

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Alat dan Bahan**

Dalam penelitian ini dibutuhkan beberapa alat dan bahan sebagai penunjang keberhasilan penelitian. Alat dan bahan tersebut adalah sebagai berikut:

##### **3.1.1 Alat**

Hardware berupa:

- a. Laptop
- b. Perangkat Input dan Output( Keyboard, Mouse, Scanner, dan Printer)

Software berupa:

- a. Sistem Operasi (Windows 8.1)
- b. SQL Server 2014
- c. Visual Studio 2014
- d. Microsoft Visio 2010
- e. Balsamiq

##### **3.1.2 Bahan**

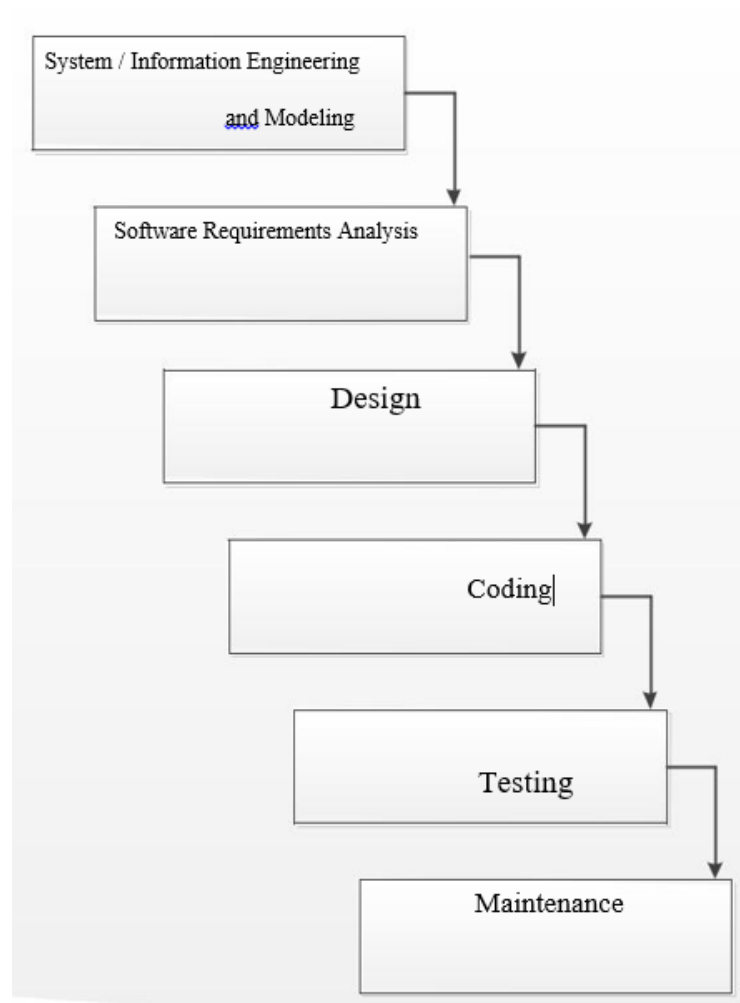
Dalam penelitian ini bahan yang akan digunakan adalah:

- a. Data yang diperoleh melalui studi *literature* berdasarkan penelitian sebelumnya, yang masih memiliki keterkaitan dengan penelitian yang akan dikembangkan. Dari data tersebut dapat diperoleh pemahaman

- b. mengenai kebutuhan atau persyaratan aplikasi yang akan dikembangkan pada penelitian ini.
- c. Data-data yang diperoleh dari proses penjualan dan pembelian.
- d. Hasil wawancara langsung dengan pemilik toko.
- e. Item yang diperjual belikan di toko.
- f. Maka akan dibuat aplikasi dengan spesifikasi sebagai berikut:
  - 1. Login Owner ataupun Karyawan.
  - 2. Input Output Data Penjualan dan Pengembalian.
  - 3. Cek Stok Barang
  - 4. Cek Harga Barang
  - 5. Cek History Penjualan.

### **3.2 Alur Penelitian**

Untuk pengembangan sistem, metode yang digunakan adalah model SDLC (*Software Development Life Cycle*). Model SDLC yang dipakai adalah model *Waterfall*. *Waterfall Model* atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Dengan mengikuti pendapat Roger S.Pressman, tahapan pada model SDLC *Waterfall* dalam pembuatan aplikasi dapat dilihat pada Gambar 3.1



**Gambar 3. 1** Metode *waterfall*

### 3.2.1 System or Information Engineering and Modelling

Langkah pertama adalah mencari kebutuhan dari keseluruhan sistem yang diaplikasikan ke dalam bentuk *software*. Seperti komponen *hardware* yang dibutuhkan dan basis data yang digunakan.

### **3.2.2 *Software Requirement Analysis***

Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*, seperti bagaimana tampilan *user interface* dan kebutuhan fungsi-fungsi yang digunakan dalam aplikasi.

### **3.2.3 *System Design***

*System Design* merupakan proses yang digunakan untuk mengubah kebutuhan-kebutuhan sebelumnya menjadi representasi ke dalam bentuk “*blueprint*” *software* sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya.

### **3.2.4 *Coding***

Untuk dapat dimengerti oleh mesin yang dalam hal ini adalah komputer, maka desain aplikasi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*, dalam hal ini bahasa pemrograman yang digunakan adalah bahasa pemrograman *C#*.

### **3.2.5 *Testing***

Dalam tahap *testing* dilakukan pengujian *software* yang sudah dibuat. Semua fungsi-fungsi *software* harus diuji coba agar *software* bebas dari *error* atau kesalahan dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

### 3.2.6 *Maintenance*

Pemeliharaan dalam aplikasi diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada *error* kecil yang tidak ditemukan sebelumnya atau ada penambahan fitur-fitur yang belum ada pada *software*.

### 3.3 Analisis Prosedur

Prosedur penerimaan barang masuk yang sedang berjalan di Toko Gandrung Elektrik dan Elektronik ada beberapa tahapan antara lain:

1. Pemilik Toko memesan barang-barang yang diperlukan toko.  
Kemudian pemilik toko menentukan apakah barang yang dipesan mau dibayar langsung atau kredit.
2. Tanda tangan nota.  
Setelah itu pemilik toko mencatat harga modal, harga satuan di toko, harga grosir dan eceran dari setiap item yang telah dibeli. Jika konsumen membeli barang 12 atau lebih, maka akan mendapatkan diskon dari toko.
3. Mengirim barang ke toko lain dengan menyiapkan barang yang di order (diluar barang yang masuk ke toko). Lalu membuat nota penjualan keluar disertai tanda tangan dan stempel.
4. Bawa pulang nota asli apabila sistem pembayarannya menggunakan tempo.
5. Kalau penjualannya menggunakan cash maka nota asli dipegang oleh toko tujuan tersebut.

Prosedur input dan output data yang sedang berjalan di Toko Gandrung Elektrik dan elektronik melalui beberapa tahapan antara lain:

1. Admin akan login seebagai owner atau sebagai karyawan
2. Ketika login sebagai owner, fungsi yang bisa dijalankan adalah input penjualan, pengembalian, pembelian, cek history, dan pengelolaan sistem password untuk admin.
3. Ketika login sebagai karyawan maka fungsi yang bisa dijalankan hanya menu penjualan dan pengembalian barang saja.
4. Pada form penjualan, admin dapat mengelola data penjualan, melihat sisa stok barang, dan laporan penjualan harian.
5. Pada form pengembalian admin dapat mengedit barang yang di return berdasarkan alasan konsumen.

### **3.4 Rancangan Perangkat Lunak**

Metode perancangan yang dilakukan untuk mengembangkan perangkat lunak ini adalah dengan menggunakan UML.

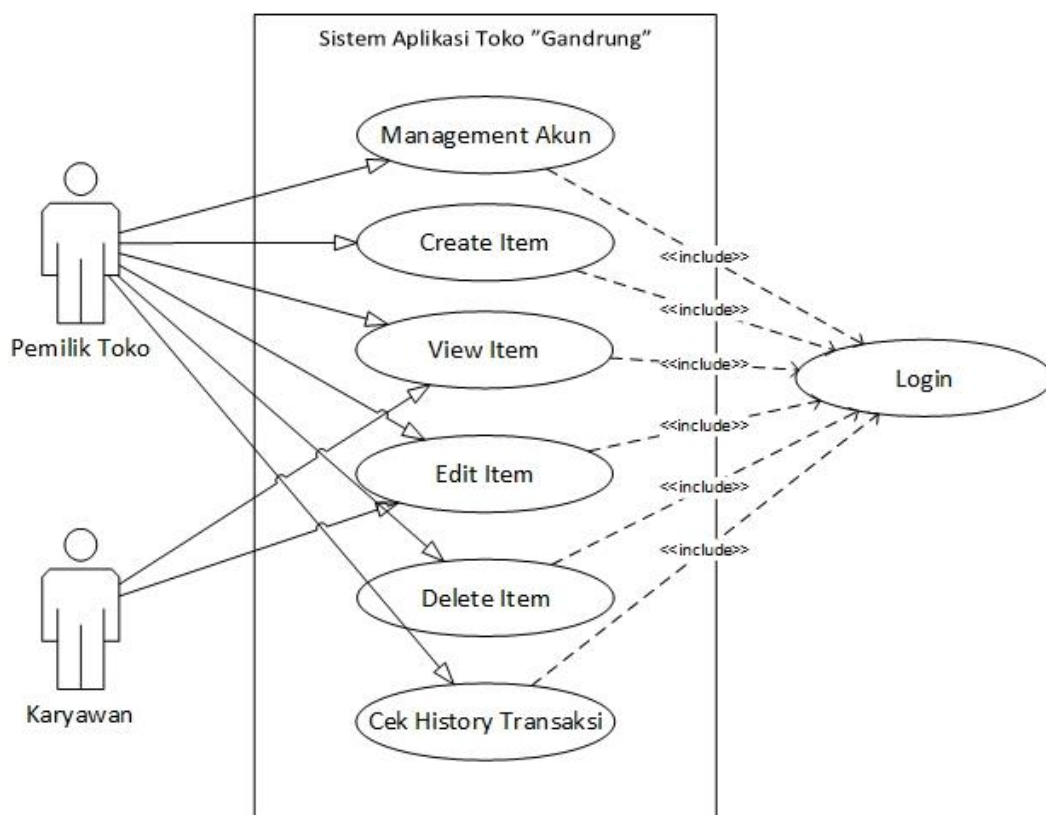
#### **3.4.1 Diagram UML (*Unified Modeling Language*)**

UML adalah sebuah bahasa yang telah menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak (Afif, 2002). Pada penelitian ini, diagram-diagram yang digunakan untuk memodelkan aplikasi berorientasi objek dalam UML meliputi *use case diagram*, *activity diagram* dan *class diagram*.

### 3.4.2 Use Case Diagram

*Use Case diagram* merupakan teknik pemodelan untuk menjelaskan setiap aktifitas yang mungkin terjadi dengan menggunakan sudut pandang dari aktor sebagai pengguna sistem. *Use case diagram* lebih menekankan pada “siapa” melakukan “apa” dalam lingkungan sistem perangkat lunak yang akan dibangun.

*Use Case diagram* untuk aplikasi toko ini dapat dilihat pada Gambar 3.2



**Gambar 3. 2** Use case diagram aplikasi toko

Gambar 3.2 menunjukkan *use case* seorang administrator. Seorang administrator harus melakukan *login* untuk dapat menjalankan fungsinya. Fungsi-fungsi yang dimiliki administrator adalah untuk melakukan input, output, edit item

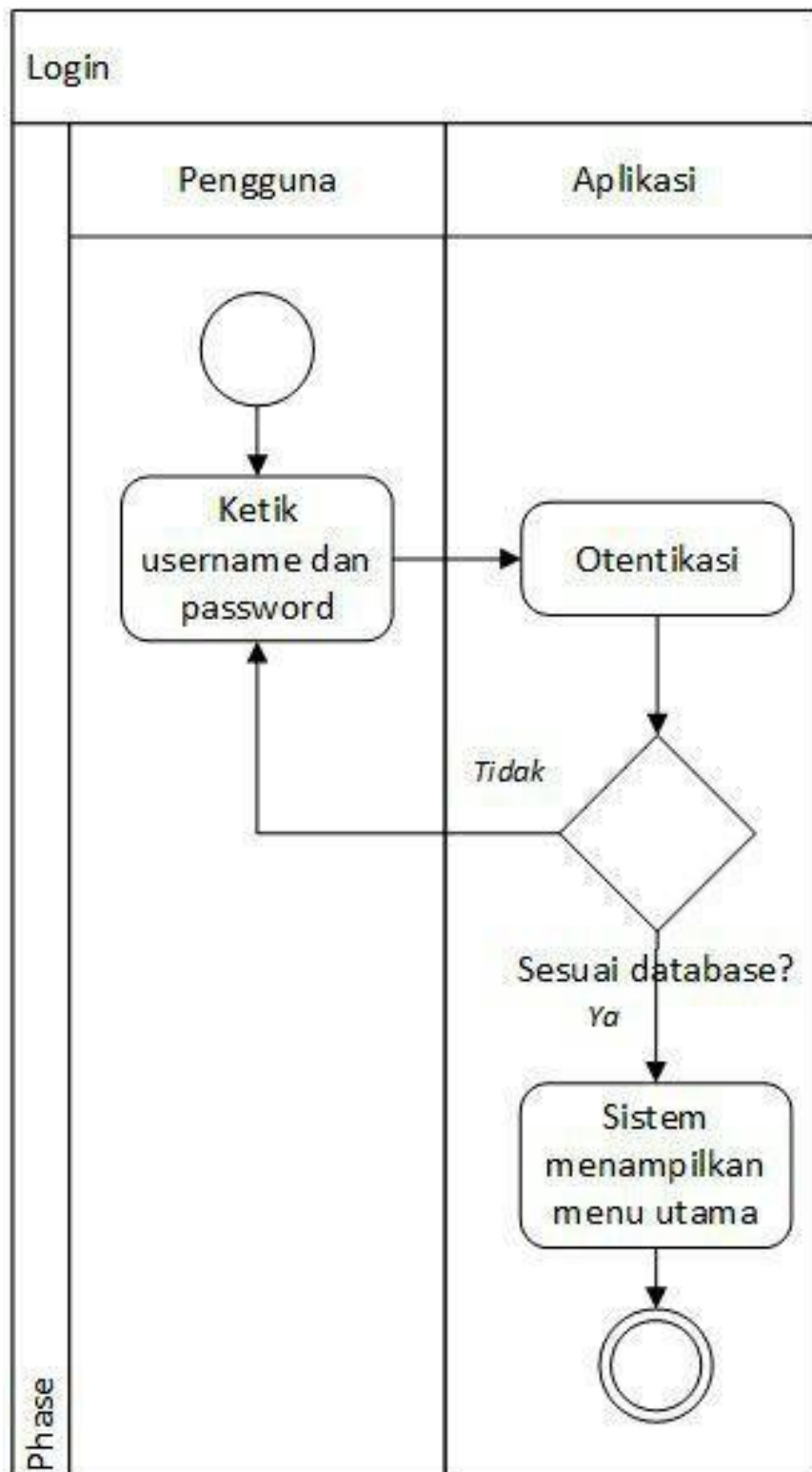
### 3.4.3 *Activity Diagram*

*Activity diagram* menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses parallel yang mungkin terjadi pada beberapa eksekusi.

#### A. *Login*

*Activity diagram* berikut memaparkan bagaimana *admin* melakukan *login* ke sistem seperti pada Gambar 3.3

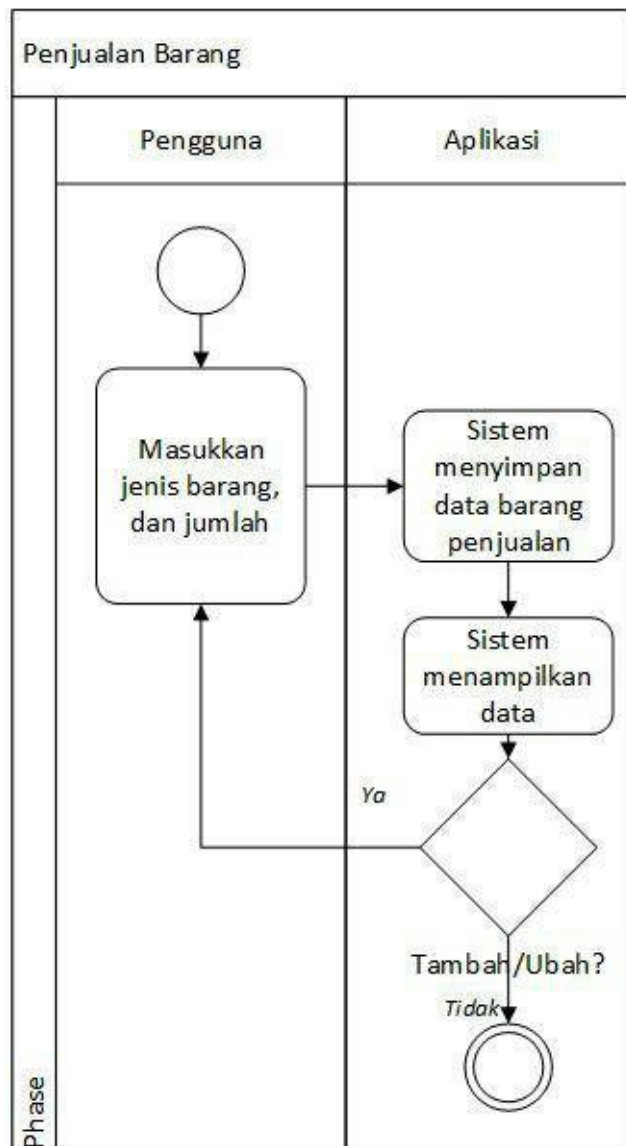




Gambar 3.3 Activity login

## B. Mengelola Data Penjualan Barang

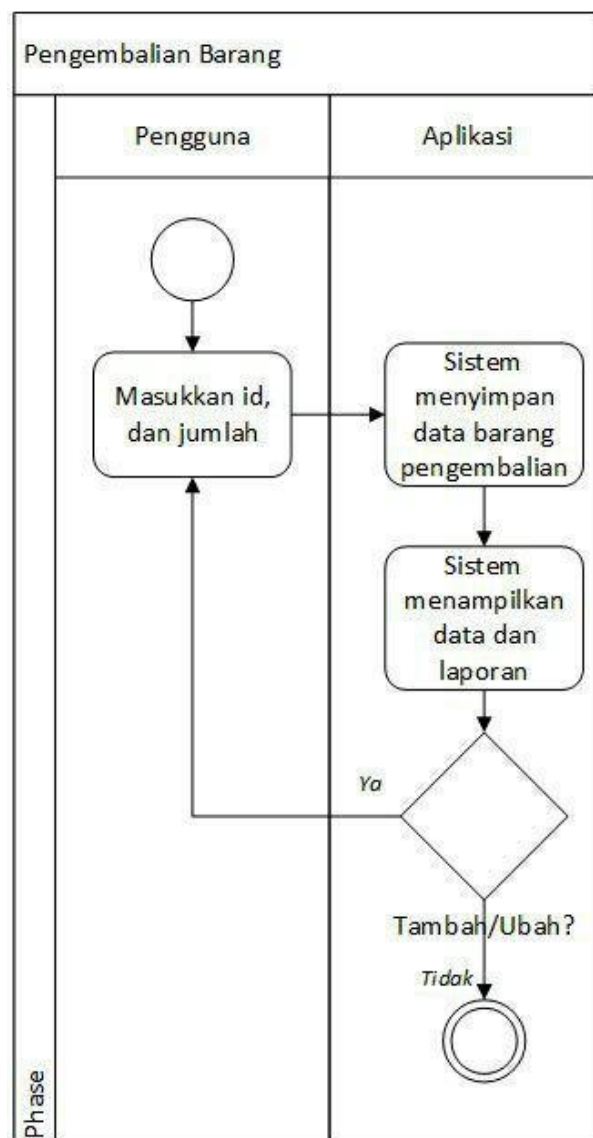
*Activity diagram* mengelola data penjualan barang menggambarkan proses *admin* melakukan penambahan data, mengedit data ataupun melihat data barang yang masuk. *Activity diagram* mengelola data penjualan barang dapat dilihat pada Gambar 3.4



**Gambar 3. 4** *Activity* penjualan barang

### C. Mengelola Data Pengembalian Barang

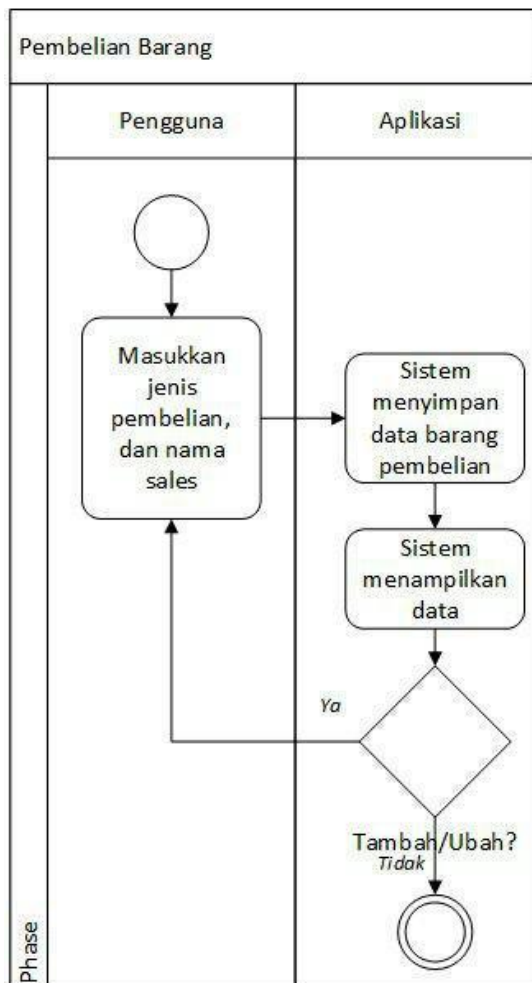
*Activity Diagram* mengelola data pengembalian barang menggambarkan tentang alur kerja dari proses *input* data yang selanjutnya semua akan tersimpan di dalam *database*. *Activity diagram* mengelola data pengembalian barang dapat dilihat pada Gambar 3.5



**Gambar 3.5** Activity pengembalian barang

### D. Mengelola Data Pembelian Barang

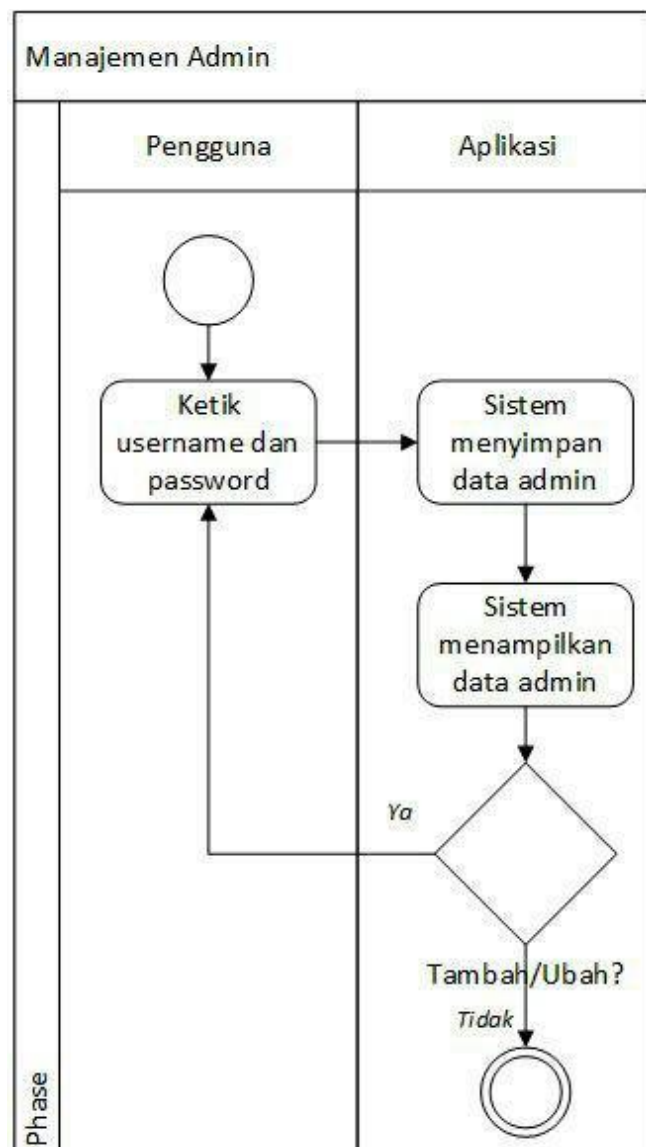
*Activity Diagram* mengelola data pembelian barang menggambarkan tentang alur kerja dari proses input data yang selanjutnya semua akan tersimpan di dalam *database*. *Activity diagram* edit data barang masuk dapat dilihat pada Gambar 3.6



**Gambar 3. 6** Activity pembelian barang

### E. Manajemen Admin

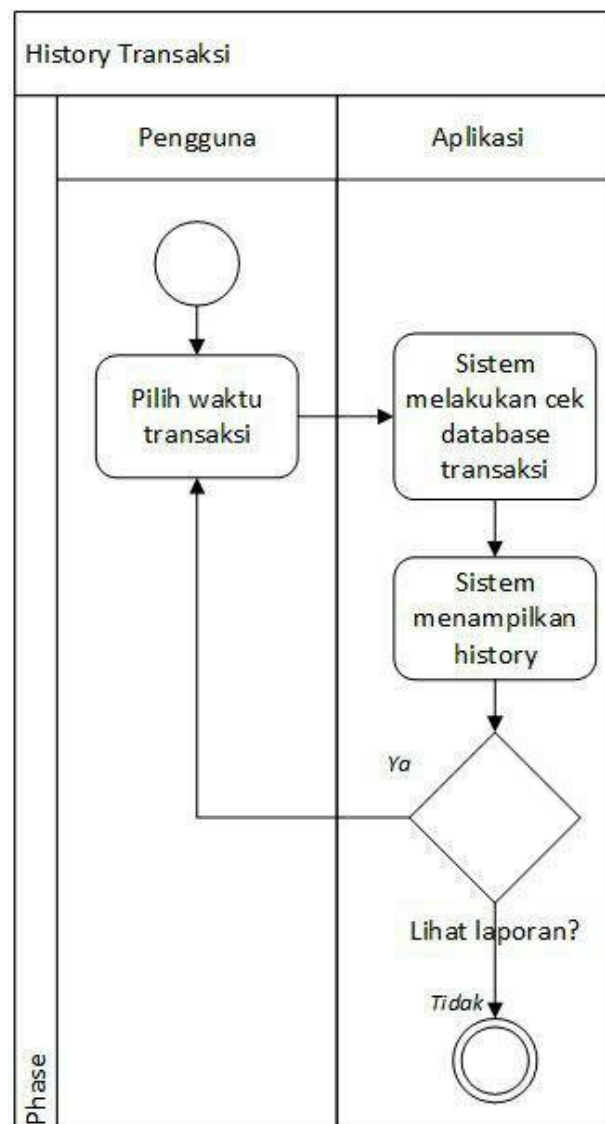
*Activity Diagram* mengelola manajemen admin menggambarkan tentang alur kerja dari proses pembuatan password karyawan baru. Pada sistem ini *admin* dapat membuat, mengedit password karyawan baru. *Activity diagram* Manajemen Admin dapat dilihat pada Gambar 3.7.



**Gambar 3. 7** Activity manajemen admin

## F. Cek History

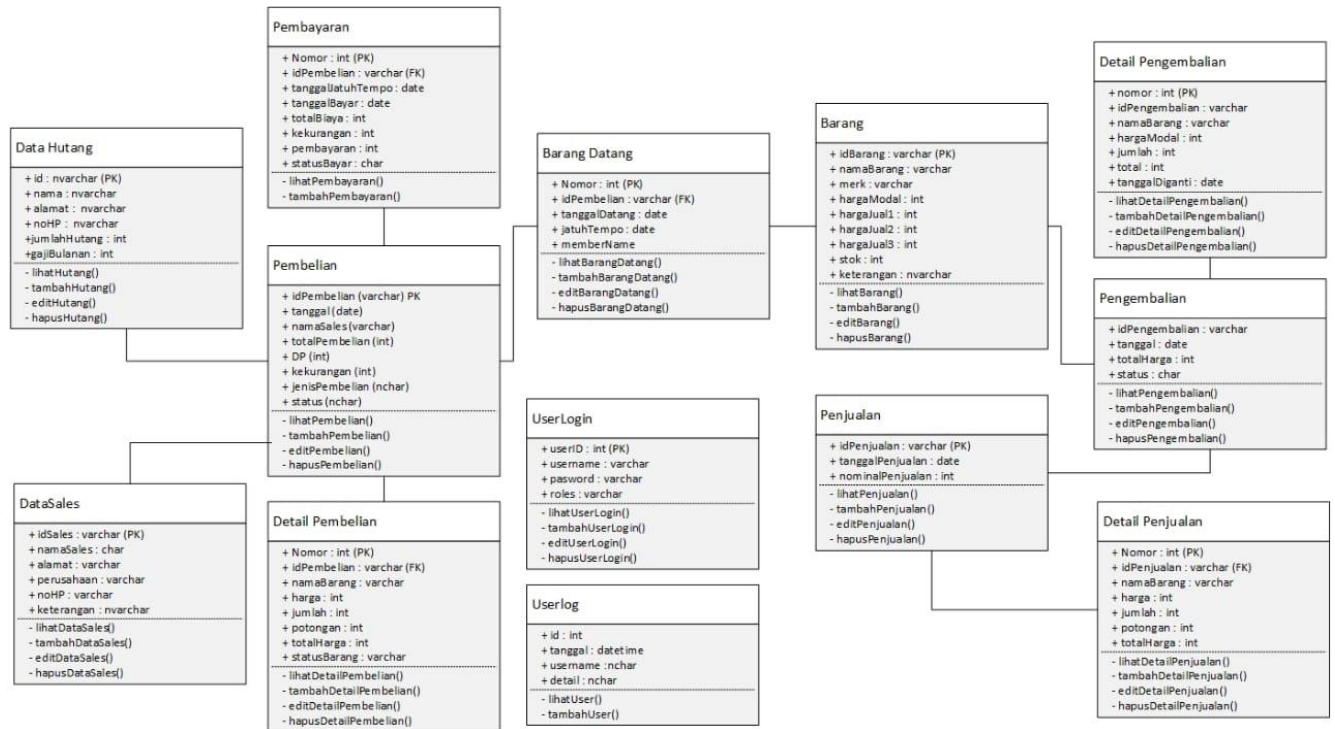
*Admin* mempunyai hak untuk melihat laporan data penjualan dengan yang berlangsung di toko. Lapornya akan ditampilkan di Crystal Report dan *admin* juga dapat mencetak laporannya. *Activity diagram* laporan dapat dilihat pada Gambar 3.8.



**Gambar 3. 8** Activity history

### 3.4.4 Class Diagram

Gambar 3.9 menunjukkan *Class Diagram* yang digunakan dalam aplikasi.



Gambar 3.9 Class diagram

Berikut penjelasan tentang Gambar 3.9

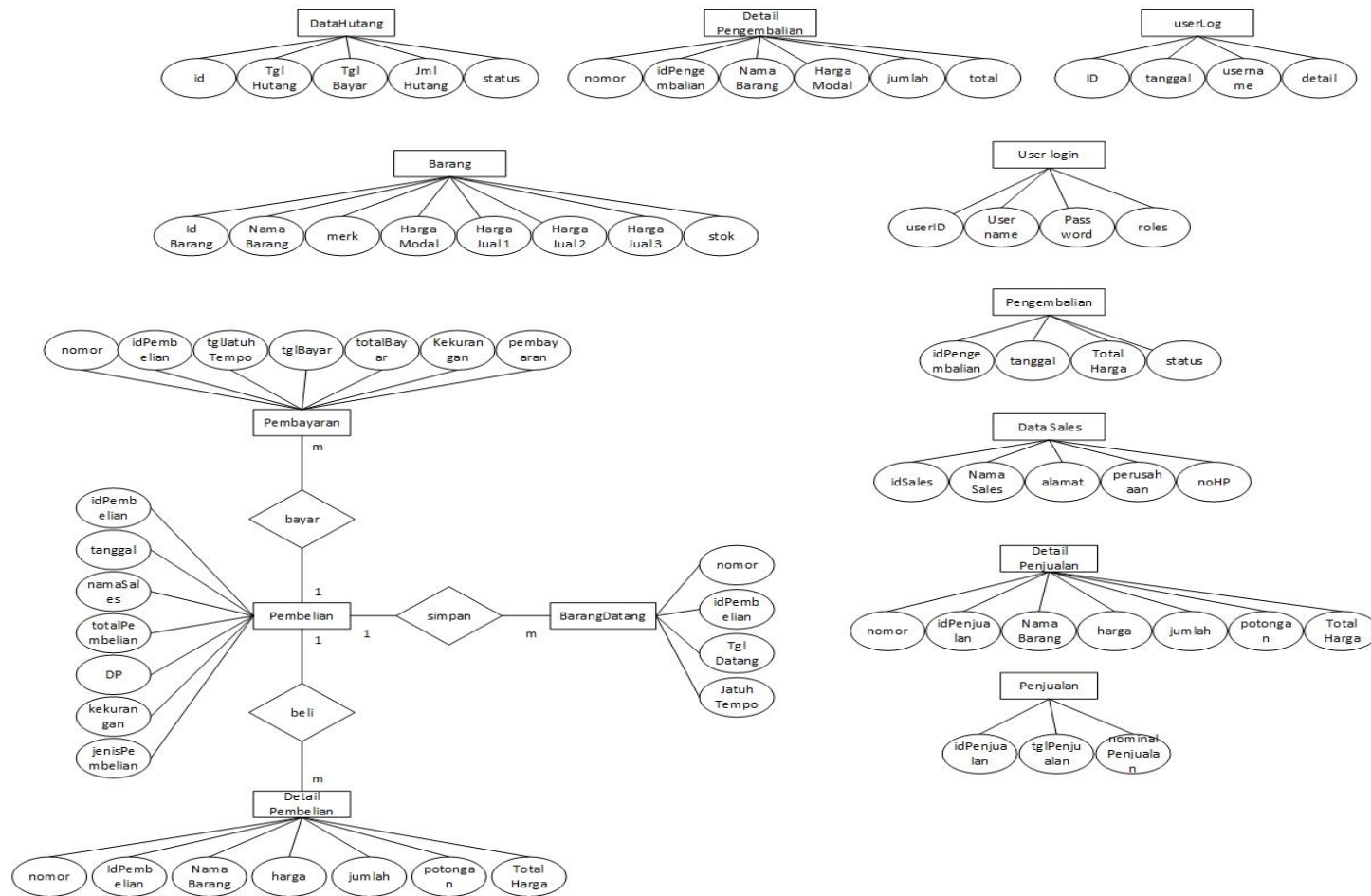
1. Semua kelas pada diagram digunakan untuk inialisasi ke dalam table model dan basis data aplikasi.
2. Kelas Pembayaran memiliki asosiasi dengan kelas pembelian. Minimal satu barang keluar harus memiliki satu klasifikasi.
3. Kelas Barang Datang memiliki asosiasi dengan kelas pembelian. Minimal satu barang masuk harus memiliki satu klasifikasi.
4. Kelas Barang memiliki asosiasi dengan kelas barang datang. Barang yang datang langsung terdata di kelas barang.

5. Kelas Pengembalian memiliki asosiasi dengan kelas barang. Misalkan pengembalian, dia mengambil id dari kelas barang.
6. Kelas Penjualan memiliki asosiasi dengan kelas detail penjualan. Misalkan setiap item yang terjual memiliki data lengkap mengenai status penjualannya.
7. Kelas data Sales memiliki asosiasi dengan kelas Pembelian. Misalkan setiap item yang terbeli terdapat data mengenai sales tempat pengambilan barang.

#### **3.4.5 ER Diagram**

Untuk pengembangan aplikasi pemodelan basis data yang digunakan adalah *ER Diagram*. *ER Diagram* merupakan diagram yang digunakan untuk merancang basis data dan memperlihatkan relasi antar entitas atau objek dengan atributnya. Tujuan *ER Diagram* dalam pengembangan aplikasi adalah dapat memberikan gambaran umum tentang sistem yang dikembangkan sehingga memudahkan dalam merancang basis data yang ditunjukkan pada Gambar 3.10





Gambar 3. 10 ER Diagram aplikasi toko

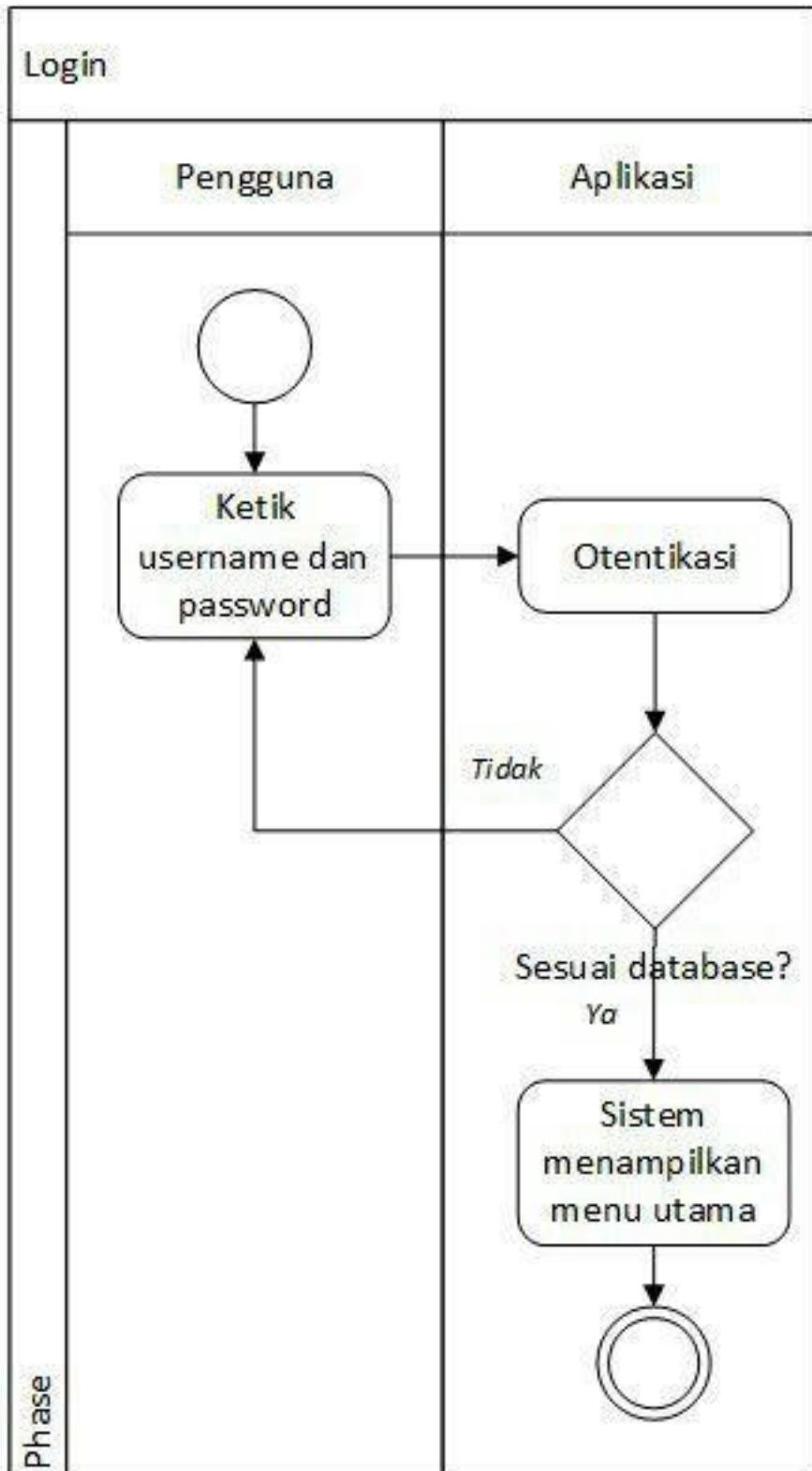


### **3.5 Rancangan Antarmuka (*User Interface*)**

Pada tahap ini dilakukan rancangan bentuk *interface* program aplikasi input output data di Toko Gandrung Elektrik & Elektronik, dengan tujuan memberikan kemudahan dan tidak membingungkan bagi *user* dalam melakukan aktivitasnya. Rancangan *interface* ini adalah gambaran secara umum mengenai tampilan aplikasi.

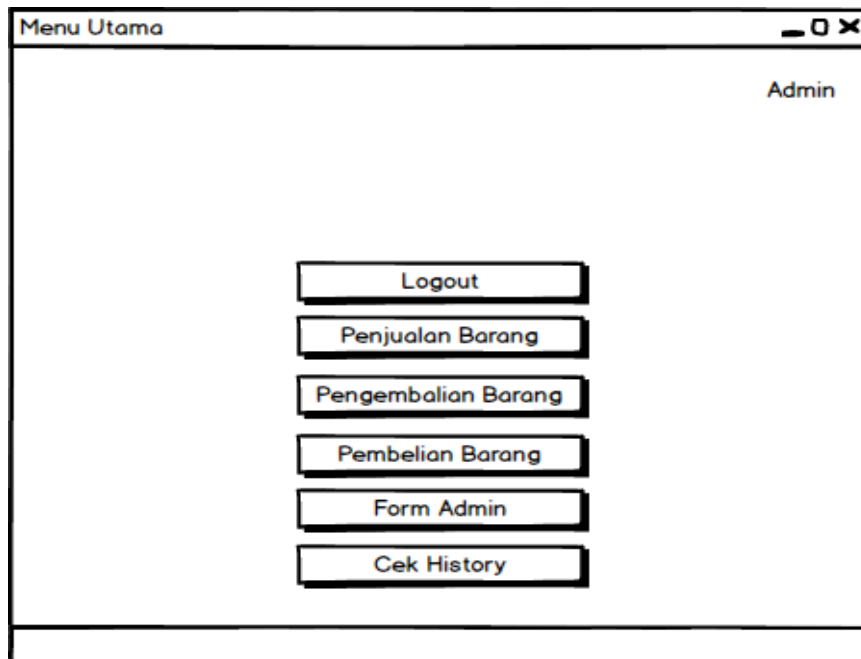
#### **3.5.1 *Form Login***

Rancangan antarmuka ini digunakan oleh *admin* untuk dapat masuk ke *form* utama aplikasi. Gambar rancangan antarmuka *login* dapat dilihat pada gambar 3.11.



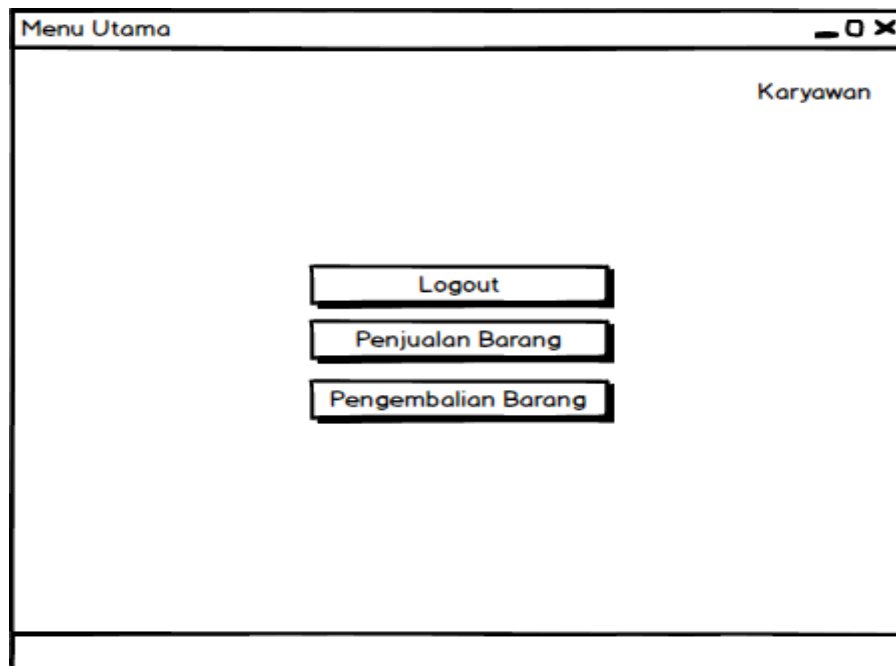
Gambar 3. 11 Rancangan form login

Rancangan antarmuka *form* utama ini merupakan tampilan utama yang dapat dibuka oleh *admin* setelah memasukkan *username* dan *password*. Ada dua fungsi disini, yaitu login sebagai owner dan uga sebagai karyawan. Berikut Gambar rancangan antarmuka *form* utama login sebagai owner dapat dilihat pada Gambar 3.12



**Gambar 3. 12** Rancangan menu utama

Sedangkan yang berikut ini adalah gambar rancangann antarmuka *form* utama login sebagai karyawan dapat dilihat pada Gambar 3.13



**Gambar 3. 13** Rancangan menu utama karyawan

### 3.5.2 Form Penjualan Barang

Rancangan antarmuka *form* penjualan barang merupakan *form* untuk *admin* melakukan aktifitas penginputan data penjualan. Dalam *form* penjualan barang *admin* dapat menambah, mengedit dan menghapus data penjualan. Rancangan antarmuka *form* penjualan barang terdapat tombol tambah barang, edit/hapus, cetak dan hapus layar. Gambar rancangan antarmuka *form* penjualan barang dapat dilihat pada Gambar 3.14

The image shows a software interface window titled "Penjualan". In the top right corner, there are standard window control buttons (minimize, maximize, close). Below the title bar, the text "ID Penjualan" is displayed. The main area contains a table with four columns: "Nama Barang", "Jumlah", "Harga", and "Total". The table has three rows, with the first row shaded. Below the table, there are several interactive elements: a button labeled "Tambah Barang", a button labeled "Edit/Hapus Barang", a label "Sub Total" followed by an empty input field, a button labeled "Cetak", and a button labeled "Hapus Layar".

Gambar 3. 14 Rancangan *form* penjualan

### 3.5.3 Form Pengembalian Barang

Rancangan anatrmuka *form* pengembalin barang merupakan *form* untuk *admin* memasukkan data barang yang di kembalikan oleh konsumen dengan berbagai alasan. *Admin* dapat melihat alasan pengembalian barang di laporan pengembalian. Gambar rancangan antarmuka *form* dapat dilihat pada Gambar 3.15

ID Pengembalian

Nama Barang	Jumlah	Harga	Total

Tambah Barang    Edit/Hapus Barang    Sub Total

Simpan    Hapus Layar

**Gambar 3. 15** Rancangan *form* pengembalian

### 3.5.4 *Form* Pengembalian Langsung

Rancangan antarmuka *form* pengembalian langsung merupakan *form* untuk *admin* memasukkan data pembelian yang direturn oleh konsumen berupa tucker barang dengan yang baru atau dengan yang lebih baik dari sebelumnya. Gambar rancangan antarmuka *form* dapat dilihat pada Gambar 3.16



The image shows a software window titled "Pengembalian - Langsung" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, the text "ID Pengembalian" is located in the top right corner. Below this is a table with four columns: "Nama Barang", "Jumlah", "Harga", and "Total". The table has five rows, with the first row being a header and the others being empty. Below the table, there are four buttons: "Tambah Barang", "Edit/Hapus Barang", "Sub Total" (with an adjacent empty input field), and "Simpan". At the bottom right, there is another button labeled "Hapus Layar".

Nama Barang	Jumlah	Harga	Total

Tambah Barang    Edit/Hapus Barang    Sub Total

Simpan    Hapus Layar

**Gambar 3. 16** Rancangan pengembalian langsung

### 3.5.5 *Form* Pembelian Barang

Rancangan antarmuka *form* pembelian barang merupakan *form* untuk *admin* memasukkan data. Dalam *form* pembelian barang *admin* dapat menambah, mengedit / menghapus data dari item yang dijual ataupun dibeli. Rancangan antarmuka *form* pembelian barang terdapat tombol tambah barang, *Edit/hapus barang*, dan tombol untuk cek harga. Gambar rancangan antarmuka *form* dapat dilihat pada Gambar 3.17

Pengembalian \_ 0 X

ID Pengembalian

Tanggal   Bulan   Tahun

Nama Sales   Jenis Pembelian

Nama Barang	Jumlah	Harga	Total

Total

Total DP

Total Kekurangan

**Gambar 3.17** Rancangan *form* pembelian

### 3.5.6 *Form* Jatuh Tempo Pembelian

Rancangan antarmuka *form* jatuh tempo merupakan *form* untuk *admin* agar bisa melihat tanggal jatuh tempo dari item yang dibeli ke produsen. Gambar rancangan antarmuka *form* dapat dilihat pada Gambar 3.18

Pembelian - Jatuh Tempo

Bulan 1 Tahun 2018

Lihat Tanggal Jatuh Tempo

**Gambar 3. 18** Rancangan pembelian jatuh tempo

### **3.5.7 Form Laporan Pembelian**

Rancangan antarmuka *form* laporan pembelian merupakan *form* yang digunakan oleh admin untuk mengetahui data laporan pembelian setiap item barang. Meliputi jumlah barang yang dibeli dengan harganya, beserta detail pembelian dari setiap item bisa diakses dari form ini. Gambar rancangan antarmuka *form* laporan pembelian dapat dilihat pada Gambar 3.19

Laporan Pembelian

Bulan  Tahun

Total Pembelian  Total Dp

Jumlah Barang yang dibeli  Total Kekurangan

Detail Barang Dibeli:


**Gambar 3. 19** Rancangan laporan pembelian

### 3.5.8 *Form* Pembelian Pembayaran

Rancangan antarmuka *form* pembelian pembayaran merupakan *form* untuk admin dapat melihat laporan pembelian dari setiap item yang dibeli. Admin juga dapat melihat total biaya dan total kekurangan yang harus dilunasi dari setiap item yang dibeli oleh pemilik toko. Gambar rancangan antarmuka *form* pembelian pembayaran dapat dilihat pada Gambar 3.20

Pembelian - Pembayaran

ID Pembelian

Total Biaya

Total Kekurangan

Tanggal   Bulan   Tahun

Tanggal   Bulan   Tahun

Total Pembayaran

**Gambar 3. 20** Rancangan *form* pembelian pembayaran

### 3.5.9 *Form* Barang Datang

Rancangan antarmuka *form* barang datang merupakan *form* unuk input data pembelian barang beserta tanggal pembelian dan tanggal jatuh tempo untuk setiap pembelian yang dibeli oleh pemilik toko. Gambar rancangan antarmuka *form* barang datang dapat dilihat pada Gambar 3.21

Form Pembelian - Barang Datang

ID Pembelian xxx ▼

Tanggal barang datang xxx ▼ Bulan xxx ▼ Tahun xxx ▼

Tanggal jatuh tempo xxx ▼ Bulan xxx ▼ Tahun xxx ▼

Simpan Batal

**Gambar 3. 21** Rancangan form pembelian barang datang

### 3.5.10 *Form Tambah Barang*

Rancangan antarmuka *form* tambah barang merupakan *form* yang berfungsi untuk input data harga dari setiap item. Pada *form* ini admin akan menentukan harga penjualan berdasarkan situasi dn kondisi yang terjadi di lapangan. Gambar rancangan antarmuka *form* tambah barang dapat dilihat pada Gambar 3.22

The image shows a window titled "Tambah Barang" with standard window controls (minimize, maximize, close). Inside the window, there are six input fields arranged vertically:

- ID Barang: A dropdown menu.
- Nama Barang: A dropdown menu.
- Harga Satuan: A text box.
- Jumlah: A text box.
- Diskon/Potongan: A text box.
- Total Harga: A text box.

At the bottom of the form, there are two buttons: "Tambah" and "Batal".

**Gambar 3. 22** Rancangan form tambah barang

### **3.5.11 Form Admin Edit Data Barang**

Rancangan antarmuka *form* edit data barang merupakan *form* yang berfungsi untuk edit data barang di toko. Didalam *form* ini, admin dapat melakukan perubahan harga penjualan dari setiap item barang yang ada di toko. Meliputi edit harga jual 1 / harga satuan toko harga jual 2 = untuk harga retail, dan harga jual 3 = harga per lusin. Gambaran rancangan antarmuka *form* edit *data barang* dapat dilihat pada Gambar 3.23

The image shows a window titled "Admin - Data Barang" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a form with the following elements:

ID Barang	<input type="text"/>	
Nama Barang	<input type="text"/>	<input type="button" value="Edit"/>
Merk	<input type="text"/>	<input type="button" value="Hapus"/>
Harga Modal	<input type="text"/>	<input type="button" value="Simpan"/>
Harga Jual 1	<input type="text"/>	<input type="button" value="Batal"/>
Harga Jual 2	<input type="text"/>	
Harga Jual 3	<input type="text"/>	
Stok	<input type="text"/>	
Keterangan	<input type="text"/>	

**Gambar 3. 23** Rancangan admin data barang

### **3.5.12 Form Admin Tambah Data Barang**

Rancangan antarmuka *form* tambah data barang merupakan *form* yang berfungsi untuk input harga modal, harga jual 1 (Harga Satuan Toko), harga jual 2 (Harga Untuk Retail), harga jual 3 (Harga Eceran). Selain itu juga di form ini admin bisa input jumlah stok dari setiap item yang dibeli. Didalam *form* tambah data barang ini terdapat empat tombol yaitu tombol edit, hapus, simpan, dan batal. Tombol edit untuk mengedit data harga barang ketika ada yang mau diperbaharui. Tombol hapus untuk menghapus data bila ada kekeliruan. Tombol simpan untuk menyimpan perubahan harga, dan tombol batal untuk membatalkan perubahan. Gambaran rancangan antarmuka *form* data barang dapat dilihat pada Gambar 3.24



The image shows a window titled "Admin - Data Barang" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a form with the following fields and controls:

ID Barang	<input type="text"/>
Nama Barang	<input type="text"/>
Merk	<input type="text"/>
Harga Modal	<input type="text"/>
Harga Jual 1	<input type="text"/>
Harga Jual 2	<input type="text"/>
Harga Jual 3	<input type="text"/>
Stok	<input type="text"/>
Keterangan	<input type="text"/>

On the right side of the form, there are two buttons: "Simpan" (Save) and "Batal" (Cancel).

**Gambar 3. 24** Rancangan admin tambah data barang

### 3.5.13 *Form Admin Edit User*

Rancangan antarmuka *form* admin edit user merupakan *form* yang berfungsi untuk mengedit nama pengguna sebelumnya. Edit disini mencakup eit nama, ganti password lama dengan yang baru. Didalam *form* edit user terdapat empat tombol yaitu tombol edit untuk mengubah data yang sudah ada, tombol hapus untuk menghapus perubahan, tombol simpan untuk menyimpan perubahan data user, dan tombol batal untuk membatalkan perubahan.

Gambaran rancangan antarmuka *form* edit user dapat dilihat pada gambar 3.25

The image shows a window titled "Admin - Edit User" with a standard window control bar (minimize, maximize, close). Inside the window, there is a label "Username" followed by a rectangular input field. Below the input field, there are four buttons arranged horizontally: "Edit", "Hapus", "Simpan", and "Batal".

**Gambar 3. 25** Rancangan *admin edit user*

#### **3.5.14 Form Admin Tambah User**

Rancangan antarmuka *form* tambah user merupakan *form* yang berfungsi untuk sebagai tempat input data beserta password untuk login sebagai karyawan ataupun sebagai owner. Didalam *form* tambah user terdapat dua tombol yaitu tombol simpan untuk menyimpan perubahan data dan tombol batal untuk membatalkan perubahan. Gambaran rancangan antarmuka *form* edit *password* dapat dilihat pada gambar 3.26

The image shows a window titled "Admin - Tambah User" with a standard window control bar (minimize, maximize, close). Inside the window, there is a form with the following elements:

- A label "Username" followed by a text input field.
- A label "Password" followed by a text input field.
- A label "Confirm Password" followed by a text input field.
- A label "Role" followed by a dropdown menu.
- Two buttons at the bottom: "Simpan" (Save) and "Batal" (Cancel).

**Gambar 3. 126** Rancangan *form* admin tambah user

### 3.6 Metode Pengujian

Metode pengujian yang dipakai adalah *Black box*. *Black box testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian *black box* ini mengevaluasi hanya dari tampilan luarnya (*interface*-nya), fungsionalitasnya. Tanpa mengetahui apa sesungguhnya yang terjadi dalam proses detailnya (hanya mengetahui *input* dan *output*).