# BAB IV

# ANALISA DAN PERANCANGAN

## 4.1 Analisa Sistem

Analisa Sistem merupakan tahap awal dalam perancangan dan pengembangan sebuah sistem yang akan dirancang, karena pada tahap ini akan diukur dan di evaluasi tentang kinerja dari sistem yang dirancang. Dalam melakukan analisa sistem terlebih dahulu harus mengetahui dan memahami sistem, untuk menganalisa sistem diperlukan data dari sistem untuk dianalisa. Data yang diperlukan adalah hal - hal yang dibutuhkan untuk defenisi data. Sehingga dari data-data yang telah didefenisi dapat dilakukan identifikasi atas masalah-masalah yang ada dan membuat langkah-langkah perancangan yang dibutuhkan sehingga hasil rancangan sesuai dengan yang diharapkan.

Merencanakan suatu perancangan terhadap sistem penjualan yang akan dibangun diharapkan dapat meminimalisasi kesalahan-kesalahan yang terjadi. Untuk itu perlu dilakukan analisa sistem tentang bagaimana prosedur aliran sistem informasi datanya.

### 4.1.1 Analisa Sistem Lama

Sistem penjualan dan stok barang pada Toko Raffa Bangunan yang masih mengandalkan pencatatan buku menunjukkan beberapa poin penting yang perlu diperhatikan. Pada umumnya, sistem manual ini memiliki beberapa kelemahan yang dapat mempengaruhi efisiensi dan akurasi *operasional* toko. Pencatatan buku cenderung memakan waktu dan membutuhkan upaya manusia yang *signifikan*. Proses manual ini dapat menyebabkan keterlambatan dalam mencatat transaksi, yang pada gilirannya dapat mempengaruhi keakuratan stok barang. Kesalahan manusia juga dapat terjadi selama proses pencatatan, meningkatkan risiko ketidakakuratan data. Ketidakakuratan stok barang merupakan masalah serius yang dapat mengakibatkan ketidak seimbangan antara penawaran dan permintaan. Pelanggan dapat mengalami kekecewaan jika produk yang mereka cari tidak tersedia, sementara stok barang yang sebenarnya mungkin masih ada tetapi tidak tercatat dengan benar.

Selain itu, sistem manual ini seringkali sulit untuk memberikan informasi *real-time* tentang penjualan dan popularitas barang. Analisis tren penjualan dan kebutuhan stok dapat menjadi tugas yang sulit dilakukan secara efektif dengan menggunakan metode manual. Kesulitan ini dapat menghambat kemampuan toko untuk *merespons* dengan cepat terhadap perubahan dalam permintaan pasar atau tren konsumen. Kesimpulannya, meskipun pencatatan buku dapat menjadi metode yang sederhana, namun memiliki keterbatasan yang *signifikan* terutama dalam hal kecepatan, akurasi, dan analisis data secara menyeluruh. Seiring dengan perkembangan teknologi, pertimbangan untuk beralih ke sistem yang lebih otomatis dan terkomputerisasi perlu dipertimbangkan guna meningkatkan efisiensi dan daya saing Toko Rafa Bangunan di pasar.

### 4.1.2 Analisa Sistem Baru

Sistem baru penjualan dan stok barang pada Toko Raffa Bangunan yang memanfaatkan aplikasi *Java NetBeans* membawa berbagai keunggulan dan peningkatan dibandingkan dengan sistem manual sebelumnya. Berikut adalah beberapa poin penting terkait sistem baru ini:

Pertama, aplikasi *Java NetBeans* menyediakan otomatisasi yang *signifikan* dalam pencatatan transaksi penjualan dan pemantauan stok barang. Dengan menggunakan teknologi ini, proses *input* data dapat dilakukan lebih cepat dan akurat, mengurangi risiko kesalahan manusia dalam pencatatan. Sistem ini juga memungkinkan integrasi dengan perangkat keras lainnya, seperti mesin kasir dan barcode scanner, untuk meningkatkan efisiensi *operasional*.

Selain itu, aplikasi *Java NetBeans* memungkinkan pembuatan laporan penjualan dan stok barang secara *real-time*. Hal ini mempermudah manajemen untuk memantau kinerja toko secara keseluruhan dan *merespons* dengan cepat terhadap perubahan dalam permintaan pasar. Analisis tren penjualan dan popularitas barang dapat diakses dengan lebih mudah, membantu pengambilan keputusan yang lebih tepat waktu. Keamanan data juga menjadi fokus dalam sistem baru ini. Aplikasi *Java NetBeans* memiliki kemampuan untuk mengimplementasikan kontrol akses, memastikan bahwa informasi sensitif terkait penjualan dan stok barang hanya dapat diakses oleh pihak yang berwenang. Ini memberikan perlindungan tambahan terhadap potensi kebocoran data atau penggunaan yang tidak sah.

Sistem ini juga mendukung manajemen persediaan yang lebih efisien. Dengan pemantauan stok barang secara *real-time*, Toko Raffa Bangunan dapat mengelola persediaan dengan lebih baik, menghindari kekurangan atau kelebihan stok yang dapat menghambat kinerja *operasional*. Dengan menerapkan aplikasi *Java NetBeans*, Toko Raffa Bangunan dapat meningkatkan produktivitas, akurasi, dan *responsivitasnya* dalam mengelola penjualan dan stok barang.

## 4.2 Perancangan Sistem Dan Aplikasi

Perancangan sistem dan aplikasi untuk Toko Raffa Bangunan akan melibatkan pengembangan *platform* berbasis *Java NetBeans* yang terintegrasi. Aplikasi ini akan mencakup modul penjualan untuk mencatat transaksi dengan mudah, dan mendukung berbagai opsi pembayaran. Sistem ini juga akan memiliki modul stok barang untuk pemantauan persediaan secara *real-time*, meminimalkan risiko kekurangan atau kelebihan stok. Integrasi dengan mesin kasir dan pelacakan pelanggan akan meningkatkan efisiensi operasional dan kualitas layanan. Keamanan data dan kontrol akses akan menjadi fokus, memastikan informasi sensitif terjaga. Dengan perancangan ini, Toko Raffa Bangunan dapat meningkatkan produktivitas, akurasi, dan *responsivitas* dalam manajemen penjualan dan stok barang.

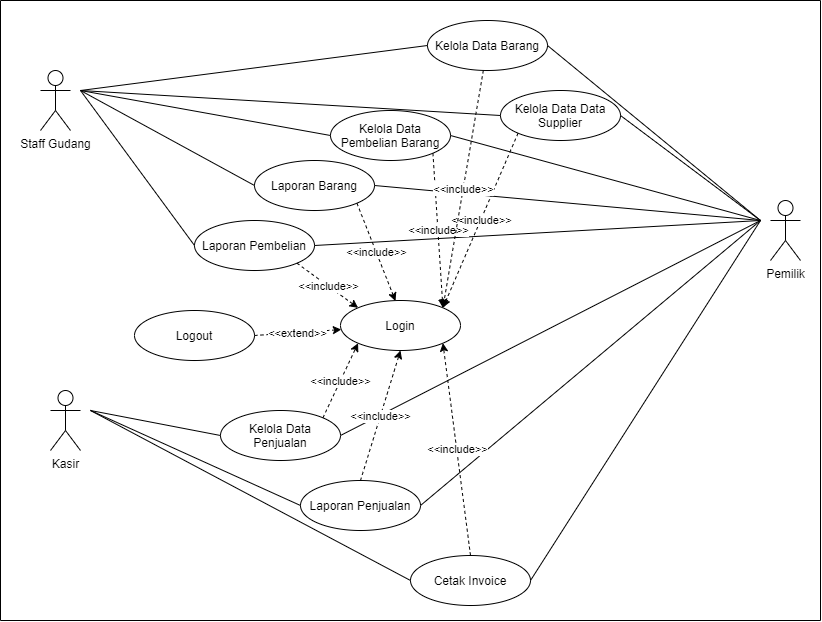
Perancangan *website* ini, dapat kita lihat juga dibawah pembahasan mengenai pemodelan desain sistem secara menyeluruh dengan menggunakan *Unified Modelling Language* *(UML)* yang berguna untuk mempermudah pemasukan data pada *MySQL.*

### 4.2.1 Desain Global

Perancangan aplikasi sistem informasi penjualan ini dirancang dengan mennggunakan alat bantu berupa *UML* (*Unified Modelling Language*) agar mempermudah memindahkan konsep sistem yang dirancang kedalam bentuk program. Dimana perancangannya dalam bentuk diagram sebagai berikut :

#### 4.2.1.1 *Use Case Diagram*

*Use case diagram* menggambarkan bagaimana proses-proses yang akan dilakukan oleh aktor terhadap sebuah sistem. Adapun *use case* diagram dari sistem yang akan di buat dapat dilihat dari gambar di bawah ini.



Gambar 4. 1 *Use Case* Diagram Sistem Informasi Laporan Penjualan Dan Stok Barang Pada Toko Raffa Bangunan

Definisi aktor dan definisi *usecase* dari diagram *usecase* diatas dapat dilihat pada table berikut :

Tabel 4. 1 Tabel *Use Case* Diagram Sistem Informasi Laporan Penjualan Dan Stok Barang Pada Toko Rafa Bangunan

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| 1 | Pemilik | Aktor pemilik adalah user atau pengelola dari pihak rafa bangunan yang bertugas untuk memantau dan mengelola data-data yang ada dalam sistem. |
| 2 | *Staff* Gudang | Aktor *staff* gudang adalah user yang bertugas mengelola data persediaan barang yang ada di dalam sistem. |
| 3 | Kasir | Aktor kasir adalah user yang bertugas mengelola data penjualan pada sistem. |

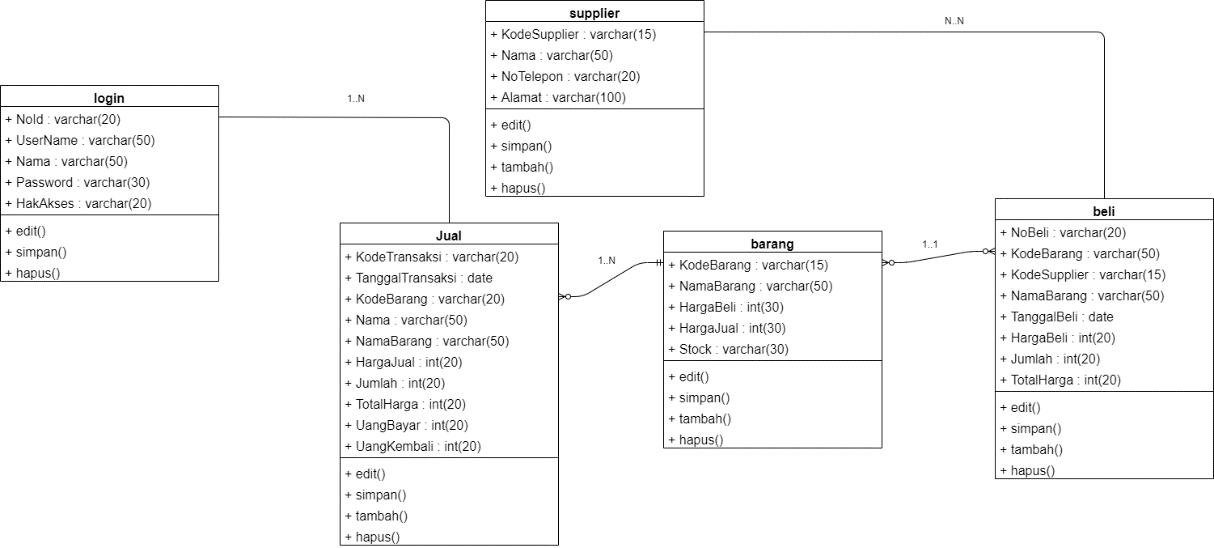
Adapun defenisi dari setiap *use case* diagram tersebut adalah sebagai berikut:

Tabel 4. 2 Defenisi Diagram Pada *Use Case Diagram* Sistem Informasi Laporan Penjualan Dan Stok Barang Pada Toko Raffa Bangunan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | ***Use Case*** | **Deskripsi** | **Aktor** |
| **(1)** | **(2)** | **(3)** | **(4)** |
| 1 | *Login* | Proses masuk ke dalam sistem. | Pemilik, Kasir dan Persediaan |
| 2 | *Logout* | Proses untuk keluar dari sistem. | Pemilik, Kasir dan *Staff* Gudang |
| 3 | Kelola Data Penjualan | Proses mengelola data penjualan pada sistem. | Pemilik dan Kasir |
| **(1)** | **(2)** | **(3)** | **(4)** |
| 4 | Laporan Penjualan | Proses untuk mencetak semua laporan penjualan | Pemilik dan Kasir |
| 5 | Cetak dan Invoice | Proses untuk mencetak data penjualan yang baru diinputkan | Pemilik dan Kasir |
| 6 | Kelola Data Barang | Proses untuk mengelola data barang | *Staff* Gudang dan Pemilik |
| 7 | Kelola Data *Supplier* | Proses untuk mengelola data *supplier* | *Staff* Gudang dan Pemilik |
| 8 | Kelola Data Pembelian Barang | Proses untuk mengelola data pembelian barang | *Staff* Gudang dan Pemilik |
| 9 | Laporan Barang | Proses untuk mencetak data laporan barang | *Staff* Gudang dan Pemilik |
| 10 | Laporan Pembelian | Proses untuk mencetak laporan pembelian | *Staff* Gudang dan Pemilik |

#### 4.2.1.2 *Class Diagram*

*Class Diagram* menampilkan eksistensi atau keberadaan dari kelas-kelas dan hubungan (*relationship*) dalam desain *logikal* dari sebuah sistem. *Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek pengembangan dan desain berorientasi objek.



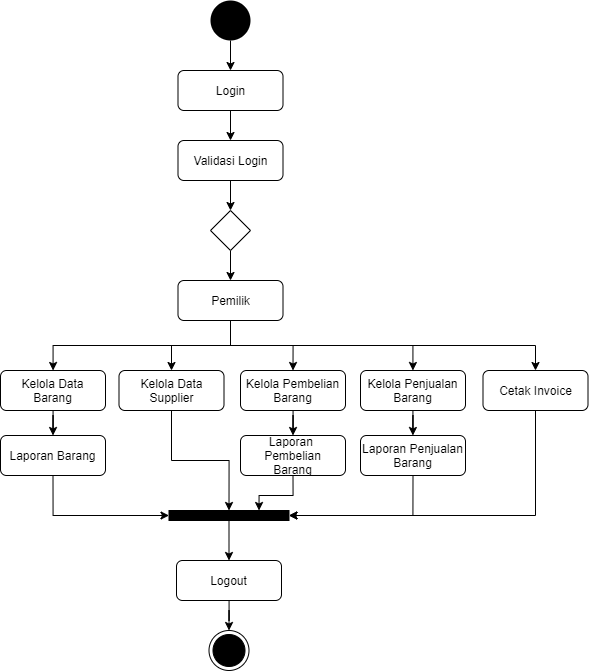
Gambar 4. 2 *Class Diagram* Sistem Informasi Laporan Penjualan Dan Stok Barang Pada Toko Raffa Bangunan

#### 4.2.1.3 *Activity Diagram*

*Activity diagram* pada dasarnya menggambarkan macam-macam aliran aktifitas yang akan dirancang dalam sebuah sistem. Dimana masing-masing diagram memiliki awal, keputusan yang mungkin terjadi pada sistem, dan akhir dalam sistem tersebut. *Activity diagram* pada dasarnya memiliki struktur yang hampir mirip dengan *flowchart* atau diagram alir dalam perancangan sistem secara terstruktur. *Activity diagram* ini dibuat berdasarkan sebuah *use case* atau beberapa *use case* dalam *use case* diagram. Adapun *activity diagram* sebagai berikut:

1. *Activity Diagram* Pemilik

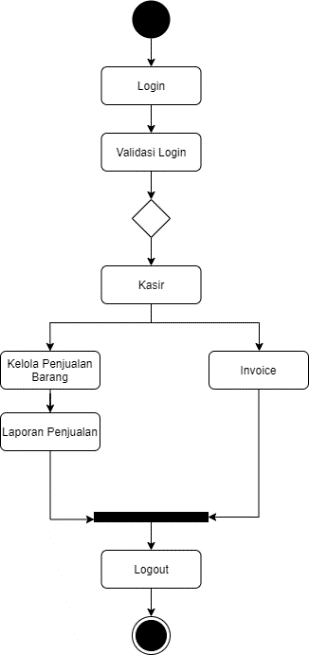
Adapun *Activity Diagram* pemilik pada ini dapat digambarkan seperti Gambar 4.3.



Gambar 4. 3 *Activity Diagram* Pemilik Pada Sistem Informasi Laporan Penjualan Dan Stok Barang Pada Toko Raffa Bangunan

1. *Activity Diagram* Kasir

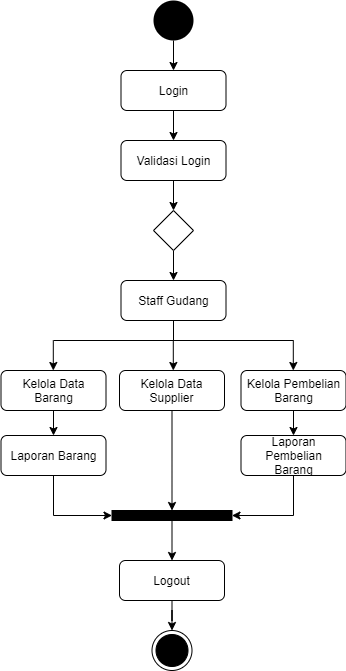
Adapun *Activity Diagram* kasir pada ini dapat digambarkan seperti Gambar 4.4.



Gambar 4. 4 *Activity Diagram* Kasir Pada Sistem Informasi Laporan Penjualan Dan Stok Barang Pada Toko Rafa Bangunan

1. *Activity Diagram* *Staff* Gudang

Adapun *Activity Diagram* persediaan pada ini dapat digambarkan seperti Gambar 4.5.



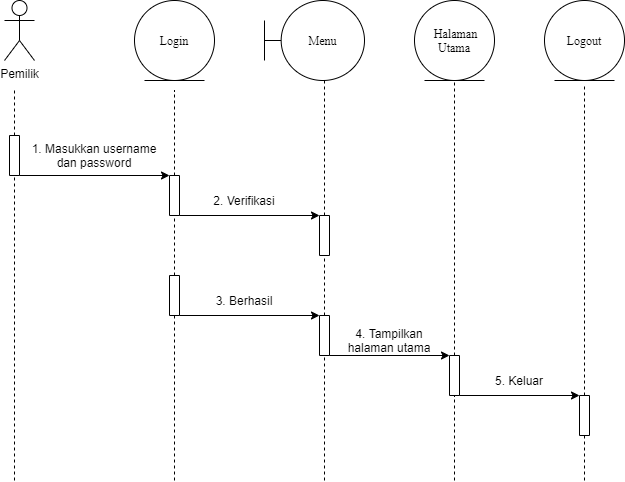
Gambar 4. 5 *Activity Diagram* Persediaan Pada Sistem Informasi Laporan Penjualan Dan Stok Barang Pada Toko Rafa Bangunan

#### 4.2.1.4 *Sequence Diagram*

*Sequence* *diagrams* merupakan diagram yang menggambarkan kelakuan objek pada sistem. Adapun gambaran *sequence diagram* dapat dilihat sebagai berikut:

1. *Sequence* *Diagram Login User*

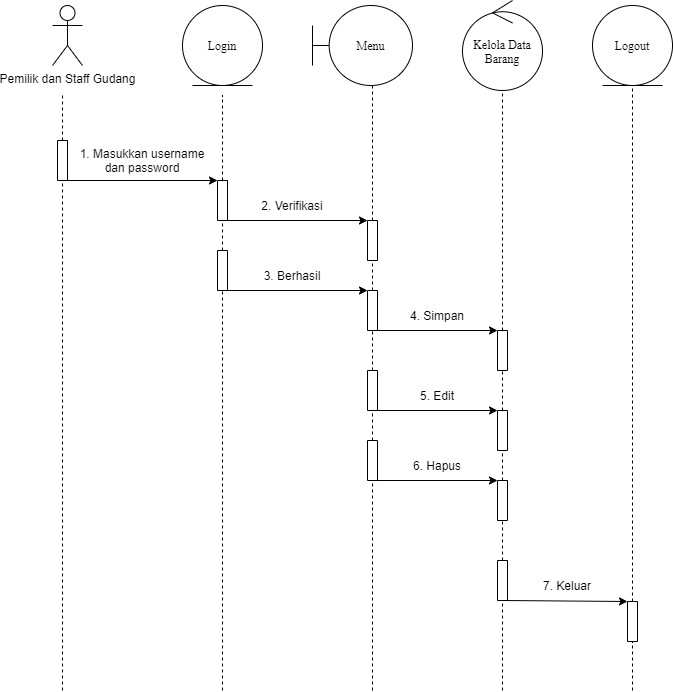
Berikut merupakan *sequence* diagram *login user* yang dapat dilihat pada Gambar 4.6 berikut ini:



Gambar 4. 6 *Sequence Diagram Login User*

1. *Sequence* Diagram Kelola Data Barang

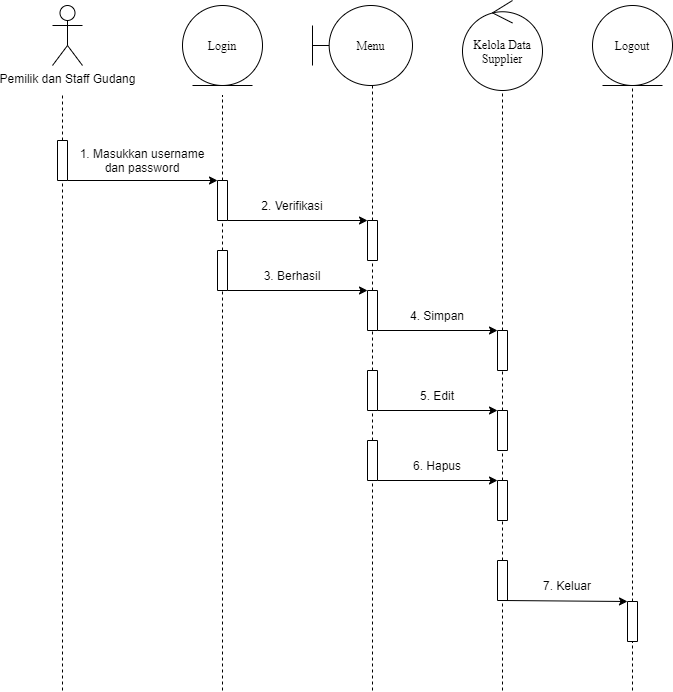
Berikut merupakan *sequence* diagram kelola data barang yang dapat dilihat pada Gambar 4.7 berikut ini:



Gambar 4. 7 *Sequence Diagram* Kelola Data Barang

1. *Sequence* Diagram Kelola Data *Supplier*

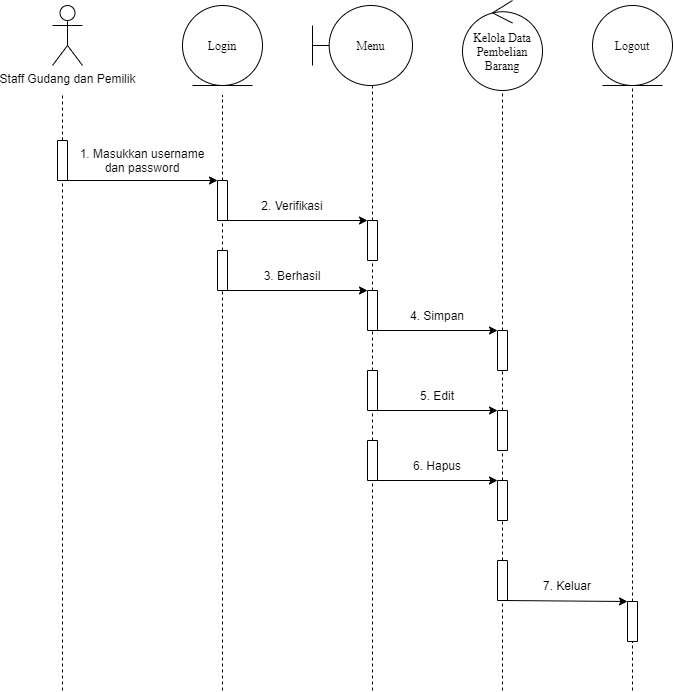
Berikut merupakan *sequence* diagram kelola data *supplier* yang dapat dilihat pada Gambar 4.8 berikut ini:



Gambar 4. 8 *Sequence* Diagram Kelola Data *Supplier*

1. *Sequence* Diagram Kelola Pembelian Barang

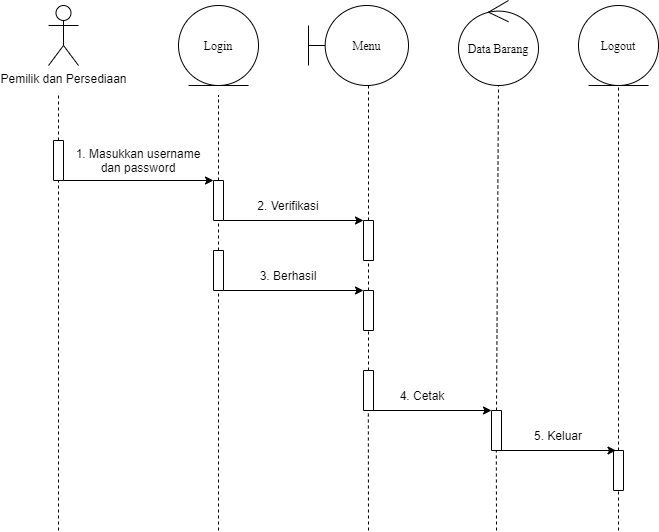
Berikut merupakan *sequence diagram* kelola data pembelian barang yang dapat dilihat pada Gambar 4.9 berikut ini:



Gambar 4. 9 *Sequence Diagram* Kelola Pembelian Barang

1. *Sequence* Diagram Cetak Laporan Barang

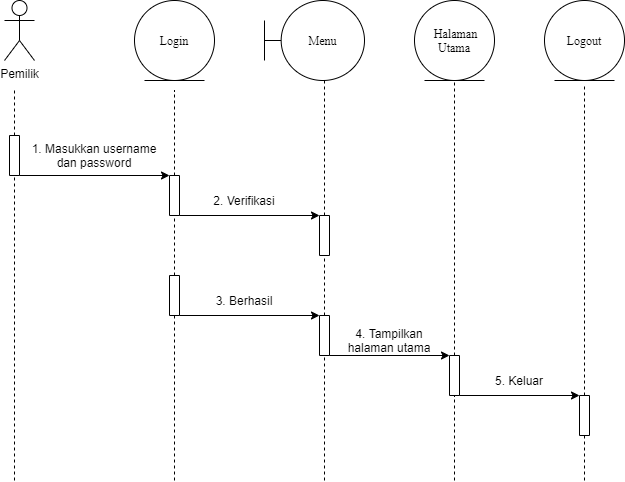
Berikut merupakan *sequence diagram* cetak laporan barang yang dapat dilihat pada Gambar 4.10 berikut ini:



Gambar 4. 10 *Sequence Diagram* Cetak Laporan Barang

1. *Sequence* Diagram Cetak Laporan Pembelian

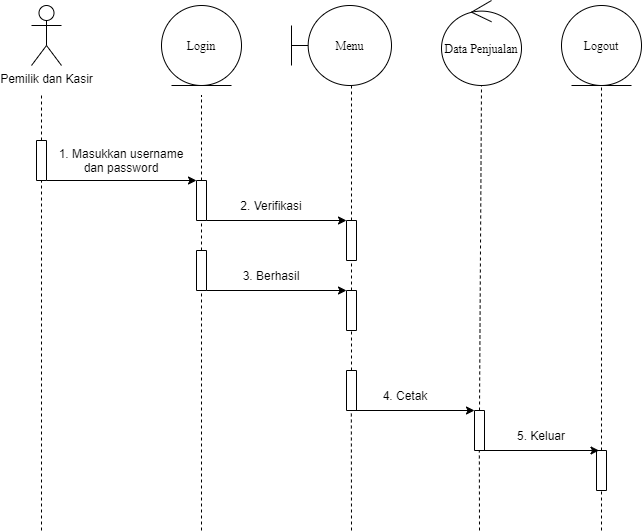
Berikut merupakan *sequence diagram* cetak laporan pembelian yang dapat dilihat pada Gambar 4.11 berikut ini:



Gambar 4. 11 *Sequence Diagram* Cetak Laporan Pembelian

1. *Sequence* Diagram Cetak Laporan Penjualan

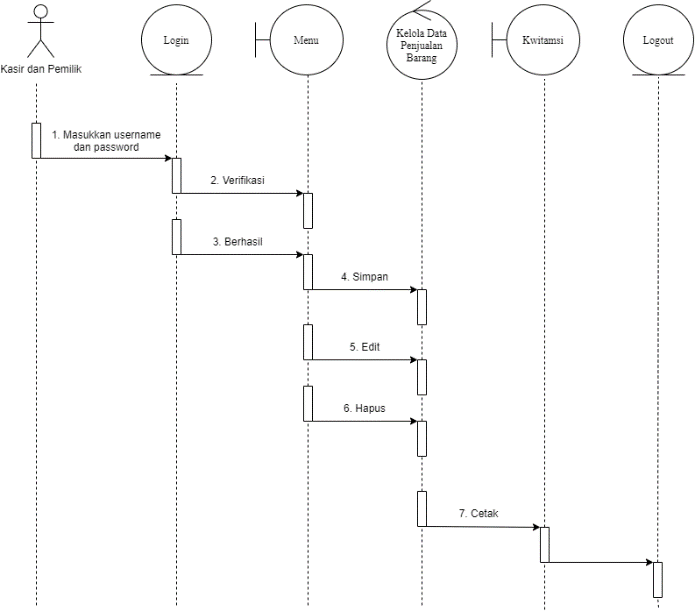
Berikut merupakan *sequence diagram* cetak laporan penjualan yang dapat dilihat pada Gambar 4.12 berikut ini:



Gambar 4. 12 *Sequence Diagram* Cetak Laporan Penjualan

1. *Sequence* Diagram Kelola Data Penjualan

Berikut merupakan *sequence diagram* kelola data penjualan yang dapat dilihat pada Gambar 4.13 berikut ini:



Gambar 4. 13 *Sequence Diagram* Kelola Data Penjualan

### 4.2.2 Desain Terinci

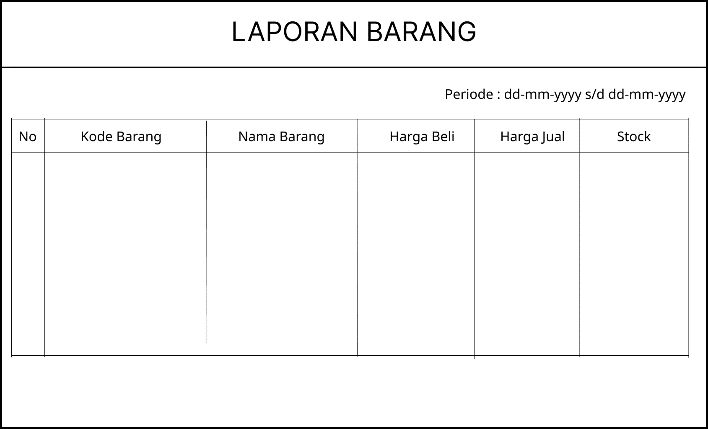
Desain terinci yang dimaksud di sini adalah untuk menjelaskan bentuk-bentuk dari *output* yang dihasilkan, *input* yang dibutuhkan untuk *file-file* yang digunakan dalam sistem penjualan alat dan bahan bangunan.

#### 4.2.2.1 Desain Output

Bagian ini merupakan bentuk-bentuk laporan yang dihasilkan dari data yang *diinputkan*. Adapun bentuk laporan yang akan dihasilkan dapat dilihat pada Gambar di bawah ini, antara lain :

*Form* Laporan Data Barang

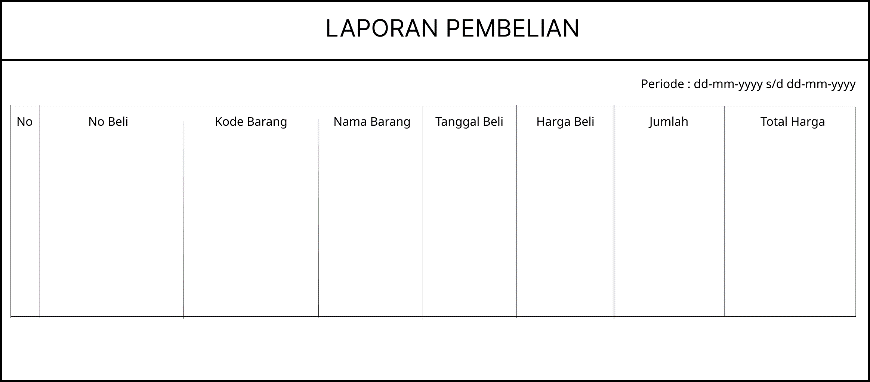
Merupakan desain *form* laporan data barang, desainnya dapat dilihat sebagai berikut pada Gambar 4.14 :



Gambar 4. 14 Desain *Form* Laporan Data Barang

Form Laporan Pembelian

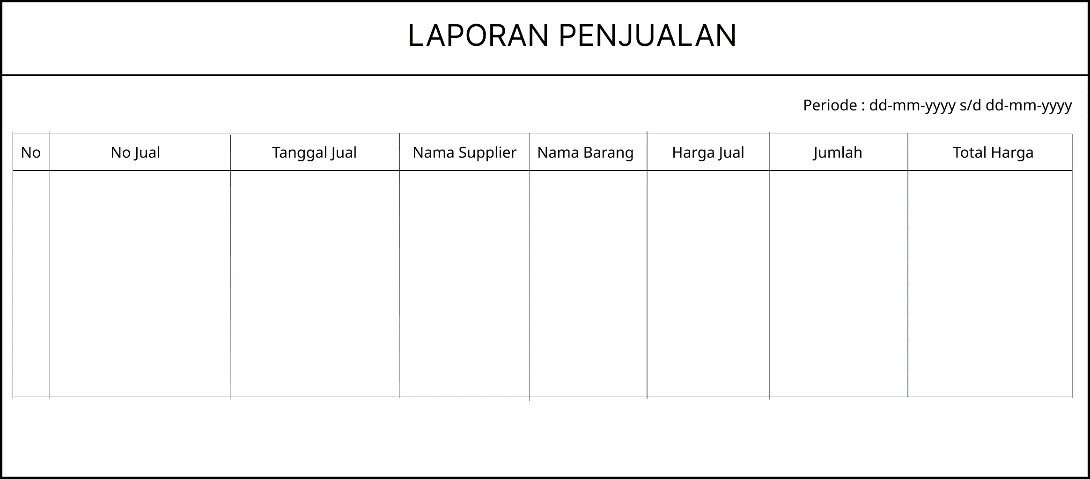
Merupakan desain *form* laporan pembelian, desainnya dapat dilihat sebagai berikut pada Gambar 4.15 :



Gambar 4. 15 Desain *Form* Laporan Pembelian

*Form* Laporan Penjualan

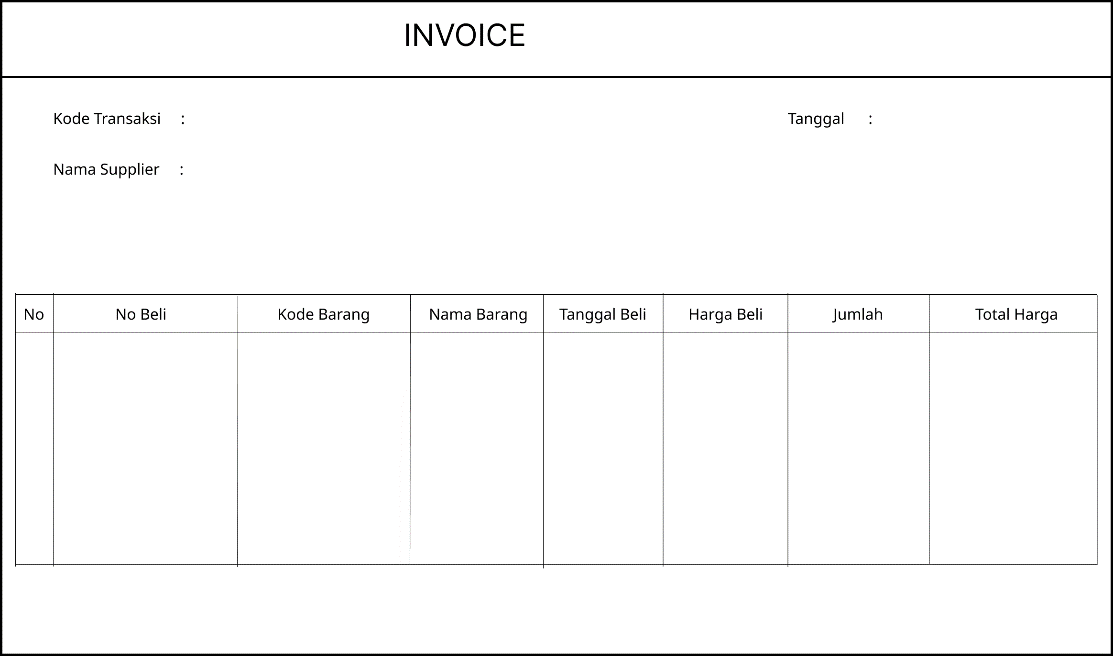
Merupakan desain *form* laporan data penjualan, desainnya dapat dilihat sebagai berikut pada Gambar 4.16 :



Gambar 4. 16 Desain *Form* Laporan Penjualan

*Form* *Invoice*

Merupakan desain *invoice*, desainnya dapat dilihat sebagai berikut pada Gambar 4.17 :



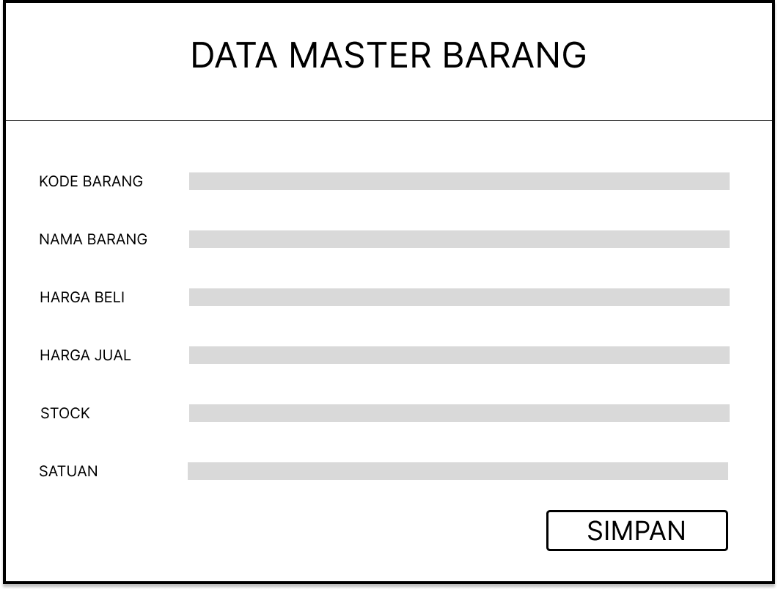
Gambar 4. 17 Desain *Form* *Invoice*

#### 4.2.2.2 Desain *Input*

Berdasarkan dari bentuk *sequence* diagram yang dirancang pada sebelumnya maka input data yang dirancang adalah sebagai berikut:

1. Desain *Form* Tambah Data Barang

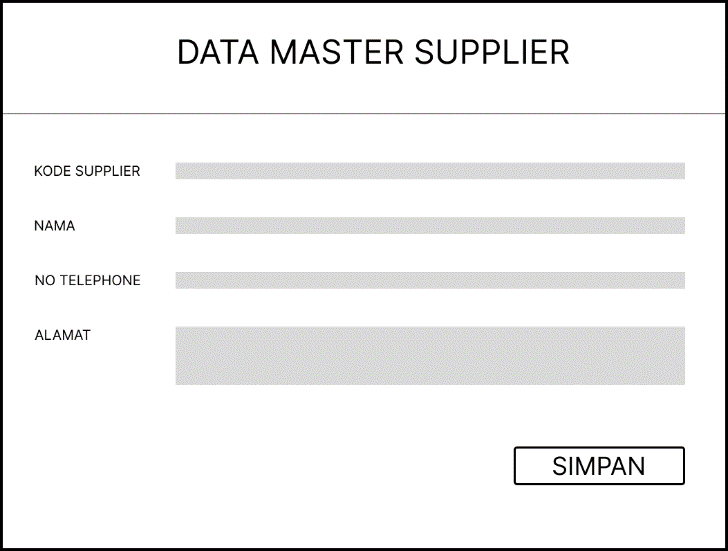
Merupakan desain form tambah data barang, desainnya dapat dilihat sebagai berikut pada Gambar 4.18 :



Gambar 4. 18 Desain *Form* Kelola Data Barang

1. Desain *Form* Tambah Data *Supplier*

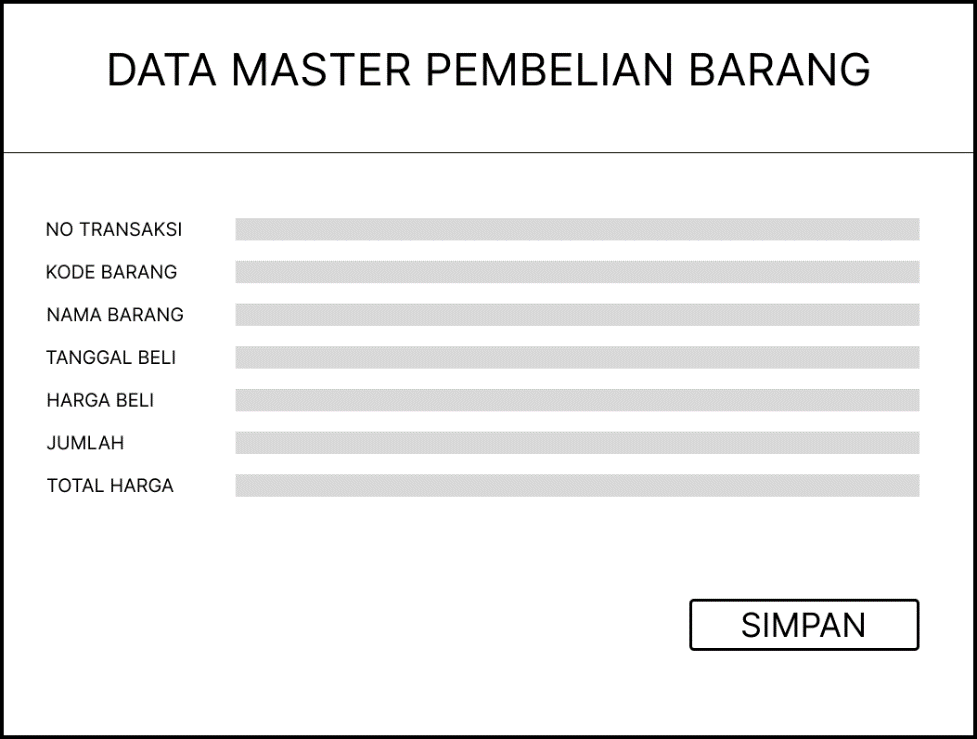
Merupakan desain form tambah data *supplier*, desainnya dapat dilihat sebagai berikut pada Gambar 4.19 :



Gambar 4. 19 Desain *Form* Kelola Data *Supplier*

1. Desain Form Tambah Data Pembelian Barang

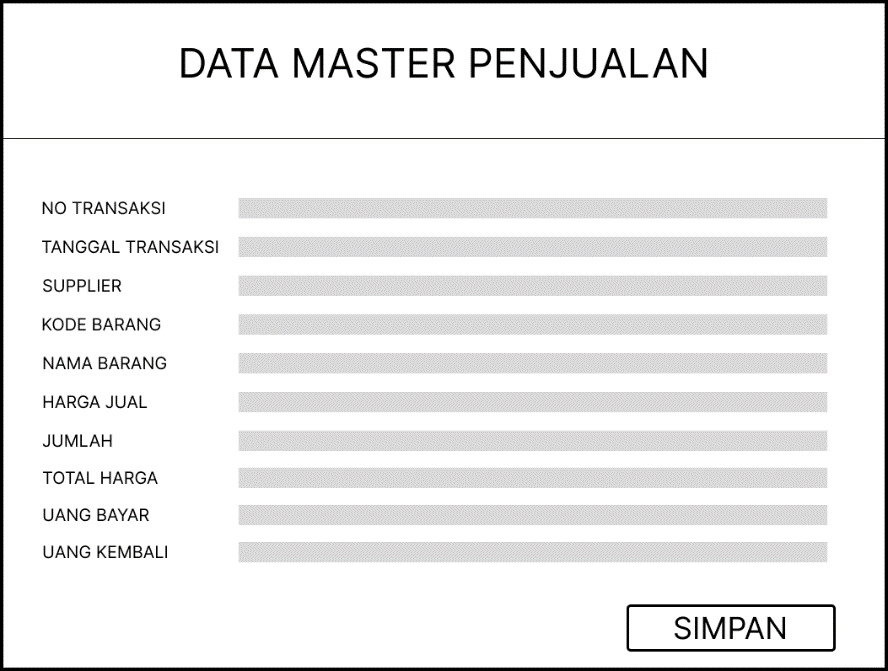
Merupakan desain *form* tambah data pembelian barang, desainnya dapat dilihat sebagai berikut pada Gambar 4.20 :



Gambar 4. 20 Desain *Form* Kelola Data Pembelian Barang

1. Desain Form Kelola Data Penjualan

Merupakan desain *form* kelola data penjualan, desainnya dapat dilihat sebagai berikut pada Gambar 4.21 :



Gambar 4. 21 Desain *Form* Kelola Data Penjualan

#### 4.2.2.3 Desain *File*

*File* adalah kumpulan dari *record* yang tersusun secara logis dimana *record-record* tersebut tersimpan dalam suatu media penyimpanan. Disain *file* yang di rancang pada sistem yang akan dikembangkan ini adalah sebagai berikut:

1. *File* Barang

Nama *database* : penjualan

Nama tabel : barang

*Primary key*  : KodeBarang

Tabel 4. 3 Desain *File* Tabel Barang

|  |  |  |  |
| --- | --- | --- | --- |
| ***No*** | ***Field*** | ***Type*** | ***Description*** |
| 1 | KodeBarang | *Varchar(15)* | *Priamry Key* |
| 2 | NamaBarang | *Varchar(50)* | Nama Barang |
| 3 | HargaBeli | *Int(30)* | Harga Beli |
| 4 | HargaJual | *Int(30)* | Harga Jual |
| 5 | *Stock* | *Varchar(30)* | Stok |
| 6 | Satuan | *Varchar(30)* | Satuan |

1. *File* Beli

Nama *database* : penjualan

Nama tabel : beli

*Primary key*  : NoBeli

Tabel 4. 4 Desain *File* Tabel Beli

|  |  |  |  |
| --- | --- | --- | --- |
| ***No*** | ***Field*** | ***Type*** | ***Description*** |
| **(1)** | **(2)** | **(3)** | **(4)** |
| 1 | NoBeli | *Varchar(20)* | *Primary Key* |
| 2 | KodeBarang | *Varchar(50)* | Kode Barang |
| 3 | NamaBarang | *Varchar(50)* | Nama Barang |
| **(1)** | **(2)** | **(3)** | **(4)** |
| 4 | TanggalBeli | *Date* | Tanggal Beli |
| 5 | HargaBeli | *Int(20)* | Harga Beli |
| 6 | Jumlah | *Int(20)* | Jumlah |
| 7 | TotalHarga | *Int(20)* | Total Harga |

1. *File* Jual

Nama *database* : penjualan

Nama tabel : jual

*Primary key* : KodeTransaksi

Tabel 4. 5 Desain File Tabel Jual

|  |  |  |  |
| --- | --- | --- | --- |
| ***No*** | ***Field*** | ***Type*** | ***Description*** |
| **(1)** | **(2)** | **(3)** | **(4)** |
| 1 | KodeTransaksi | *Varchar(20)* | *Primary Key* |
| 2 | TanggalTransaksi | *Date* | Tanggal Transaksi |
| 3 | Nama | *varchar(50)* | Nama |
| 4 | KodeBarang | *Varchar(20)* | Kode Barang |
| 5 | NamaBarang | *Varchar(50)* | Nama Barang |
| **(1)** | **(2)** | **(3)** | **(4)** |
| 6 | HargaJual | *Int(20)* | Harga Jual |
| 7 | Jumlah | *Int(20)* | Jumlah |
| 8 | TotalHarga | *Int(20)* | Total Harga |
| 9 | UangBayar | *Int(20)* | Uang Bayar |
| 10 | UangKembali | *Int(20)* | Uang Kembali |

1. *File Login*

Nama *database* : penjualan

Nama tabel : *Login*

*Primary key*  : *NoId*

Tabel 4. 6 Desain *File* Tabel *Login*

|  |  |  |  |
| --- | --- | --- | --- |
| ***No*** | ***Field*** | ***Type*** | ***Description*** |
| 1 | *NoId* | *Varchar(20)* | *Primary Key* |
| 2 | *UserName* | *Varchar(50)* | *User Name* |
| 3 | Nama | *Varchar(50)* | Nama |
| 4 | *Password* | *Varchar(30)* | *Password* |
| 5 | HakAkses | *Varchar(20)* | Hak Akses |

1. *File Supplier*

Nama *database* : penjualan

Nama tabel : *Supplier*

*Primary key* : *KodeSupplier*

Tabel 4. 7 Desain *File* Tabel *Supplier*

|  |  |  |  |
| --- | --- | --- | --- |
| ***No*** | ***Field*** | ***Type*** | ***Description*** |
| 1 | KodeSupplier | *Varchar(15)* | *Primary Key* |
| 2 | Nama | *Varcahr(50)* | Nama |
| 3 | NoTelepon | *Varchar(20)* | No. Telepon |
| 4 | Alamat | *Varchar(100)* | Alamat |

# BAB V

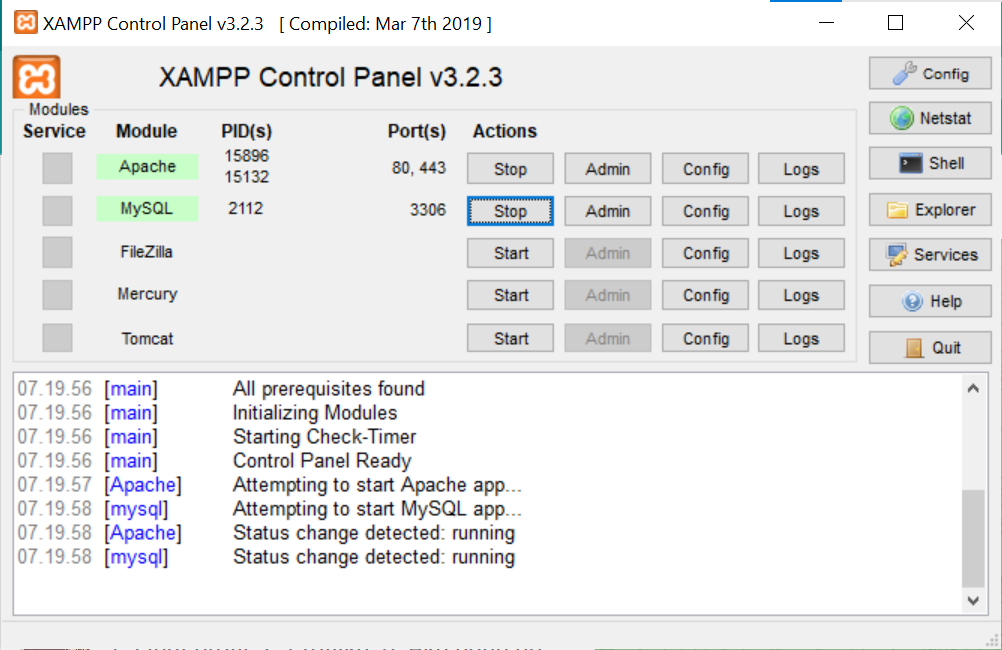
# IMPLEMENTASI DAN PENGUJIAN

## 5.1 Implementasi

Implementasi merupakan tahap penelitian yang dilakukan untuk membuktikan langsung hasil dari analisis yang bertujuan untuk menguji kebenaran proses.

### 5.1.1 *Instalisasi Software*

1. Buka *software XAMPP* setup *Installer* dengan cara klik 2 kali, setelah itu akan muncul *window* yang menjadi awal dari setup.
2. Lalu berikan tanda *Check List* Pada *window Select component*, namun biasanya sudah secara *default* langsung ada tanda *check list*. Kemudian klik tombol *Next*.
3. Selanjutnya akan muncul *window* pada *Select a Folder*, pilih folder tempat *XAMPP* akan di *install*, biasanya folder tempat *install XAMPP* adalah di folder C. setelah menentukan tempat untuk *install*.
4. Setelah itu proses instalasi akan dilakukan secara otomatis oleh komputer anda.
5. Setelah proses *install* selesai maka selanjutnya klik tombol *Finish*.
6. Pada *window* berikutnya pilih tombol *Yes* untuk menampilkan *window XAMPP* *Control Panel*. Jika proses instalasi tadi sudah sesuai dengan aturan yang benar maka akan muncul *window XAMPP* *Control Panel*.
7. Pada XAMPP *Control Panel*, klik *start* pada *Apache* dan *MySQL* untuk menjalankannya seperti Gambar 5.1 berikut ini:



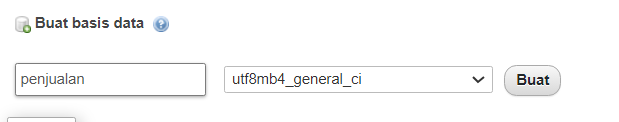
Gambar 5. 1 *XAMPP Control Panel*

Pada Gambar 5.1 di atas, dapat dilihat bahwa adanya beberapa pilihan yang dapat kita akses. Namun, kita hanya perlu menjalankan yang kita butuhkan saja yaitu *Apache dan Mysql*.

### 5.1.2 *Import Database*

Perancangan *database* menggunakan *database MySQL*. Langkah tersebut diantaranya:

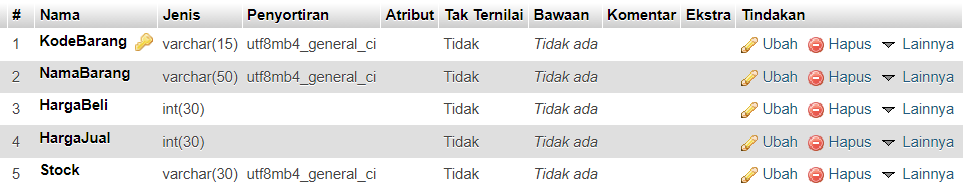
1. Membuat nama *database* pada *text field*, kemudian klik *create*



Gambar 5. 2 Tampilan *Create Database*

1. Tabel Penjualan

Tabel data penjualan berfungsi untuk menyimpan data penjualan.



Gambar 5. 3 Tampilan Tabel Data Penjualan

1. Tabel Beli

Tabel data beli berfungsi untuk menyimpan data beli.



Gambar 5. 4 Tampilan Tabel Data Beli

1. Tabel Jual

Tabel data jual berfungsi untuk menyimpan data jual.



Gambar 5. 5 Tampilan Tabel Data Jual

1. Tabel *Login*

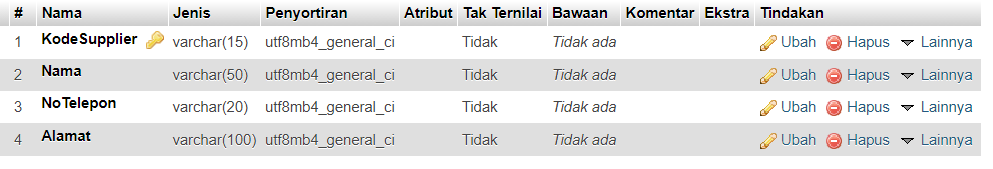
Tabel data *login* berfungsi untuk menyimpan data *login*.



Gambar 5. 6 Tampilan Tabel Data Login

1. Tabel *Supplier*

Tabel data *supplier* berfungsi untuk menyimpan data *supplier*.



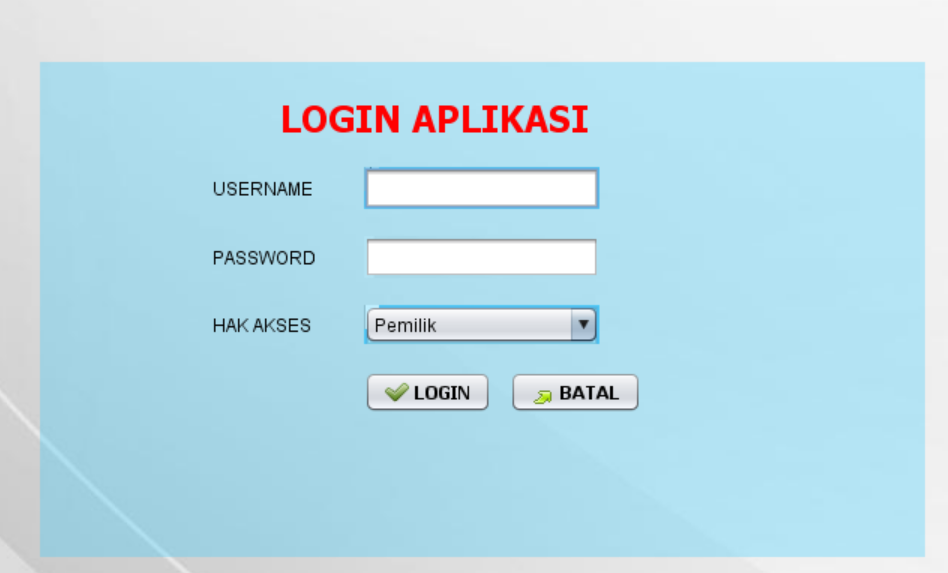
Gambar 5. 7 Tampilan Tabel Data *Supplier*

### 5.1.3 Tampilan Sistem

Pada Halaman ini memuat seluruh hasil dari desain *interface* dari setiap halaman pada sistem ini, seperti sebagai berikut.

1. Tampilan Halaman *Login*

Pada halaman ini *admin* harus memasukan *username* dan *password* yang sudah terdaftar pada *database*.



Gambar 5. 8 Tampilan Halaman *Login*

1. Tampilan Halaman Utama

Pada halaman ini digunakan sebagai tampilan utama sebelum para aktor dalam sistem memproses semua data yang akan terjadi sampai ke laporan.



Gambar 5. 9 Tampilan Halaman Utama

1. Tampilan Halaman Kelola Data Barang

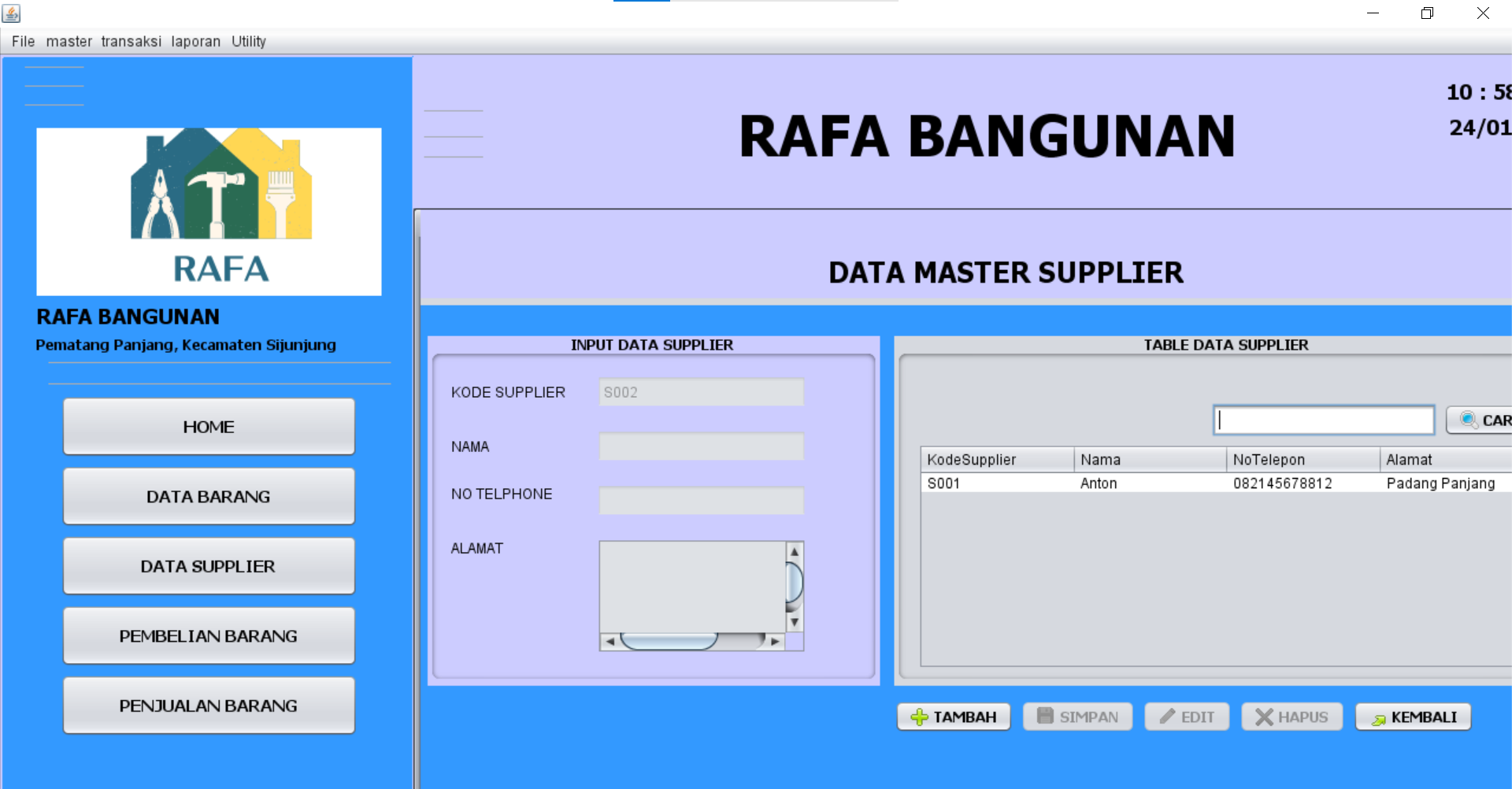
Halaman ini digunakan sebagai halaman untuk mengelola data barang mulai dari tambah, edit dan hapus.



Gambar 5. 10 Tampilan Halaman Kelola Data Barang

1. Tampilan Halaman Kelola Data *Supplier*

Halaman ini digunakan sebagai halaman untuk mengelola data *supplier* mulai dari tambah, edit dan hapus.



Gambar 5. 11 Tampilan Halaman Kelola Data *Supplier*

1. Tampilan Halaman Kelola Data Pembelian

Halaman ini digunakan sebagai halaman untuk mengelola data pembelian mulai dari tambah, edit dan hapus.



Gambar 5. 12 Tampilan Halaman Kelola Data Pembelian

1. Tampilan Halaman Kelola Data Penjualan

Halaman ini digunakan sebagai halaman untuk mengelola data penjualan mulai dari tambah, edit dan hapus.



Gambar 5. 13 Tampilan Halaman Kelola Data Penjualan

1. Tampilan Halaman Laporan Data Barang

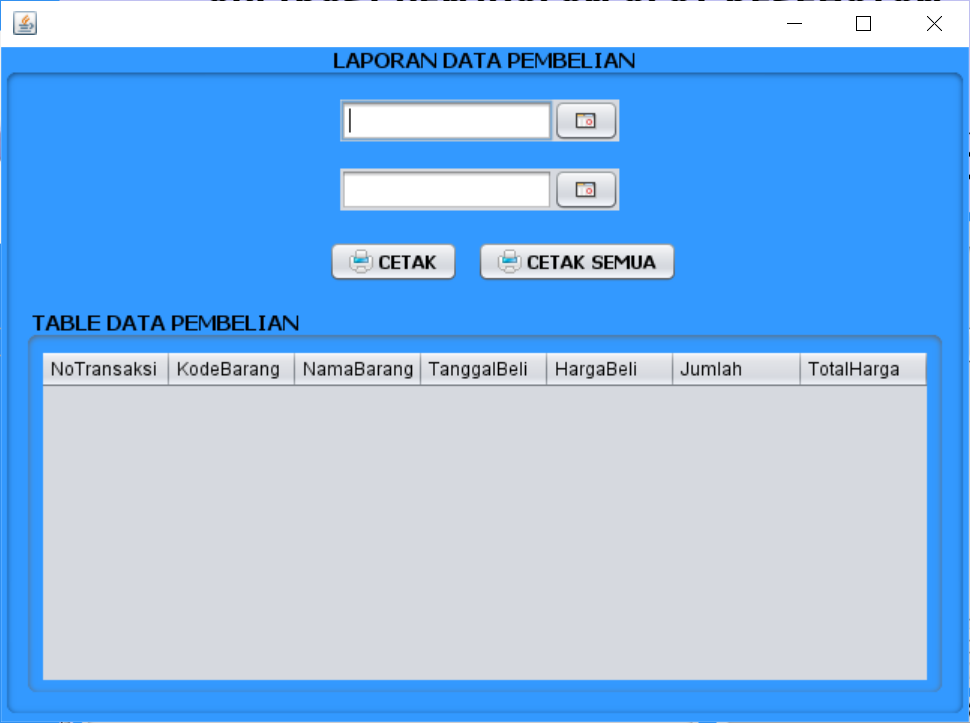
Halaman ini digunakan sebagai halaman untuk laporan data barang



Gambar 5. 14 Tampilan Halaman Laporan Data Barang

1. Tampilan Halaman Laporan Data Pembelian

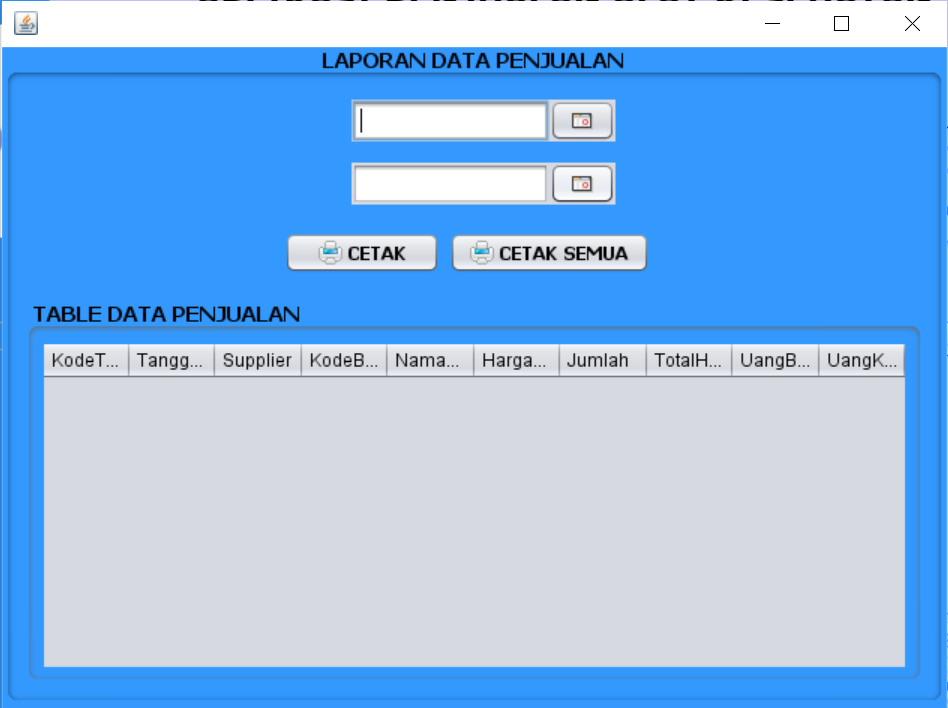
Halaman ini digunakan sebagai halaman untuk laporan data pembelian.



Gambar 5. 15 Tampilan Halaman Laporan Data Pembelian

1. Tampilan Halaman Laporan Data Penjualan

Halaman ini digunakan sebagai halaman untuk laporan data penjualan.



Gambar 5. 16 Tampilan Halaman Laporan Data Penjualan

# BAB VI

# KESIMPULAN DAN SARAN

## 6.1 Kesimpulan

Penulisan penelitian ini mulai dari tahapan analisa permasalahan yang ada hingga pengujian sistem yang baru dirancang maka dapat diambil beberapa kesimpulan, yaitu:

1. Sistem penjualan berbasis *Java NetBeans* pada Toko Raffa Bangunan membawa peningkatan *signifikan* dalam efisiensi *operasional.* Penggunaan teknologi *Java NetBeans* memungkinkan pencatatan transaksi penjualan secara cepat dan akurat, dengan kemampuan integrasi mesin kasir untuk mempercepat proses. Hal ini menghasilkan *operasional* toko yang lebih lancar dan efisien.
2. Sistem ini memberikan keunggulan dalam pemantauan stok barang *secara real-time*. Dengan modul yang terintegrasi, toko dapat dengan mudah melacak ketersediaan barang, menghindari kekurangan atau kelebihan stok. Informasi yang akurat dan *real-time* membantu manajemen membuat keputusan yang lebih cepat dan tepat terkait strategi persediaan.
3. Aplikasi berbasis *Java NetBeans* pada sistem penjualan dan stok barang mendukung peningkatan layanan pelanggan. Integrasi dengan pelacakan pelanggan memungkinkan toko untuk memberikan pengalaman berbelanja yang *personal*, sementara opsi pembayaran yang beragam meningkatkan kenyamanan pelanggan. Dengan demikian, toko dapat membangun hubungan yang lebih kuat dengan pelanggan dan meningkatkan kepuasan pelanggan secara keseluruhan.

## 6.2 Saran

Selain kemampuan aplikasi yang dibahas diatas,sebagai sebuah aplikasi yang baru dikembangkan. penulis merasa masih banyak terdapat berbagai kekurangan. Untuk pengembangan aplikasi ini dikemudian hari ada beberapa hal yang perlu diperhatikan:

1. Disarankan untuk memberikan pelatihan *reguler* kepada staf toko terkait penggunaan sistem penjualan berbasis *Java NetBeans*. Hal ini akan membantu meningkatkan pemahaman mereka terhadap fitur-fitur aplikasi dan meminimalkan kesalahan dalam penggunaan sehari-hari. Selain itu, pemeliharaan rutin terhadap perangkat lunak dan perangkat keras juga penting untuk menjaga kinerja sistem tetap *optimal*.
2. Mempertimbangkan sistem dengan *e-commerce*. Dengan memperluas bisnis *online*, Toko Rafa Bangunan dapat mencapai pasar yang lebih luas. Integrasi ini akan memungkinkan pelanggan untuk melakukan pembelian secara *online*, sementara stok barang dan informasi penjualan tetap terkini di kedua *platform*, meningkatkan *fleksibilitas* dan kemudahan bagi pelanggan.
3. Menggunakan *fitur analitik* dalam sistem untuk memantau data penjualan dan stok barang sangat dianjurkan. Analisis ini dapat memberikan wawasan mendalam tentang tren penjualan, *popularitas* barang, dan perilaku pelanggan. Selain itu, penting untuk mengumpulkan umpan balik pelanggan tentang pengalaman berbelanja mereka. Informasi ini dapat digunakan untuk terus meningkatkan layanan, menyesuaikan persediaan.

Bagian Atas Formulir