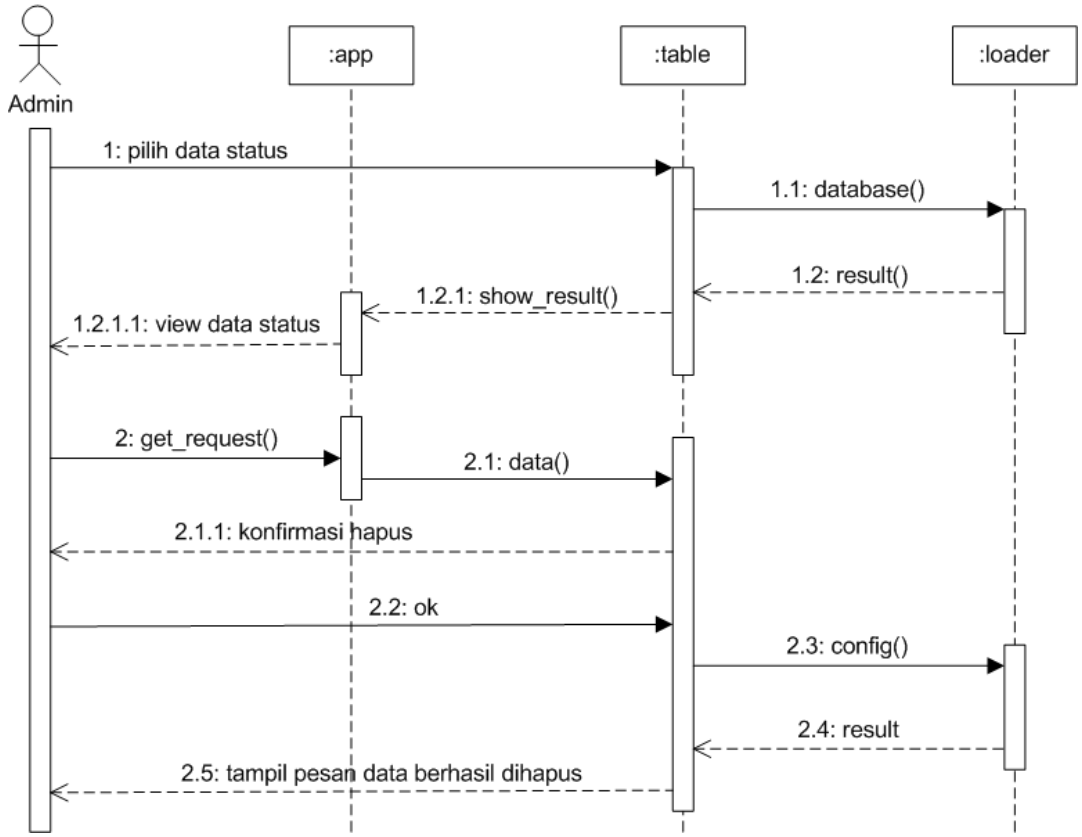

**Gambar 3.57 *Sequence diagram* tambah data status**

### 3.4.9.15 Edit Data Status



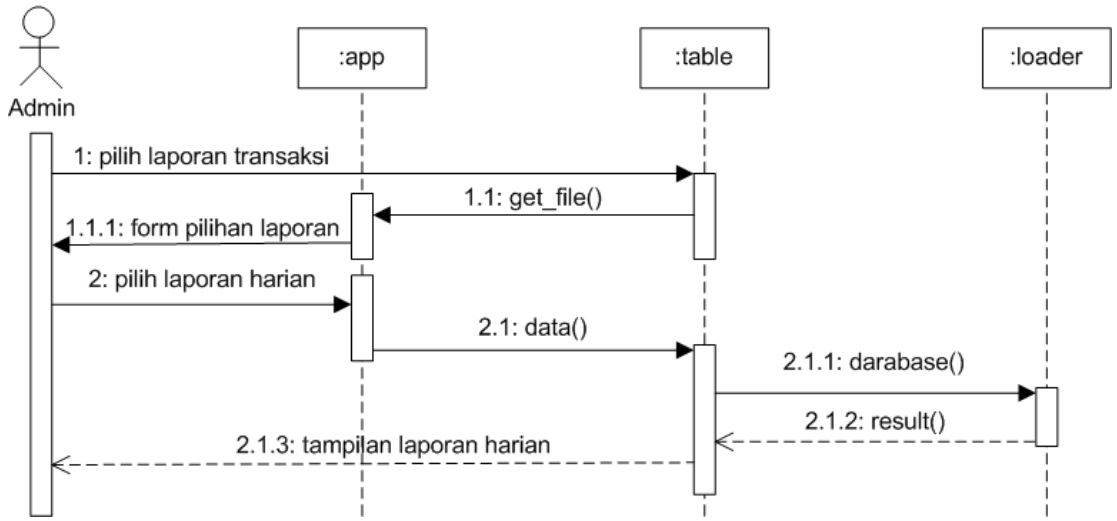

**Gambar 3.58 *Sequence diagram* edit data status**

### 3.4.9.16 Hapus Data Status



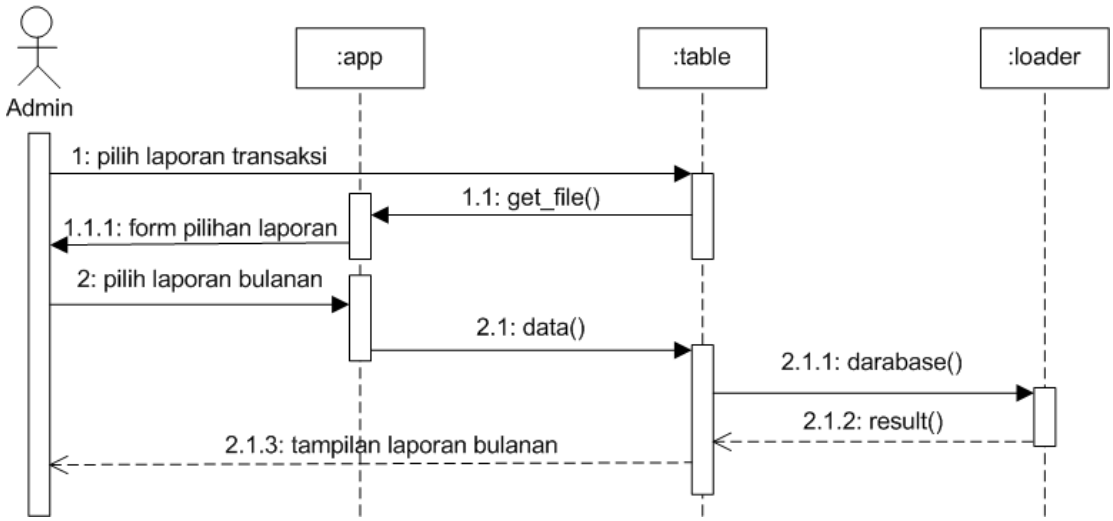

**Gambar 3.59 *Sequence diagram* hapus data status**

### 3.4.9.17 Cetak Laporan Transaksi Harian



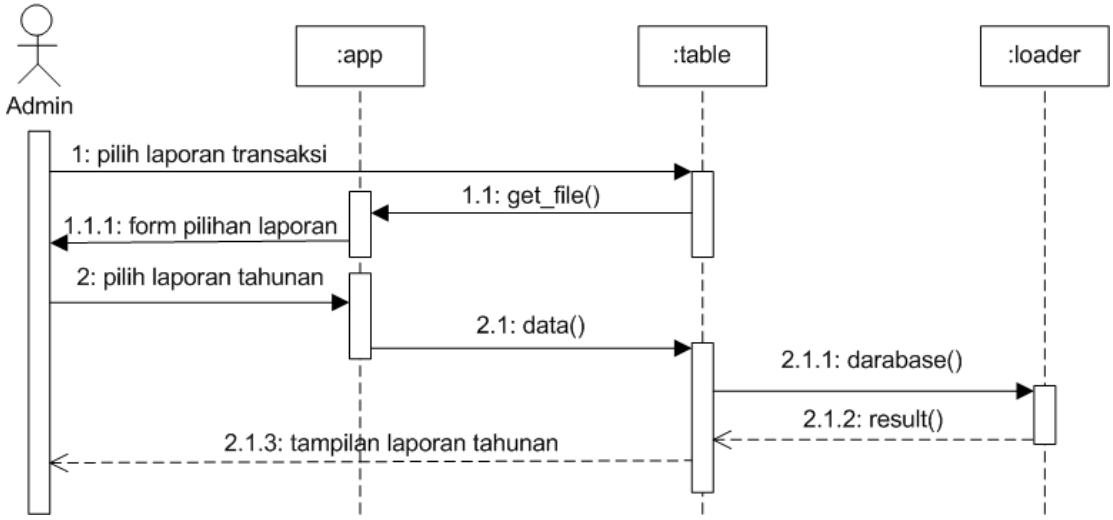

**Gambar 3.60 *Sequence diagram* laporan transaksi harian**

### 3.4.9.18 Cetak Laporan Transaksi Bulanan




**Gambar 3.61 *Sequence diagram* laporan transaksi bulanan**

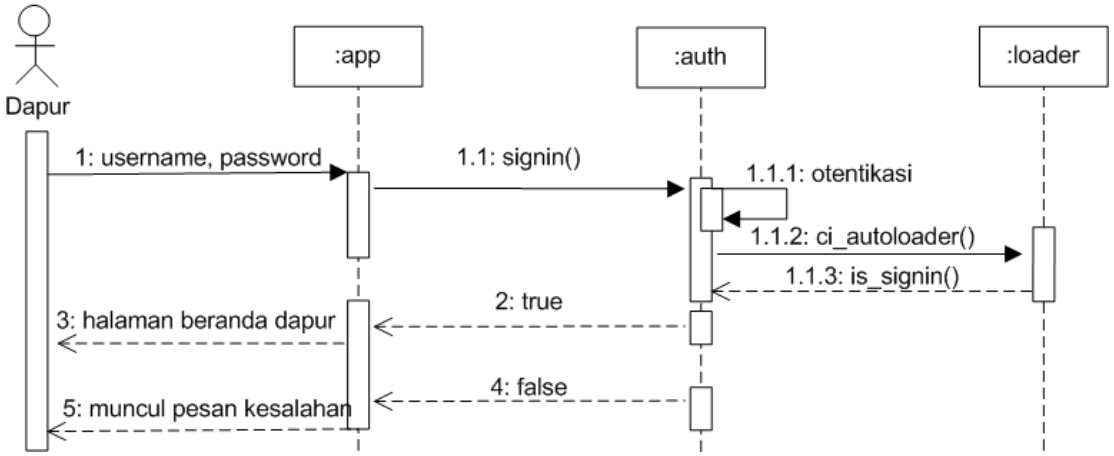
### 3.4.9.19 Cetak Laporan Transaksi Tahunan




**Gambar 3.62 *Sequence diagram* laporan transaksi tahunan**

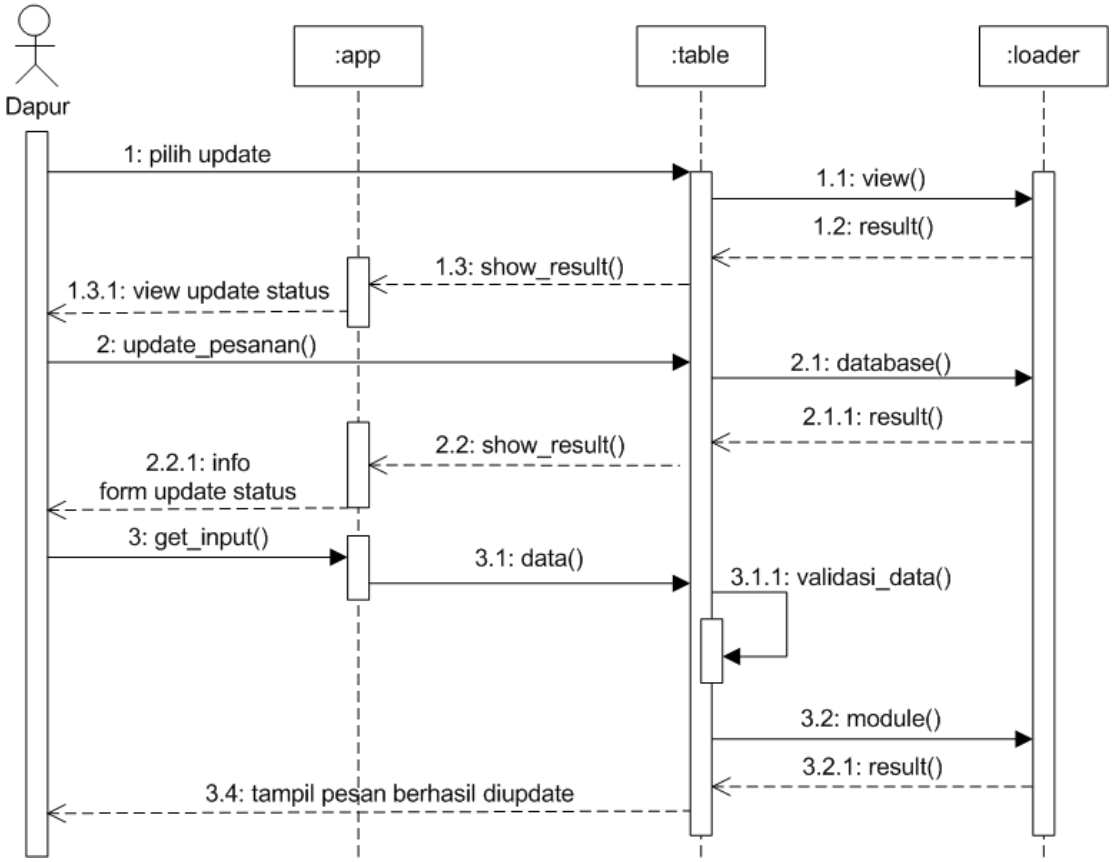
### 3.4.10 *Sequence diagram* pada Aplikasi Dapur

#### 3.4.10.1 *Login* Dapur



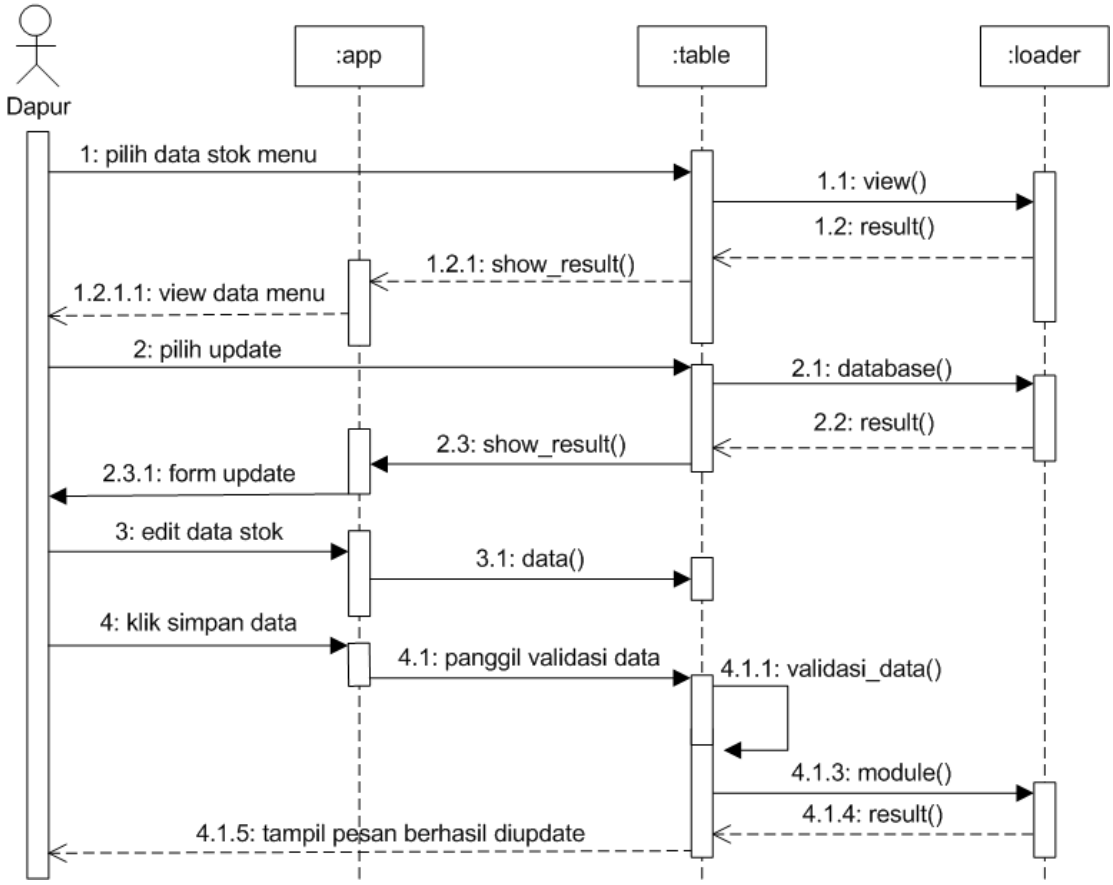

**Gambar 3.63 *Sequence diagram login* dapur**

#### 3.4.10.2 *Update* Status Pesanan




**Gambar 3.64 *Sequence diagram update* status pesanan**

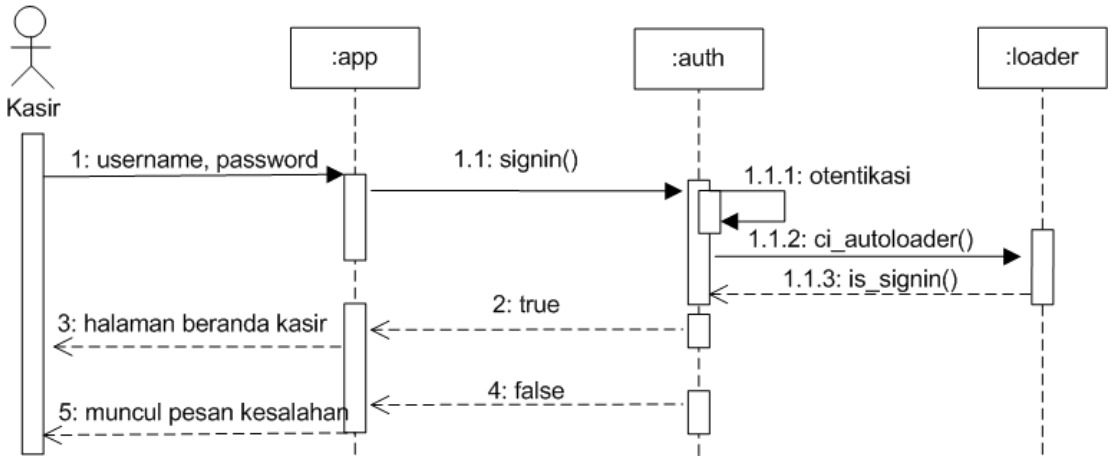
### 3.4.10.3 *Update* Data Stok Menu




**Gambar 3.65 *Sequence diagram update* data stok menu**

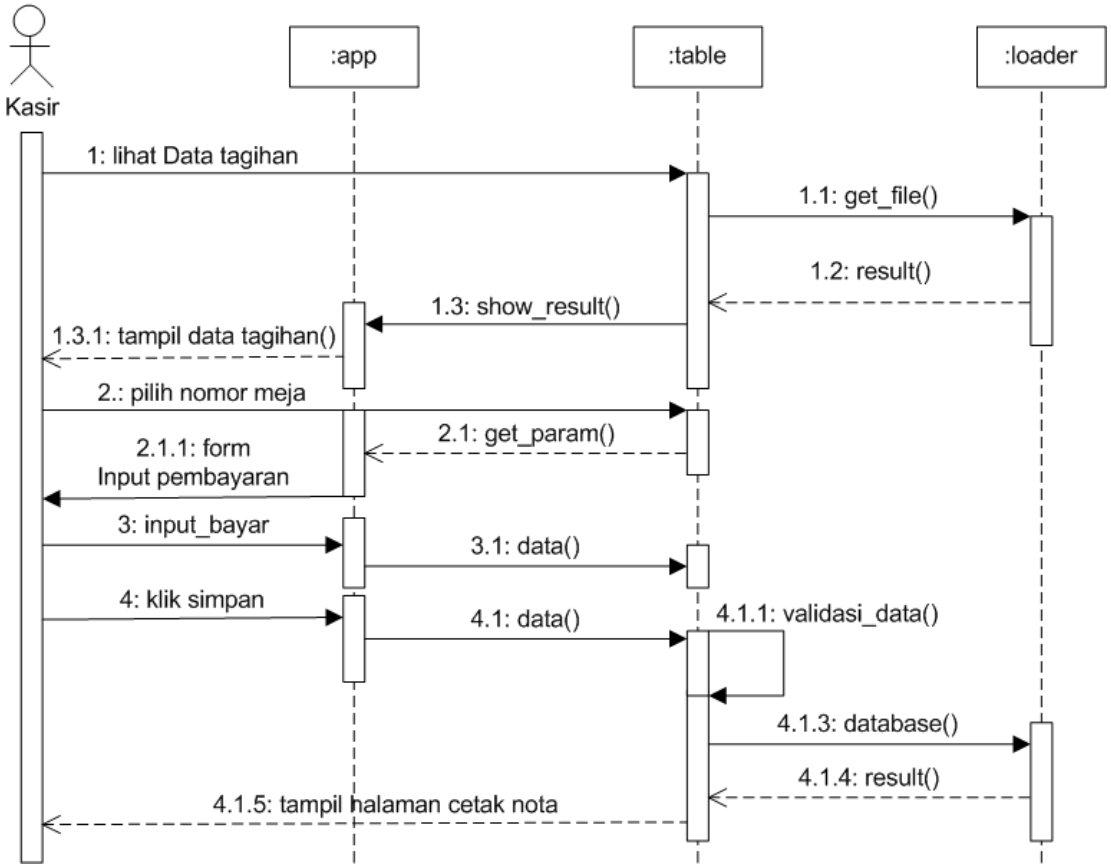
### 3.4.11 *Sequence diagram* pada Aplikasi Kasir

#### 3.4.11.1 *Login* kasir



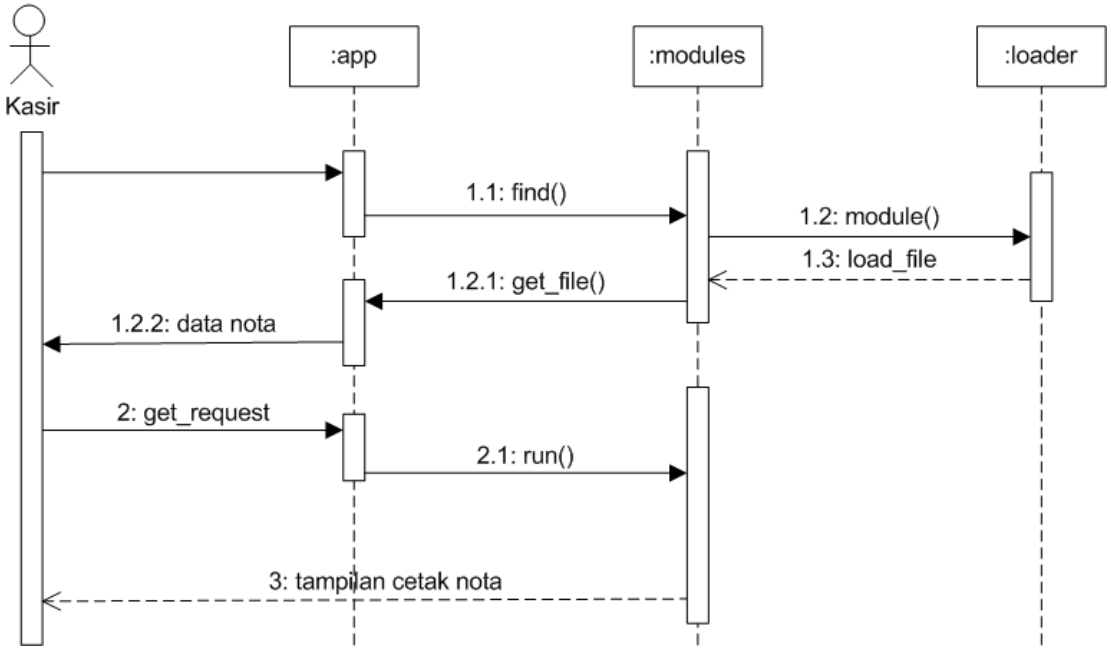

**Gambar 3.66 *Sequence diagram login* kasir**

### 3.4.11.2 *Input* Pembayaran



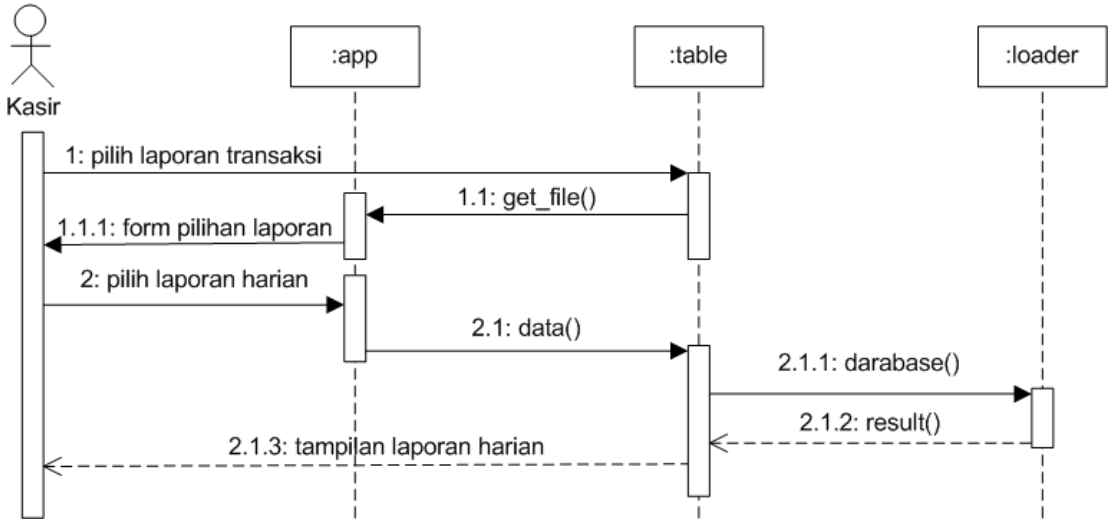

**Gambar 3.67 *Sequence diagram input* pembayaran**

### 3.4.11.3 Cetak Nota



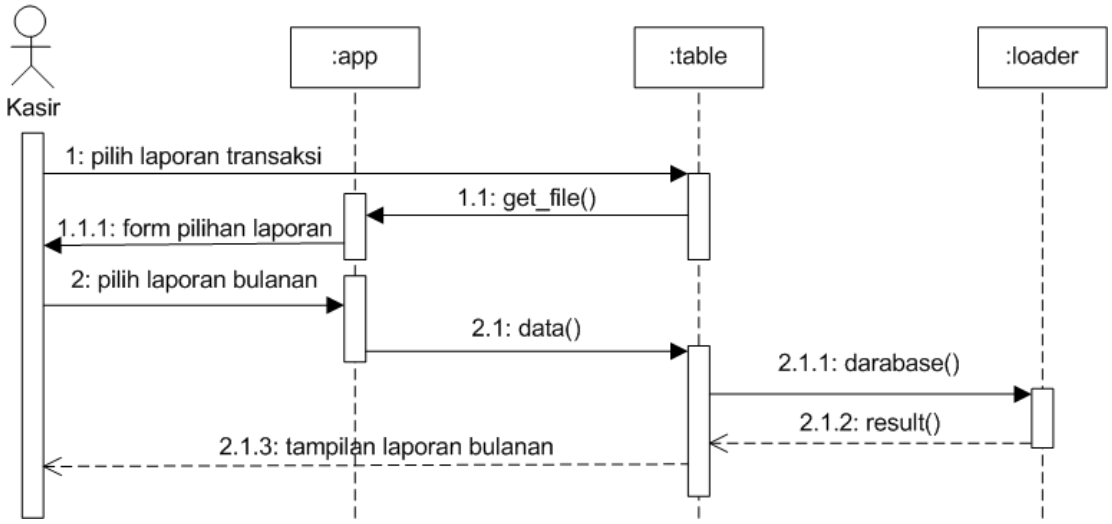

**Gambar 3.68 *Sequence diagram* cetak nota**

### 3.4.11.4 Cetak Laporan Transaksi Harian



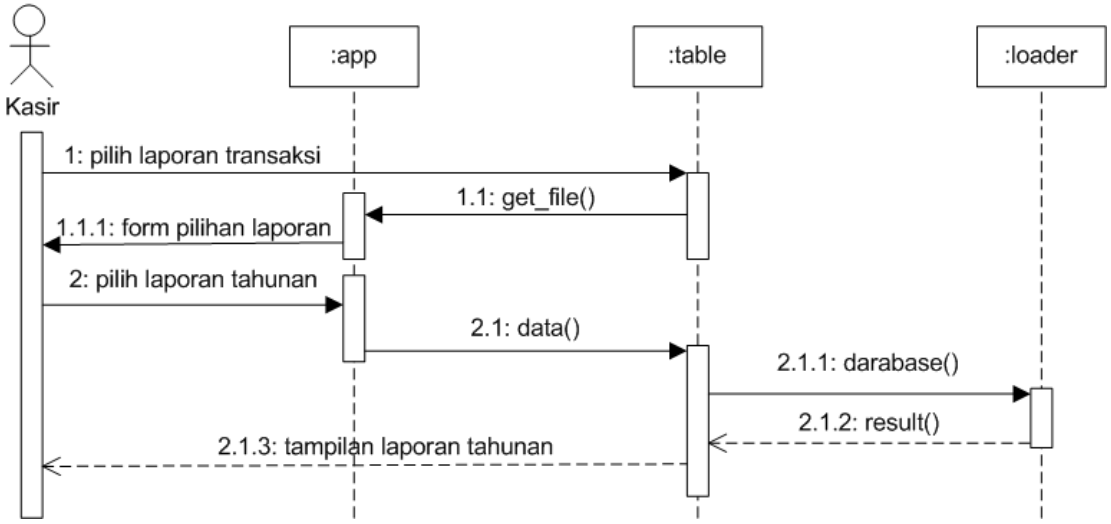

**Gambar 3.69 *Sequence diagram* cetak laporan transaksi harian**

### 3.4.11.5 Cetak Laporan Transaksi Bulanan



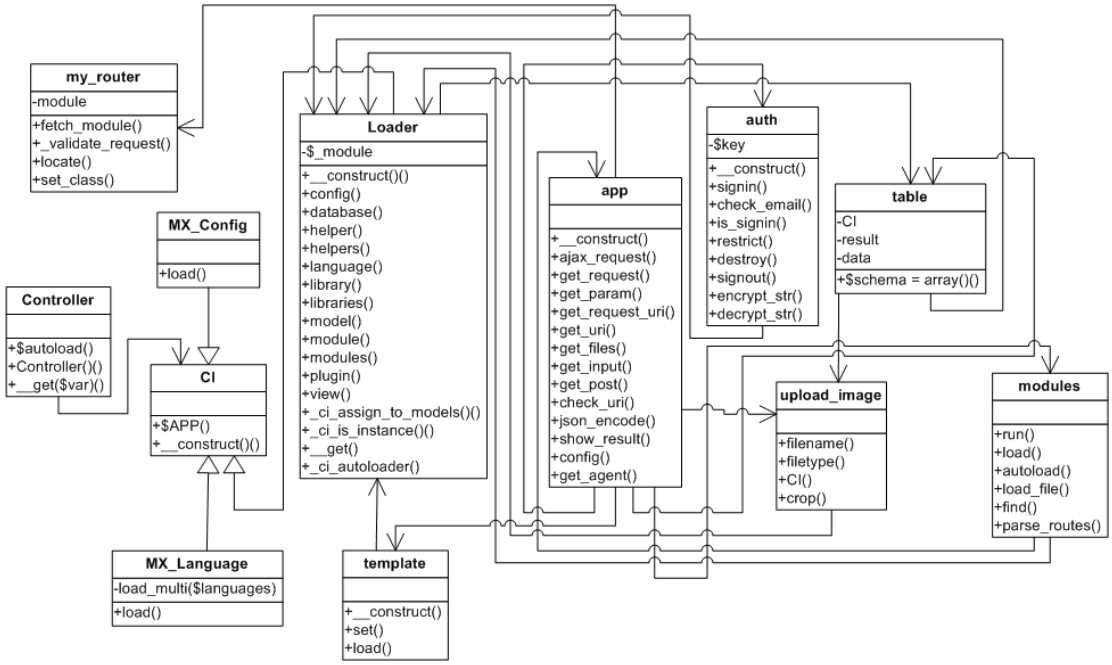

**Gambar 3.70 *Sequence diagram* cetak laporan transaksi bulanan**

### 3.4.11.6 Cetak Laporan Transaksi Tahunan



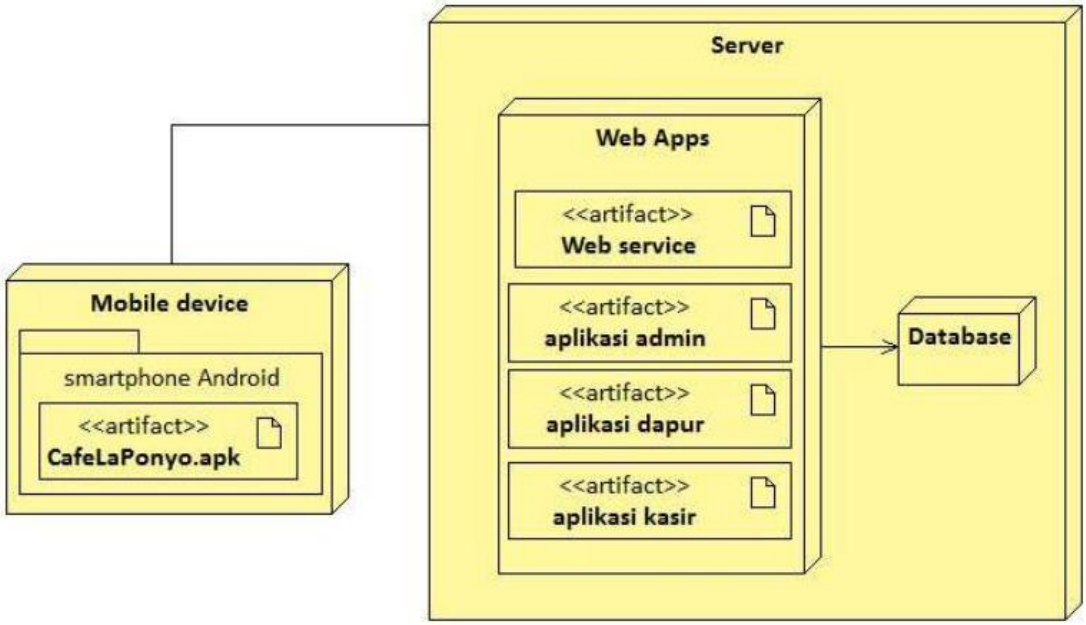

**Gambar 3.71 *Sequence diagram* cetak laporan transaksi tahunan**

### 3.4.12 *Class Diagram Web Base*




**Gambar 3.72 *Class diagram Web Base***

### 3.4.13 *Diagram Deployment*

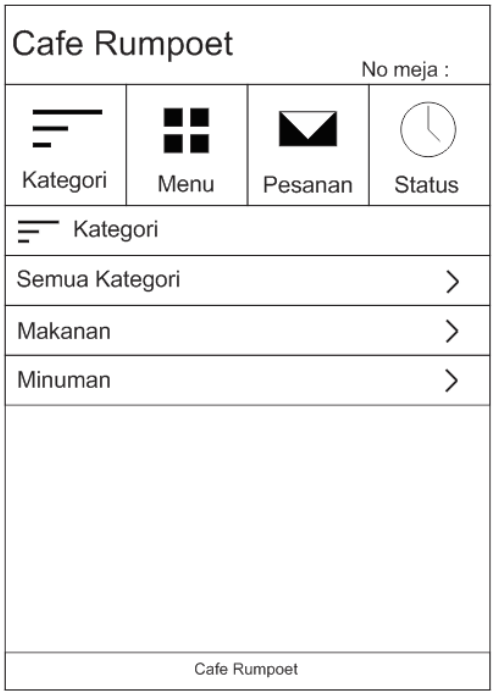



**Gambar 3.73 *Diagram Deployment***

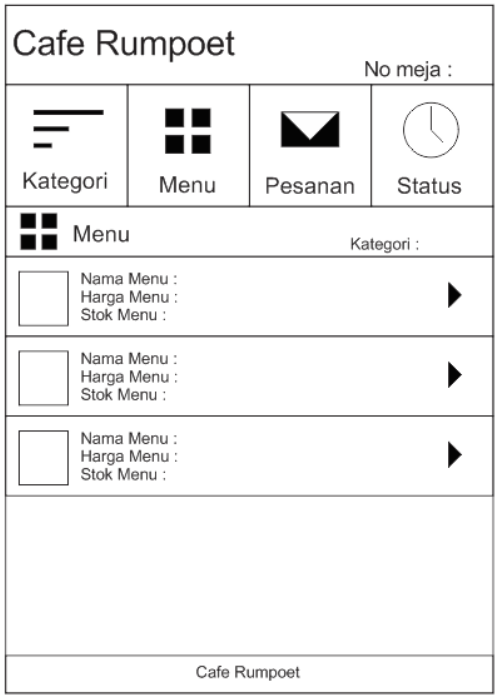
### 3.4.14 Perancangan Antar Muka

*Interface* atau antarmuka merupakan tampilan dari suatu program aplikasi yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara program dengan *user*. Sistem yang akan dibangun diharapkan menyediakan *interface* yang mudah dipahami dan digunakan oleh *user*. Berikut ini adalah perancangan antarmuka pada sistem yang akan dibuat.

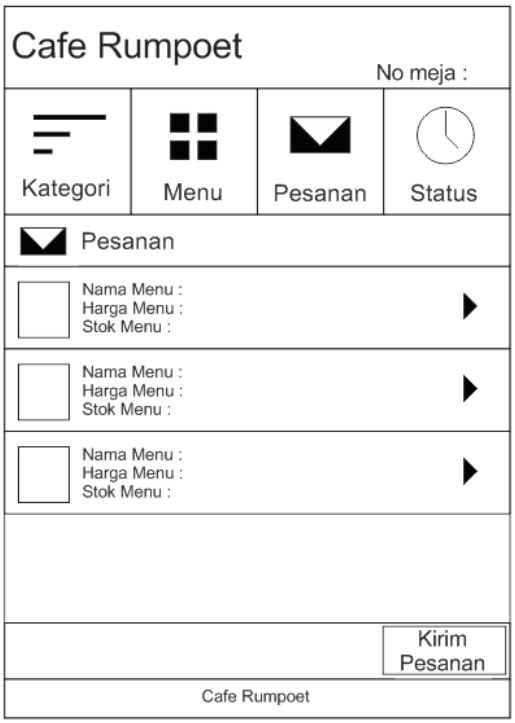
### 3.4.14.1 Perancangan Antar Muka Android



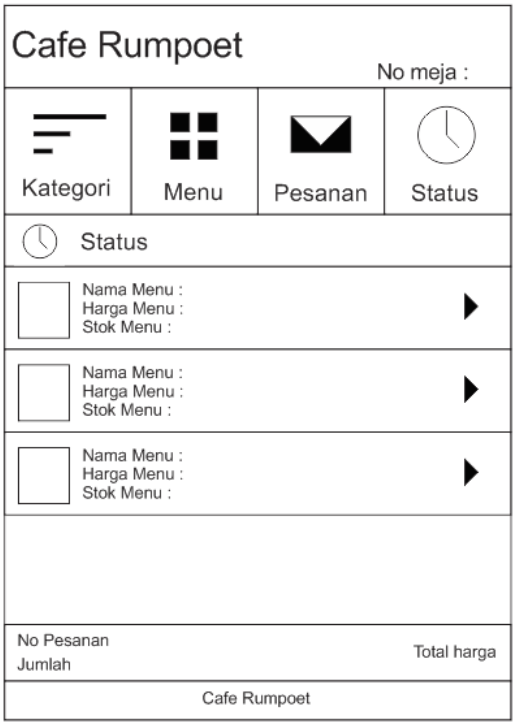

**Gambar 3.74 Antarmuka kategori**




**Gambar 3.75 Antarmuka menu**




**Gambar 3.76 Antarmuka pesanan**




**Gambar 3.77 Antarmuka status**




**Gambar 3.78 Antarmuka detail pesanan**



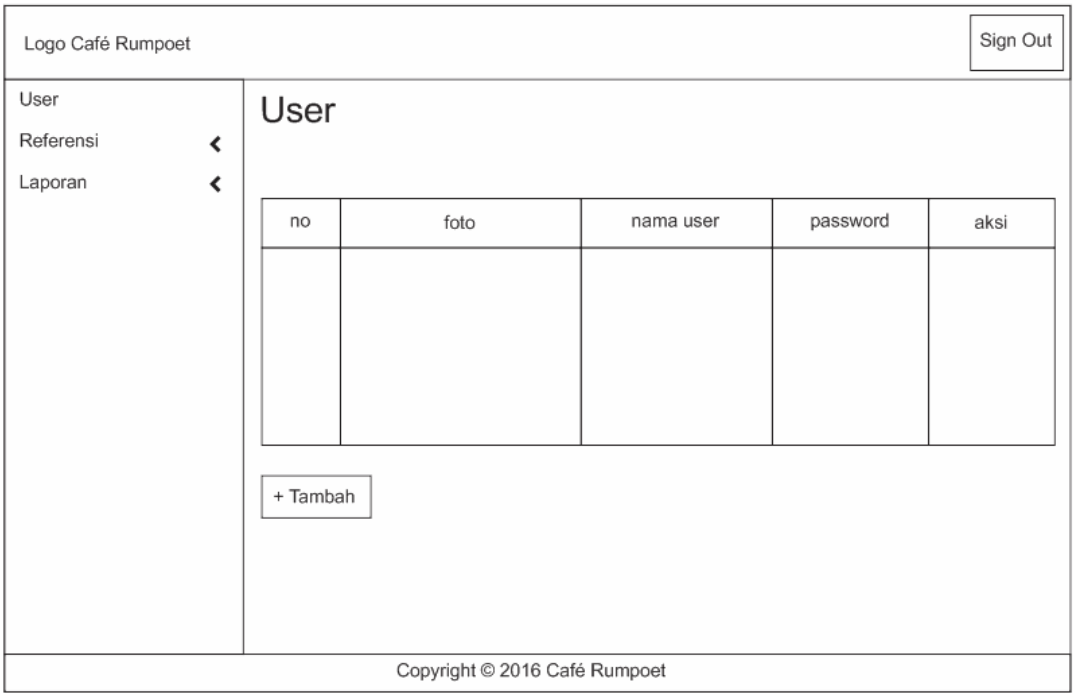

**Gambar 3.79 Antarmuka detail menu**

### 3.4.14.2 Perancangan antarmuka *login web base*

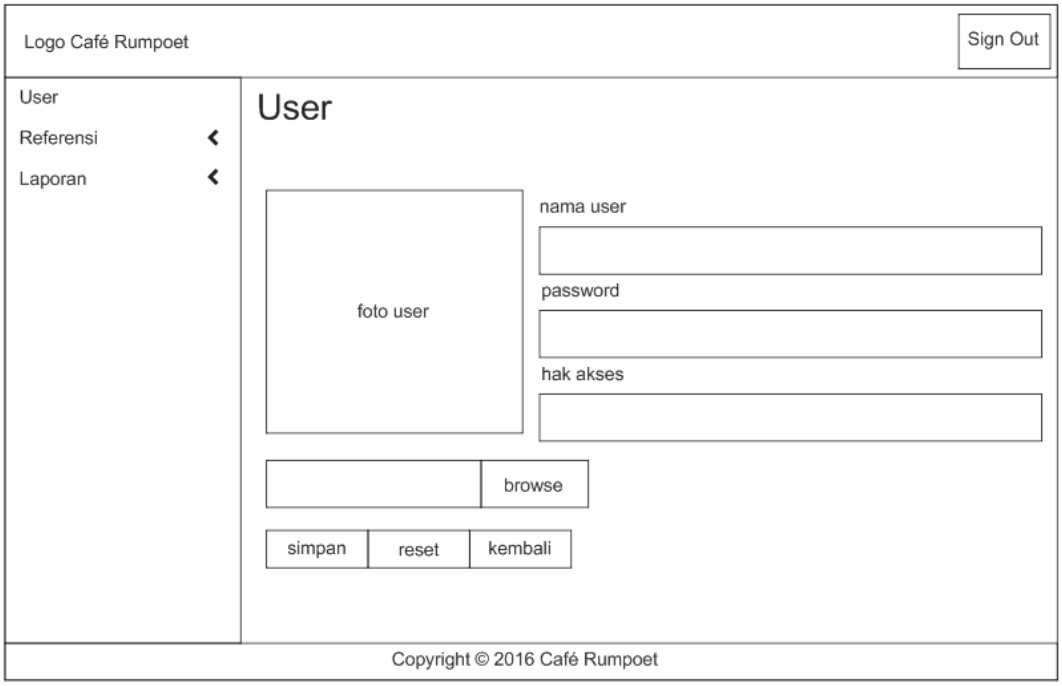



**Gambar 3.80 Antarmuka *login***

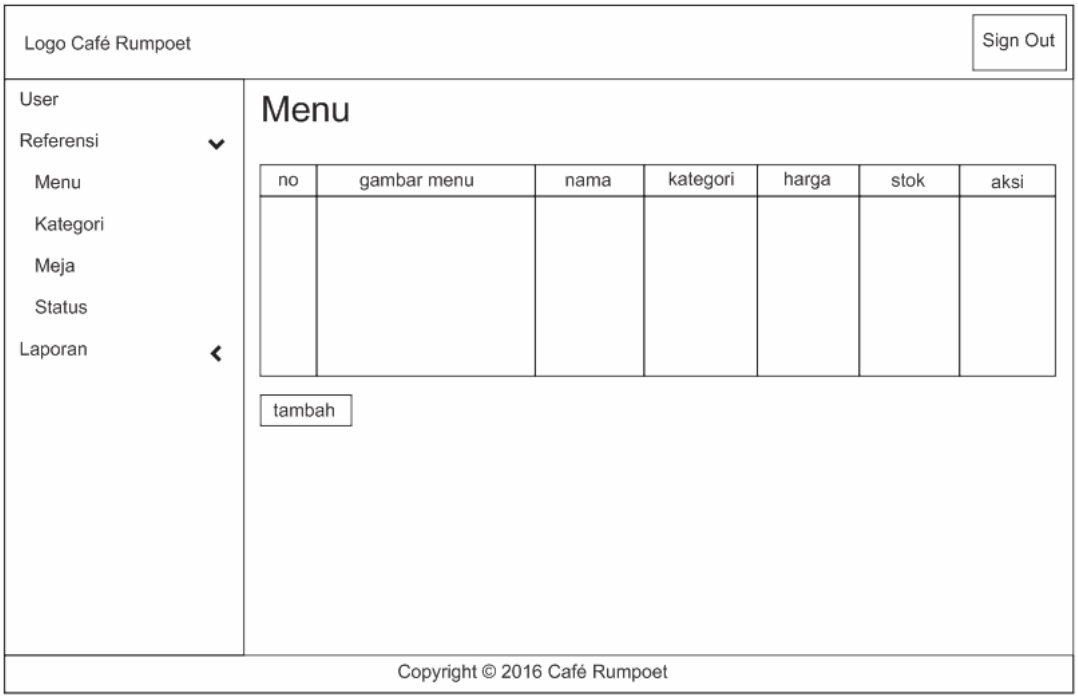
### 3.4.14.3 Perancangan antarmuka admin



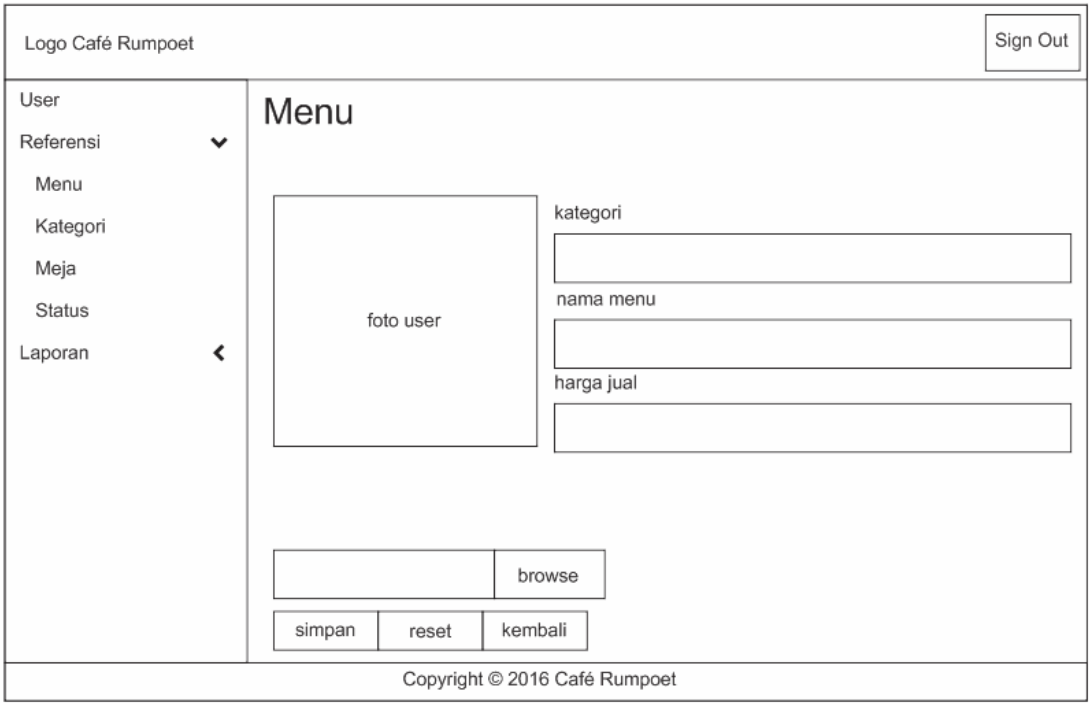

**Gambar 3.81 Antarmuka *User***



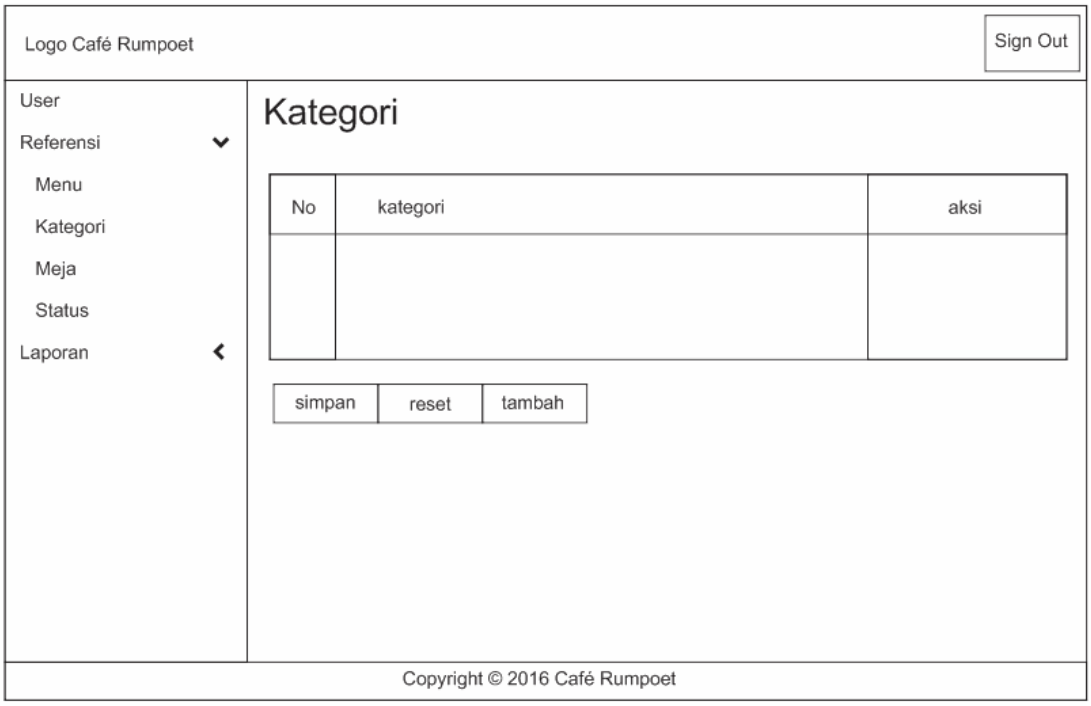

**Gambar 3.82 Antarmuka tambah & edit *user***



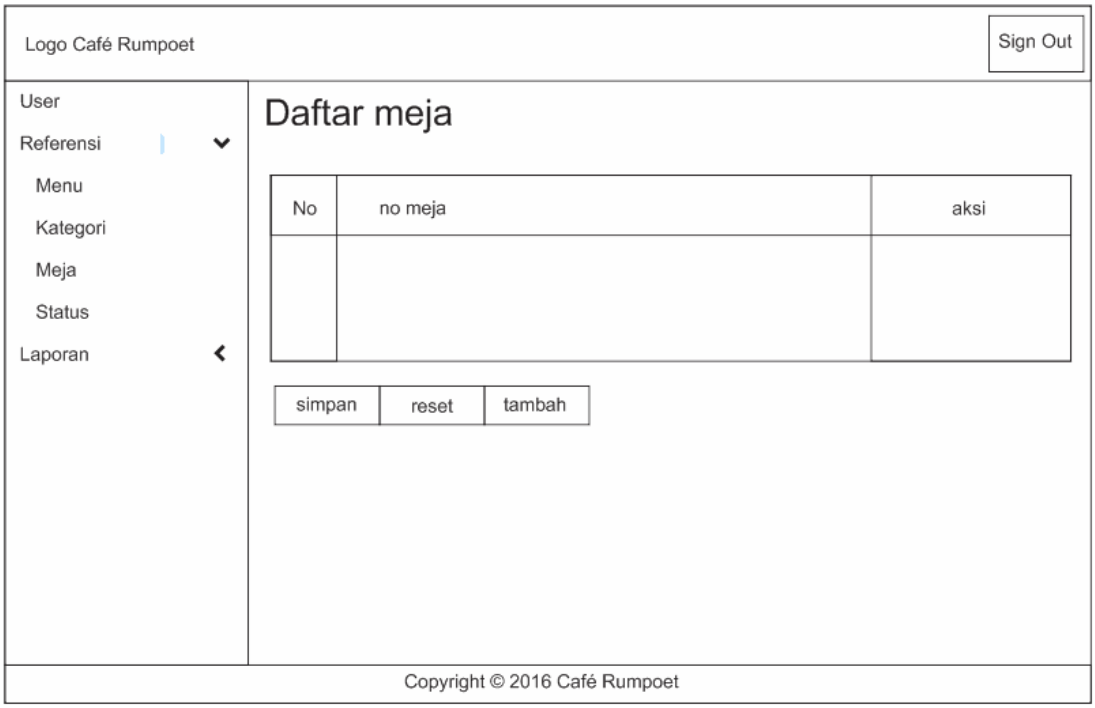

**Gambar 3.83 Antarmuka menu**



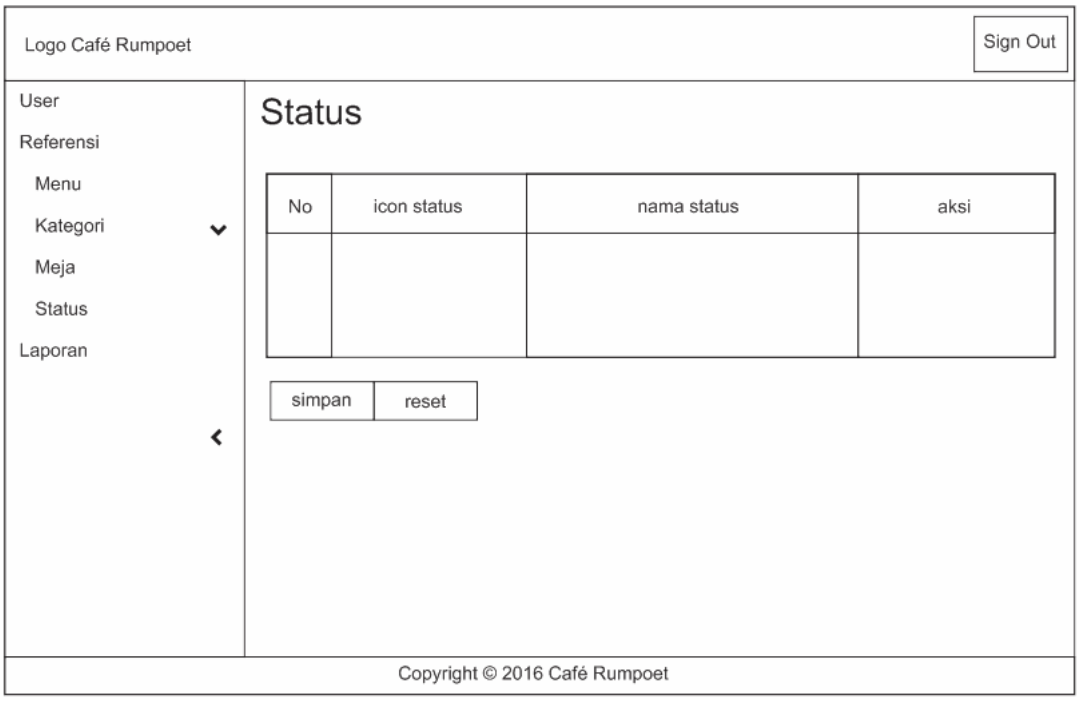

**Gambar 3.84 Antarmuka tambah & edit menu**



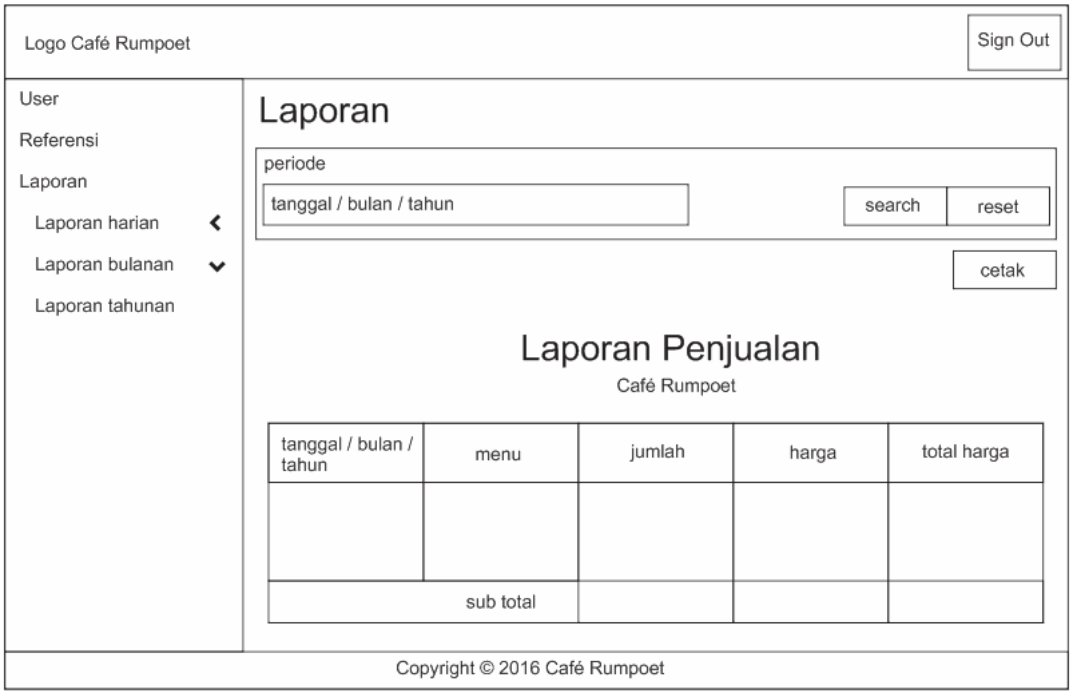

**Gambar 3.85 Antarmuka kategori**




**Gambar 3.86 Antarmuka meja**

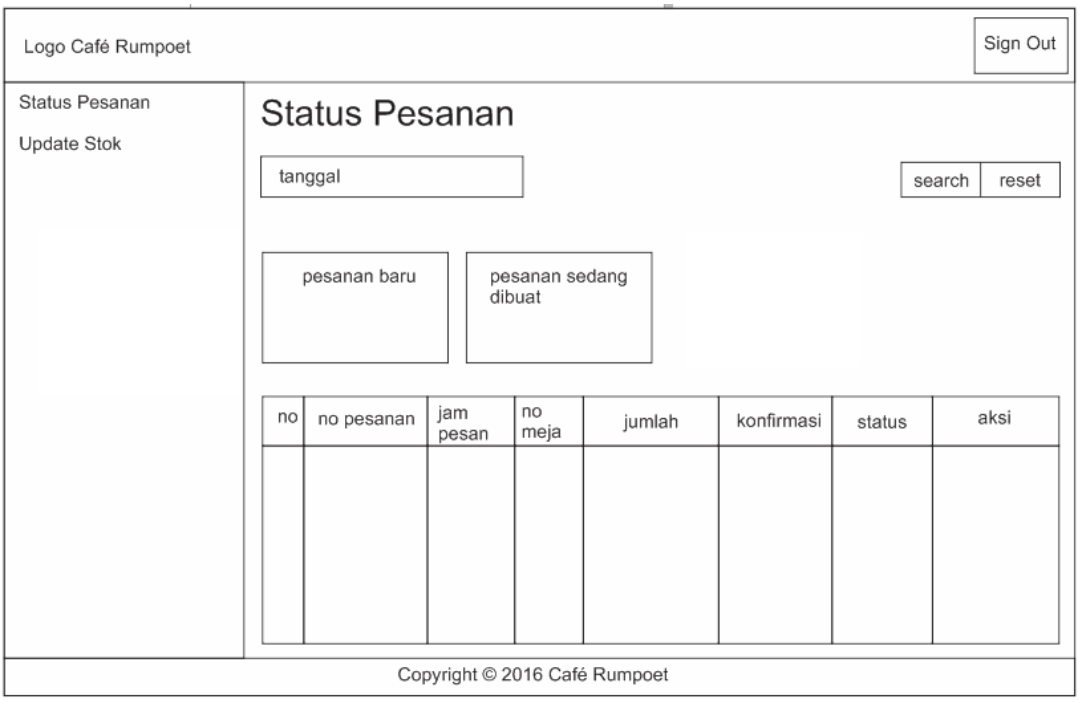



**Gambar 3.87 Antarmuka status**

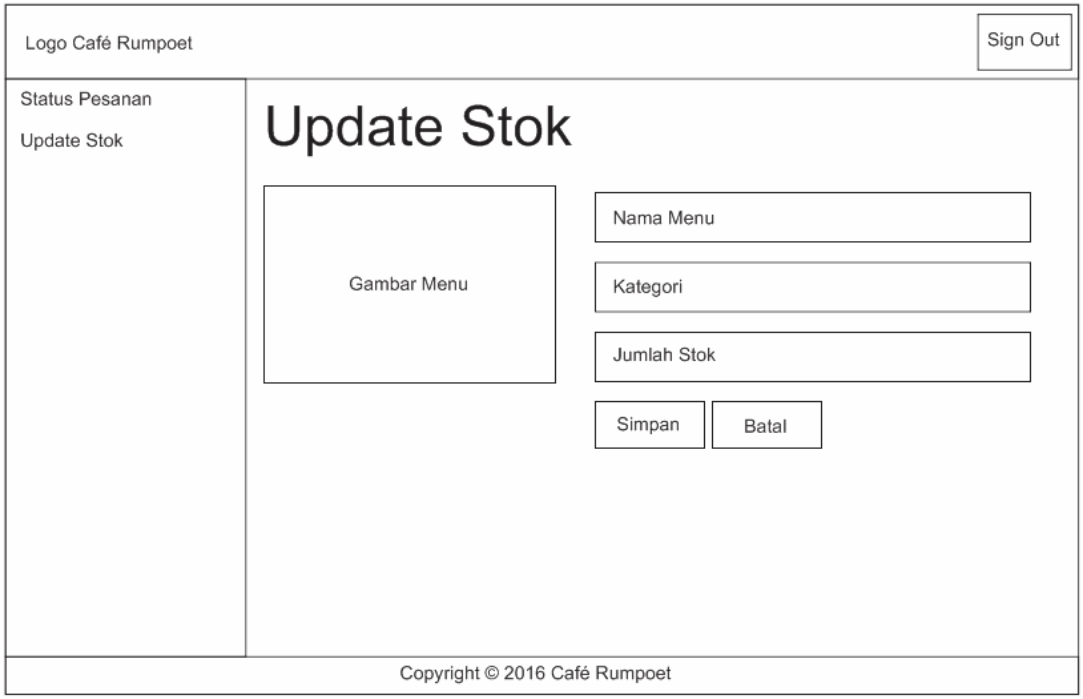



**Gambar 3.88 Antarmuka laporan**

### 3.4.14.4 Perancangan antarmuka bagian dapur

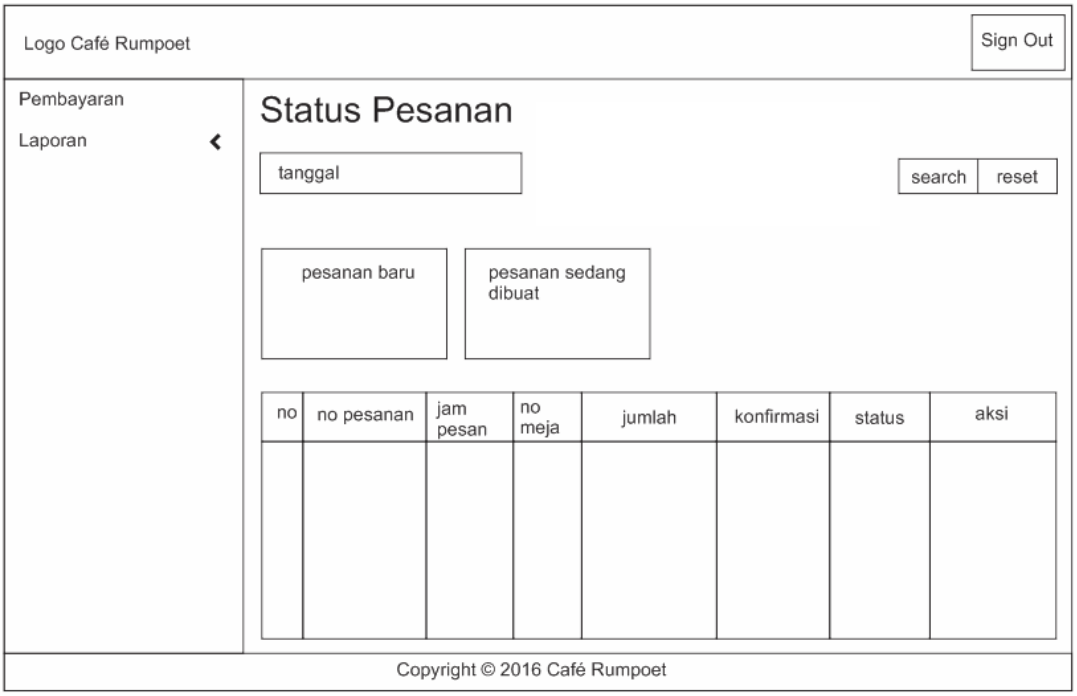



**Gambar 3.89 Antarmuka status pesanan**

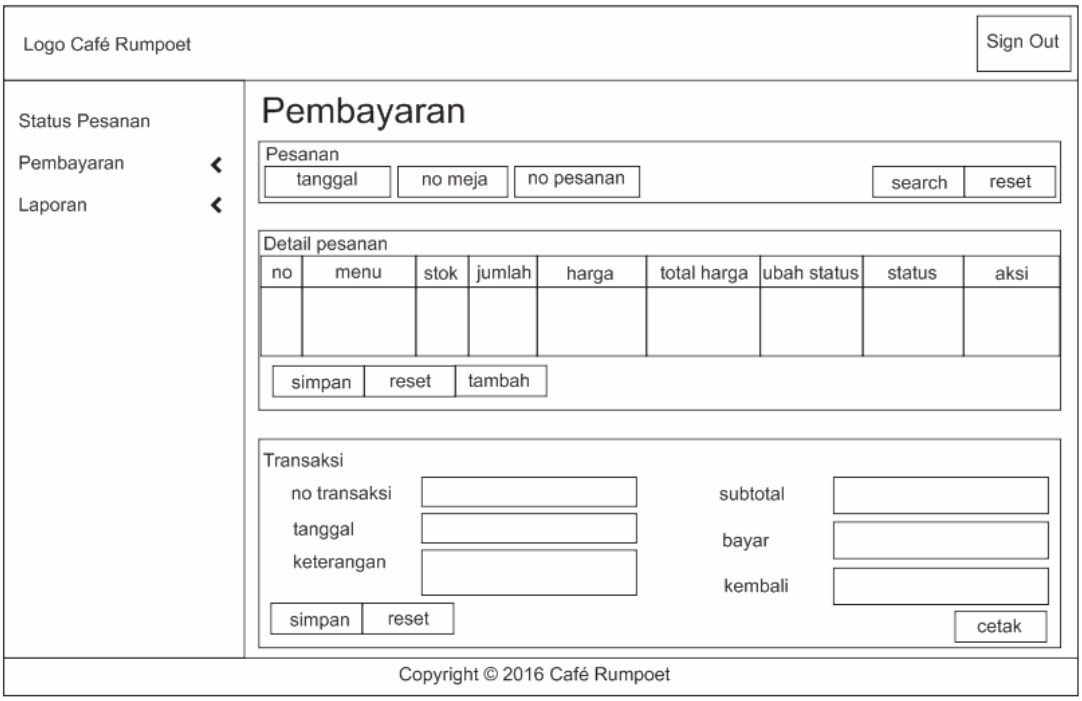



**Gambar 3.90 Antarmuka *update* stok**

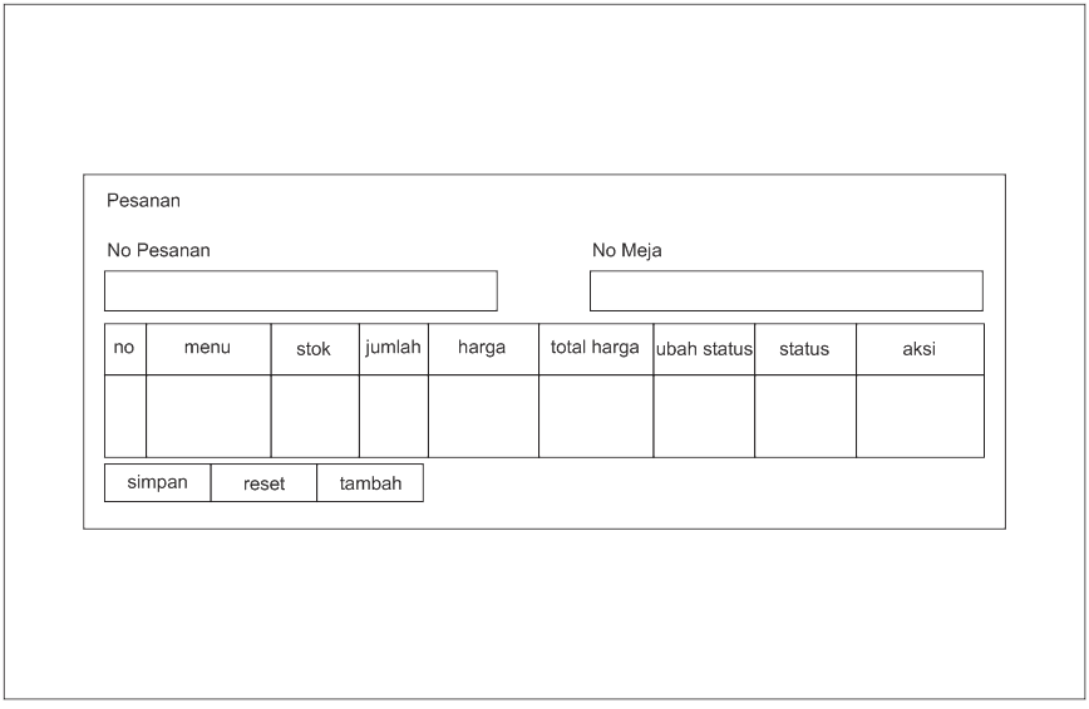
### 3.4.14.5 Perancangan antarmuka pada bagian kasir



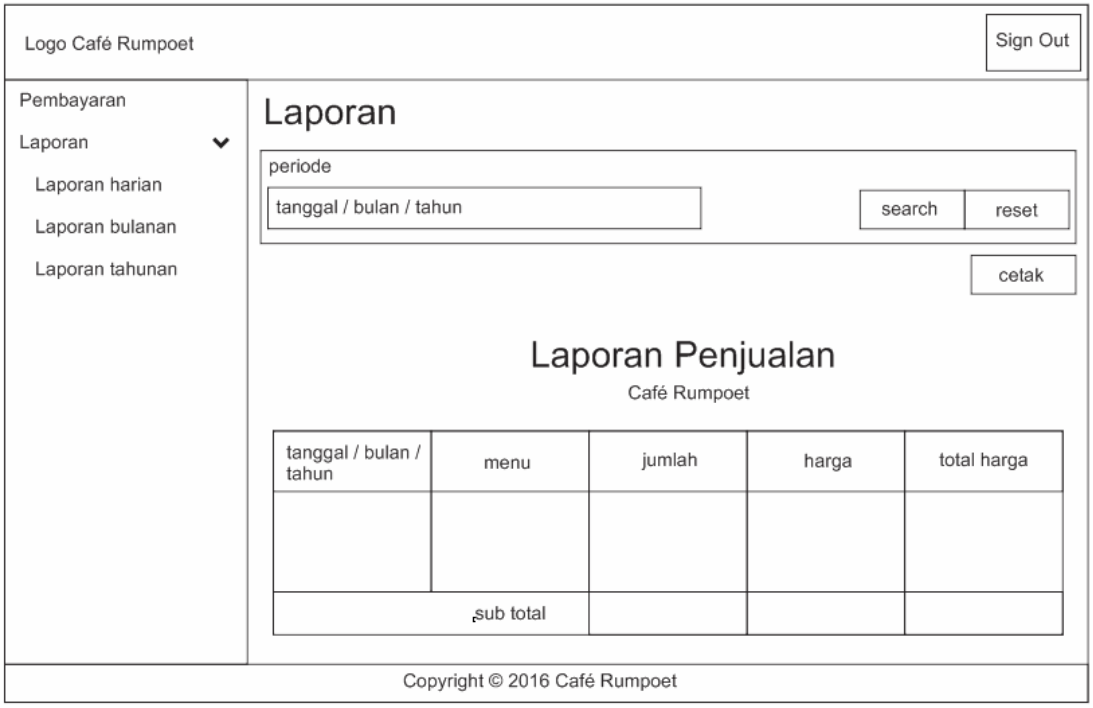

**Gambar 3.91 Antarmuka status pesanan**




**Gambar 3.92 Antarmuka pembayaran**




**Gambar 3.93 Antarmuka buat pesanan baru**



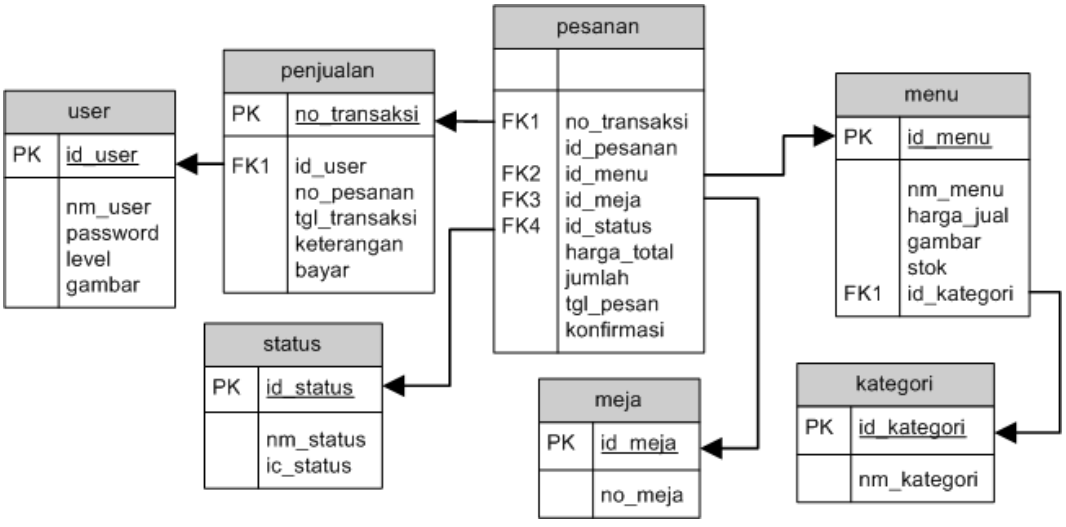

**Gambar 3.94 Antarmuka laporan**

### 3.4.15 Perancangan Basis Data

Perancangan basis data adalah menciptakan atau merancang kumpulan data yang terhubung dan tersimpan secara bersama-sama.

#### 3.4.15.1 Skema Relasi

Skema relasi menggambarkan hubungan antar data, arti data dan batasannya dijelaskan dengan baris dan kolom.




**Gambar 3.95 Skema Relasi**

### 3.4.15.2 Struktur Tabel

Tabel-tabel yang terdapat dalam basis data yang digunakan dalam sistem ini adalah sebagai berikut :

#### 3.4.15.2.1 Tabel *User*

Tabel ini adalah perancangan untuk tabel admin yang akan digunakan pada pembangunan sistem. Struktur tabel *user* adalah sebagai berikut :

**Table 3.43 Struktur *user***

| No | Nama field | Tipe | Ukuran | Keterangan |
|---|---|---|---|---|
| 1 | id_user | integer | 11 | *Primary key |
| 2 | nm_user | varchar | 30 | |
| 3 | password | varchar | 40 | |
| 4 | level | char | 1 | |
| 5 | gambar | blob | | |

#### 3.4.15.2.2 Tabel Menu

Tabel ini adalah perancangan untuk tabel menu yang akan digunakan untuk menampung seluruh data menu. Struktur tabel menu adalah sebagai berikut:

**Tabel 3.44 Struktur tabel menu**

| No | Nama field | Tipe | Ukuran | Keterangan |
|---|---|---|---|---|
| 1 | id_menu | integer | 11 | *Primary key |
| 2 | id_kategori | integer | 11 | **Foreign key, kategori (id_kategori) |
| 3 | nm_menu | varchar | 50 | |
| 4 | harga_jual | decimal | 10,2 | |
| 5 | gambar | blob | | |
| 6 | stok | integer | 11 | |

#### 3.4.15.2.3 Tabel Pesanan

Tabel ini adalah perancangan untuk tabel pesanan yang akan digunakan untuk menampung seluruh pemesanan yang diinputkan pelanggan. Struktur tabel data pesan adalah sebagai berikut:

**Tabel 3.45 Struktur tabel pesanan**

| No | Nama field | Tipe | Ukuran | Keterangan |
|---|---|---|---|---|
| 1 | id_pesanan | integer | 11 | |
| 2 | no_transaksi | char | 13 | **Foreign key, penjualan (no_transaksi) |
| 3 | id_menu | integer | 11 | **Foreign key, menu (id_menu) |
| 4 | id_meja | integer | 11 | **Foreign key, meja (id_meja) |
| 5 | id_status | integer | 11 | **Foreign key, status (id_status) |
| 6 | harga_total | decimal | 15,2 | |
| 7 | jumlah | decimal | 15,2 | |
| 8 | tgl_pesan | datetime | | |
| 9 | konfirmasi | tinyint | 1 | |

#### 3.4.15.2.4 Tabel Kategori

Tabel ini adalah perancangan untuk tabel kategori yang akan digunakan untuk menampung kategori dari menu. Struktur tabel kategori adalah sebagai berikut:

**Tabel 3.46 Struktur tabel kategori**

| No | Nama field | Tipe | Ukuran | Keterangan |
|---|---|---|---|---|
| 1 | id_kategori | integer | 11 | *Primary key |
| 2 | Nm_kategori | varchar | 30 | |

#### 3.4.15.2.5 Tabel Penjualan

Tabel ini adalah perancangan untuk tabel penjualan yang akan digunakan untuk menampung data penjualan pesanan. Struktur tabel penjualan adalah sebagai berikut:

**Tabel 3.47 Struktur tabel penjualan**

| No | Nama field | Tipe | Ukuran | Keterangan |
|---|---|---|---|---|
| 1 | no_transaksi | char | 13 | *Primary key |
| 2 | id_user | integer | 11 | **Foreign key, user (id_user) |
| 3 | no_pesanan | char | 13 | |
| 4 | tgl_transaksi | datetime | | |
| 5 | keterangan | varchar | 50 | |
| 6 | bayar | decimal | 15,2 | |

#### 3.4.15.2.6 Tabel Meja

Tabel ini adalah perancangan untuk tabel meja yang akan digunakan untuk menampung data meja. Struktur tabel meja adalah sebagai berikut:

**Tabel 3.48 Struktur tabel meja**

| No | Nama field | Tipe | Ukuran | Keterangan |
|---|---|---|---|---|
| 1 | id_meja | integer | 11 | *Primary key |
| 2 | no_meja | smallint | 6 | |

#### 3.4.15.2.7 Tabel Status

Tabel ini adalah perancangan untuk tabel status yang akan digunakan untuk menampung data status pesanan. Struktur tabel status adalah sebagai berikut:

**Tabel 3.49 Struktur tabel status**

| No | Nama field | Tipe | Ukuran | Keterangan |
|---|---|---|---|---|
| 1 | id_status | integer | 11 | *Primary key |
| 2 | nm_status | varchar | 45 | |
| 3 | ic_status | varchar | 45 | |

### 3.4.15.3 Perancangan Kode

#### 3.4.15.3.1 Id Transaksi

Format id pemesanan : TRN YYMMDD XXXX

TRN : Transaksi

YYMMDD : Tahun, bulan, dan tanggal pemesanan

XXXX : Nomor urut pesanan

#### 3.4.15.3.2 Id Pemesanan

Format id pemesanan : PSN YYMMDD XXXX

PSN : Pesanan menu

YYMMDD : Tahun, bulan, dan tanggal pemesanan

XXXX : Nomor urut pesanan

Untuk nomor urut transaksi pada id pemesanan setiap tanggal yang berbeda maka nomor akan di ulang dari 0001 kembali. Misal pada tangal 201205 no terakhir 0007, dan jika pada tanggal selanjutnya no urut transaksi di reset menjadi 0001 kembali. Ini berfungsi jika kita ingin mengetahui berapa transaksi yang telah di lakukan pada tanggal tersebut tanpa harus melakukan *query* lagi.

#### 3.4.15.4 Perancangan Status

##### 3.4.15.4.1 Status Pemesanan

PERSIAPAN : Menyatakan status pemesanan masih antri atau belum di proses oleh bagian dapur, dalam status pemesanan antri.

SEDANG DIBUAT : Menyatakan status pemesanan sedang diproses atau sedang dibuat oleh bagian dapur, pada status ini pelanggan tidak bisa merubah pesanan, hanya bisa menambah jumlah pesanan.

SUDAH DIBUAT : Menyatakan status pemesanan selesai dibuat atau diproses.

PEMBATALAN : Menyatakan status transaksi dibatalkan.

SUDAH BAYAR : Menyatakan status transaksi sudah dibayar, dan no meja akan kembali kosong.

##### 3.4.15.4.2 Status Konfirmasi Pesanan

YA : Transaksi pemesanan telah diterima oleh kasir, maka akan segera dilanjutkan ke proses selanjutnya.

TIDAK : Transaksi tidak diterima oleh kasir maka pesanan tidak akan diproses lebih lanjut ke dapur.