**PEMBUATAN APLIKASI RESERVASI HOTEL**

*“diajukan untuk memenuhi salah satu tugas mata kuliah pemrograman VI (Object II)”*

Dosen : Ardhian Ekawijana, S.T., M.T.



Disusun Oleh :

Muhammad Fadillah - 0615101038

Hariyanto – 06151010xx

**TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS WIDYATAMA**

**Bab I**

**Pendahuluan**

**1.1 LATAR BELAKANG**

Pada saat ini perkembangan teknologi semakin maju dengan pesat. Hampir setiap hari inovasi - inovasi baru terlahir terhadap teknologi. Tentunya perkembangan teknologi tersebut akan mempermudah pekerjaan manusia saat ini.

Perkembangan teknologi saat ini juga semakin cepat memasuki berbagai bidang, sehingga kini semakin banyak perusahaan yang berusaha meningkatkan usahanya terutama dalam bidang bisnis yang sangat berkaitan erat dengan teknologi informasi itu sendiri. Hal ini didukung oleh pernyataan bahwa kegunaan computer pada aplikasi bisnis adalah untuk menyediankan informasi dengan cepat dan tepat. Informasi ini ibarat darah yang mengalir di dalam tubuh suatu perusahaan. Jika di dalam suatu perusahaan, informasi tersebut terhenti atau terhambat, maka system perusahaan akan menjadi lusuh (Jogiyanto, 1999:96).

Salah satu perkembangan teknologi informasi yang penting adalah semakin dibutuhkannya penggunaan alat pengolah data yang berfungsi menghasilkan informasi yang dibutuhkan. Perusahaan – perusahaan yang ingin mengembangkan usaha dan mencapai sukses harus mengikuti era informasi dengan menggunakan alat pendukung pengolah data yaitu computer. Hal ini didukung oleh pernyataan yang diutarakan bahwa computer digunakan untuk mengelola sumber daya yang luas dari perusahaan – perusahaan yang memandang seluruh dunia sebagai pasar mereka dimana eksekutif perusahaan melakukan investasi pada teknologi informasi dengan tujuan mencapai skala ekonomis dan dapat mengembangkan produk yang dapat dijual di seluruh dunia (Mcleod, 1998:92).

Dengan adanya komputer sebagai alat pengolah data, maka semua bidang dalam suatu perusahaan ataupun instansi dapat dikomputerisasikan, dalam hal ini bidang – bidang yang dianggap penting dan utama karena hal ini dapat mendukung keberhasilan suatu perusahaan dalam mencapai tujuannya, dalam kajian ini penulis ingin memberikan suatu solusi dengan merancang dan mengaplikasikan suatu penjualan dan stok produk yang manual di bukukan, begitu kurang efektif dan efisien, dan membuat aplikasi penjualan yang terkomputerisasi, user interface untuk mengelola basis data tersebut dan aplikasi penjualan yang terkomputerisasi dengan baik antara system basis data, user interface, dan user itu. Dengan seperti ini akan memberikan solusi optimal yang telah terkomputerisasi, kecepatan dan ketepatan pengolahan data, dan mengurangi tingkat kesalahan pada waktu proses penjualan berlangsung dan dengan mudah mengetahui stok produk yang tersisa.

**1.2 RUMUSAN MASALAH**

Dari latar belakang diatas, penulis dapat menyusun perumusan masalah yang akan dibahas ini yaitu :

* 1. Bagaimana aplikasi ini dapat membantu proses penjualan?
  2. Bagaimana aplikasi ini dapat membantu dalam melihat stock produk yang tersisa ?
  3. Bagaimana aplikasi ini dapat membantu proses pembelian barang untuk stock?
  4. Bagaimana aplikasi ini dapat memantau pemasukan dan pengeluaran keuangan?
  5. Bagaimana aplikasi ini dapat melihat data pegawai yang ada dan membantu menambahkan atau mengurangi pegawai?

**1.3 BATASAN MASALAH**

Aplikasi penjualan ini hanya akan membantu proses penjualan dalam perhitungan yang efektif dan mempermudah melihat stok produk dan dapat menambahkan stok produk yang ada.

Aplikasi ini tidak akan mencatat apa saja yang harus di restock untuk suatu produk , yang mengharuskan pengguna membeli manual produk. Setelah membeli atau restock suatu produk barulah aplikasi ini membantu dengan menambahkan stok produk tersebut.

**1.4 TUJUAN PEMBUATAN**

Tujuan dari diadakannya penelitian, perancangan, dan pembuatan aplikasi penjualan dalam hal ini yaitu :

1. Membantu proses penjualan dengan aplikasi.
2. Membantu melihat stock produk yang tersisa dengan aplikasi.
3. Membantu proses pembelian barang untuk stock dengan aplikasi.
4. Membantu memantau pemasukan dan pengeluaran keuangan dengan aplikasi.
5. Dapat mengetahui data pegawai dan membantu menambahkan atau mengurangi pegawai yang ada dengan aplikasi.

**BAB II**

**Landasan Teori**

Pada proposal ini, peran teori sangat penting karena sebagai sebuah landasan dalam sebuah perancangan aplikasi. Karena tanpa landasan teori maka penelitian akan berujung pada kesalahan. Dengan adanya landasan teori ini, maka memberikan ciri bahwa sebuah penelitian itu, merupakan cara ilmiah untuk memperoleh data.

Basis data menurut Hengky W. Pramana (2002 : 3) adalah sebuah kumpulan dari sekelompok informasi yang di organisasikan dengan beberapa cara logik dan saling berhubungan.

Menurut Aji Supriyanto (2005: 190), kumpulan sekelompok informasi yang di organisasikan, memiliki sebuah tingkatan susunan organisasi data. Dengan demikian terdapat 5 tingkatan susunan organisasi data, antara lain :

**2.1 Karakter**

Merupakan bagian data terkecil, dapat berupa karakter numerik, huruf atau karakter khusus yang membentuk item data.

**2.2 Field**

Adalah suatu atribut dari record yang menunjukkan item dari data.

**2.3 Record**

Merupakan kumpulan dari field yang menggambarkan suatu unit data tertentu atau sekumpulan data item yang berhubungan secara logika dari suatu objek.

**2.4 File**

Merupakan kumpulan dari record yang menggambarkan suatu kesatuan yang sejenis.

**2.5 Database**

Merupakan kumpulan dari file yang membentuk satu kesatuan tertentu atau suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, tanpa adanya suatu kerangkapan data sehingga mudah untuk digunakan kembali.   
**2.6 Konsep Basis Data**

*Database*didefinisikan sebagai suatu kumpulan file-file yang saling berelasi sehingga membentuk suatu bangunan data. Relasi tersebut ditunjukkan dengan *key*kunci dari tiap file yang ada. Dalam suatu file terdapat *record-record*sejenis, besar dan bentuk merupakan kumpulan *entity*yang seragam. Satu *record*terdiri dari *field-field*yang saling berhubungan untuk menunjukkan bahwa *field*tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*. Isi *field*merupakan judul dari kelompok *entity*tersebut.

**2.7 MySQL**

MySQL adalah software database open source yang bersifat GPL(General Public License), artinya, software ini dapat dilihat source codenya dan dapat digunakan untuk kebutuhan apapun tanpa biaya. Pada awalnya MySQL hanya dapat berjalan pada komputer berbasis linux dan solaris, kini juga dapat berjalan pada komputer berbasis windows (Irawan, 2011).

MySQL merupakan database yang tampak sangat sederhana dengan tampilan berupa teks mode dan tampak seperti sekumpulan directory dimana database dan tabel-tabel disimpan secara terpisah, MySQL mendukung SQL(Structured Query Languages), yaitu bahasa tingkat tinggi yang digunakan untuk mendapatkan data yang diinginkan dari suatu database.

Dalam MySQL ada dua jenis statement, yaitu statement MySQL dan statement SQL, statement-statement MySQL , antara lain (Irawan, 2011) :

a.       Use (nm\_database) : Digunakan untuk mendefinisikan nama database yang akan kita gunakan.

b.      Show (statement) : Jika diikuti dengan kata databases, maka perintah ini akan menampilkan nama-nama database yang terdapat pada MySQL. Jika diikuti dengan kata tables, maka akan menampilkan tabel-tabel yang terdapat pada database yang dipilih dengan menggunakan perintah use nm\_database.

c.       Describe nm\_table : Digunakan untuk mendeskripsikan field-field dari suatu table beserta tipe datanya.

d.      Exit : Perintah untuk keluar dari MySQL.

**2.8 Konsep Basis Data**

Suatu sistem manajemen basis data berisi suatu koleksi data yang saling berelasi dan satu set program untuk mengakses dan memanipulasi data tersebut. Jadi sistem manajemen basis data terdiri dari basis data dan set program pengelola untuk menambah data, menghapus data, mengambil data dan membaca data (Irawan, 2011).

a.       Definisi dalam Konsep Basis Data

1)        *Entity* adalah orang, tempat, kejadian atau konsep yang informasinya direkam.

2)        *Atribute* adalah item data yang menjadi bagian dari suatu entitas. Istilah lain atribut adalah *property*.

3)        *Data Value*(nilai atau isi data) adalah data aktual atau informasi yang disimpan pada tiap data elemen (*atribute*).

4)        *Record /Tuple* adalah kumpulan elemen-elemen data yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap.

5)        *File*adalah kumpulan record-record sejenis yang mempunyai panjang elemen yang sama, *atribute* yang sama, namun bebeda-beda data valuenya.

6)        *Database* (basis data) adalah kumpulan tabel yang mempunyai kaitan (*relasi*) antara suatu tabel dengan tabel yang lainnya hingga terbentuk bangunan data yang menginformasikan satu instansi atau perusahaan dalam batasan-batasan tertentu. Bila terdapat tabel yang tidak dapat dipadukan atau dihubungkan dengan tabel yang lainnya berati tabel tersebut bukanlah kelompok dari basis data, yang akan membentuk suatu basis data sendiri.

7)        *Database Management System (DBMS)* adalah kumpulan file yang saling berkaitan bersama dengan program untuk pengelolaannya.

b.      Normalisasi

Proses normalisasi mempunyai pengelompokkan data elemen menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Proses normalisasi selalu diuji pada beberapa kondisi, yaitu kemungkinan ada kesulitan pada saat menambah, menghapus, mengubah, membaca pada suatu *database*. Bila ada kesulitan pada pengujian tersebut relasi dapat dipecahkan pada beberapa tabel lagi atau dengan kata lain perancangan belumlah mendapatkan *database* optimal.

Pada proses normalisasi ini perlu dikenal terlebih dahulu mengenai definisi dari tahap-tahap normalisasi. Tahap-tahap normalisasi terdiri dari :

1)       Bentuk tidak normal (*Unnnormalized Form*), Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu, bisa saja data tidak lengkap atau terduplikasi.

2)        Bentuk normal ke satu (*1NF /First Normal Form*), Bentuk normal ke satu mempunyai ciri-ciri yaitu setiap data dibentuk dalam *flat file* (file datar/rata), data dibentuk dalam satu record demi satu record dan nilai dari field-field berupa *atomic value*.

3)        Bentuk normal ke dua (*2NF /Second Normal Form*), Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal ke satu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama/ *primary key* sehingga untuk membentuk normal kedua haruslah sudah ditentukan kunci-kunci field. Kunci field haruslah unik dan dapat mewakili atribut lainnya yang menjadi anggotanya.

4)        Bentuk normal ketiga (*3NF / Third Normal Form*), Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribute bukan primer tidak punya hubungan yang transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada *primary key* dan pada *primary key*secara menyeluruh.

5)        *Boyce-Codd Normal Form (BCNF), Boyce-Codd Normal Form*mempunyai paksaan yang lebih kuat dari bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung fungsi pada atribute *superkey*.

**2.9 Xampp**

XAMPP adalah sebuah software *web server*apache yang didalamnya sudah tersedia database *server*MySQL dan dapat mendukung pemrograman PHP. XAMPP merupakan *software*yang mudah digunakan, gratis dan mendukung instalasi di *Linux*dan *Windows*. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache*Web *Server,*MySQL *Database Server,*PHP (Irawan, 2011).

**2.10 Netbeans**

Netbeans adalah salah satu aplikasi IDE yang digunakan programmer untuk menulis, mengompile, mencari kesalahan, dan menyebarkan program.netbeans ditulis dalam bahasa java namun dapat juga mendukung bahasa pemrogramman lain. program ini bebas digunakan

fitur-fitur pada netbeans Smart code completion : mengusulkan nama variable dari suatu tipe, melengkapi keyword, dan mengusulkan tipe parameter dari method Menggunakan code generator : dengan menggunakan fitur ini kita dapat meng-generate constructor, setter and getter method, dll.

Error stripe : fitur yang menandai baris yang error dengan menghiglight merah.

Bookmarking : fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi go to commands : fitur yang digunakan untuk jump ke deklarasi variable, source code atau file yang ada pada project yang sama.Database yang didukung Netbean JDBC merupakan spesifikasi standar dari javasoft API yang memungkinkan program java untuk mengakses sistem database manajemen.

JDBC API terdiri dari satu set interface dan kelas yang ditulis dengan bahasa pemrogramman java.

ojdbc (oracle java database conectivity) adalah driver yang menghubungkan program java dan oracle

mysql-connector-java adalah driver yang menghubungkan program java dan oracle JDBC-ODBC mySQL-server Netbeans sebagai IDE ditujukan untuk memudahkan pemrograman Java. Pada bulan Februari 2006 para instruktur Java dari Sun Microsystem mengikuti training untuk beralih dari pemrograman Java manual (memakai editor teks dan command prompt) ke pemrograman GUI dengan Netbeans.

Netbeans berbasis visual dan event-driven. Sama seperti IDE lainnya, misal Borland Delphi dan Microsoft Visual Studio. Netbeans mencakup compiler, builder dan debugger internal. Hal ini memudahkan proses pasca perancangan program. Proses deployment atau tes dapat dilakukan dengan Netbeans J2SE, J2EE, J2ME di Netbeans.

Netbeans seperti juga konsep Java sangat fleksibel. Sepanjang library Java tersedia, maka kita dapat melakukan pemrograman untuk jenis aplikasi apapun. Kita dapat membuat aplikasi dekstop (J2SE). Pemrograman web dan enterprise (J2EE) dapat dilakukan secara visual :

 Koneksi server database melalui JDBC dapat dilakukan dari Netbeans, baik pada saat perancangan maupun deployment program. Pembuatan komponen beans. Pembuatan Java Server Pages (JSP), web module (servicelocator dan servlet), web services dengan menggunakan wizard yang telah disediakan. Netbeans juga menyertakan paket web-server Apache Jakarta Tomcat, Sun Java System Application Server, GlassFish dll. Server ini dapat diakses dari dalam Netbeans, baik pada saat perancangan maupun saat deployment aplikasi web. Untuk membuat aplikasi mobile (MIDP), MIDlet dapat dirancang dengan Netbeans. Pada Netbeans 5.5 kita perlu menginstal Netbeans Mobility Pack, tetapi untuk Netbeans min versi 6.0, aplikasi MIDP, sudah ada. Netbeans 5.0 telah mendukung JDK 5.0, sedangkan Netbeans 5.5 telah mendukung JDK 6.0. Instal terlebih dulu min JDK 1.5. Jalankan file installer Netbeans 6.5 lalu ikuti langkah- langkah instalasi wizard.

Langkah pertama, perhatikan lisensi SPL (Sun Public License). Baca dengan teliti untuk memahami apa yang boleh dan apa yang tidak boleh dilakukan dalam pemakaian program ini.

 Langkah kedua, tentukan direktori instalasi.

Langkah ketiga, wizard melakukan pencarian otomatis terhadap JDK yang sudah terinstal dalam komputer. Jika berhasil maka JDK akan ditampilkan beserta direktori, jika tidak ditemukan maka kita dapat menentukan direktori dari JDK secara manual.

Langkah keempat, wizard akan melakukan instalasi Netbeans di lokasi direktori yang ditentukan. Proses instalasi ini akan memerlukan beberapa menit.

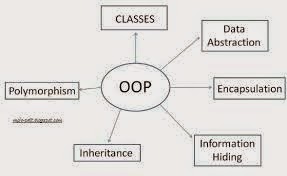
Proses instalasi akan membuatkan icon dalam menu Start. Kita dapat menjalankan Netbeans melalui icon ini.

11.) Metode OOP

OOP adalah sebuah metodologi atau cara berpikir dalam melakukan pemrograman dimana pendefinisian tipe data disertai dengan pendefinisian fungsi. Struktur data yang seperti ini disebut dengan istilah *object*. Paradigma pemrograman OOP dapat dilihat sebagai interaksi sebuah *object* dalam melakukan tugasnya.

Segelas teh tentunya memiliki berbagai spesifikasi seperti volume air, rasa, temperatur dan sebagainya. Pada pemrograman, spesifikasi-spesifikasi tersebut merupakan variabel yang dideklarasikan beserta tipe data. Pada OOP, sebuah object Teh juga memiliki *method* berupa fungsi yang dapat dipanggil untuk merubah kelakuan atau spesifikasi teh tersebut. Method minumTeh dapat dideklarasikan sebagai fungsi yang dapat dipanggil untuk mengurangi nilai variabel Volume.

Di samping itu, sebuah *object* dapat memiliki relasi terhadap *object* lainnya. Sebagai contoh, sebuah *object* dapat berupa keturunan dari *objec*t lain



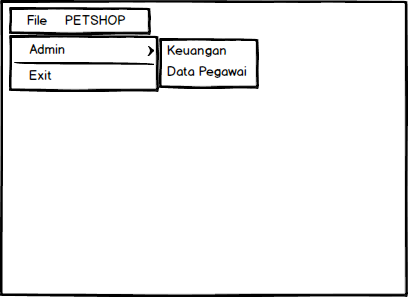
**Bab 3**

**Perancangan Sistem**

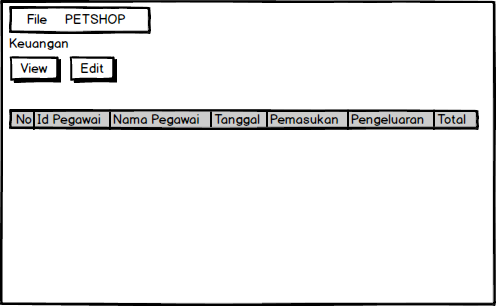
1. **Interaksi Manusia dan Komputer**

Pada interaksi manusia dan komputer ini, akan di jelaskan tampilan aplikasi Petshop. Aplikasi memiliki desain yang cukup simple sehingga sangat memudahkan user yang baru memakai aplikasi ini. Pada aplikasi ini, memiliki 5 form dan dibagi 2 untuk admin, yaitu : Keuangan dan Data Pegawai. Untuk pegawai, yaitu : Data Penjualan, Data Pembelian dan Stock Barang. Di setiap form, memiliki masing – masing fitur, berikut penjelasan setiap fitur nya :

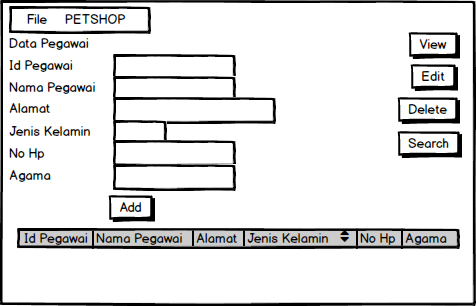
1. Form Keuangan untuk Admin

  
gambar 3.1 Keuangan

1. Menu Strip : Untuk menampilkan menu yang ada seperti File dan Petshop
2. Menu Admin : Terdapat 2 menu yaitu Keuangan dan Data Pegawai
3. Keuangan : Untuk masuk ke form Keuangan
4. Data Pegawai : Untk masuk ke form Data Pegawai
5. Exit : Untuk keluar dari applikasi

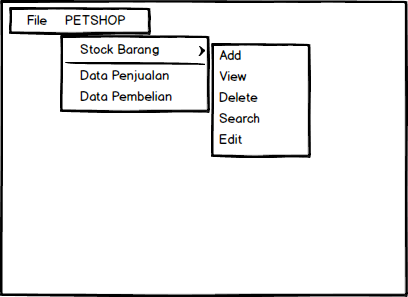
  
gambar 3.2 Keuangan untuk Admin

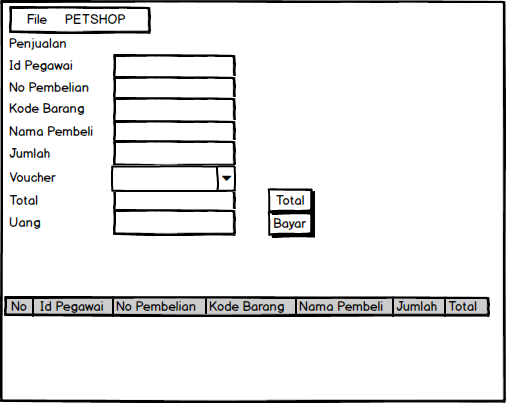
1. View : Untuk melihat database keuangan dan siapa yang menginput data tersebut dari database data pegawai.
2. Edit : Untuk mengedit data yang ada pada datagrid tersebut.
3. Data Grid View : Untuk menampilkan database dari keuangan.
4. Form Data Pegawai untuk Admin



gambar 3.3 Data Pegawai untuk Admin

1. Textbox Id Pegawai : Untuk input Id Pegawai
2. Textbox Nama Pegawai : Untuk input Nama Pegawai
3. Textbox Alamat : Untuk input Alamat
4. Textbox Jenis Kelamin : Untuk input Jenis Kelamin
5. Textbox No HP : Untuk input No Hp
6. Textbox Agama : Untuk input Agama
7. View : Untuk melihat database data pegawai
8. Edit : Untuk mengedit database data pegawai berdasarkan id pegawai
9. Delete : Untuk menghapus database data pegawai berdasarkan id pegawai
10. Search : Untuk mencari database data pegawai berdasarkan id pegawai
11. Add : Untuk menambahkan database data pegawai
12. Form Data Penjualan untuk Pegawai

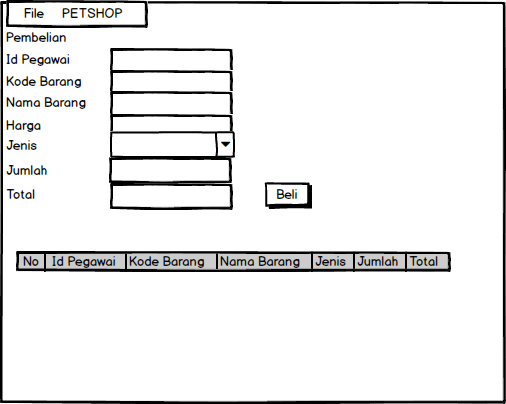
  
gambar 3.4 Tampilam Menu Petshop



gambar 3.5 Data Penjualan untuk Pegawai

* Total : Untuk menjumlahkan semua barang yang dibeli dengan potongan voucher atau tanpa voucher.
* Bayar : Dengan mengisikan textbox Uang dan bayar untuk membayar dari total pembelian dan akan dicatat dalam databse penjualan untuk masuk ke pemasukan dan mengurangi stock barang didatabase stock barang.

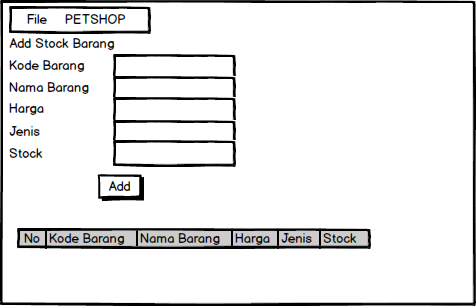
1. Form Data Pembelian untuk Pegawai



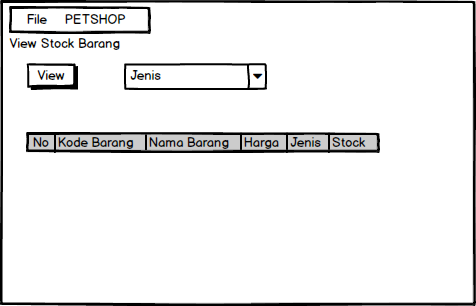
gambar 3.6 Data Pembelian untuk Pegawai

* Beli : Setelah semua diisi dengan benar button beli akan mencatat didatabase data pembelian untuk pengeluaran dan akan menambah data didatabase stock barang.

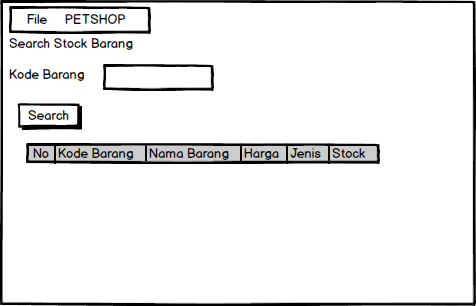
1. Form Stock Barang untuk Pegawai.



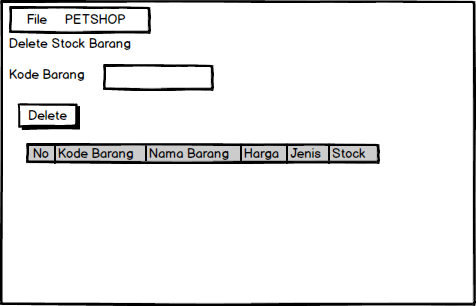
Gambar 3.7 Add Stock barang



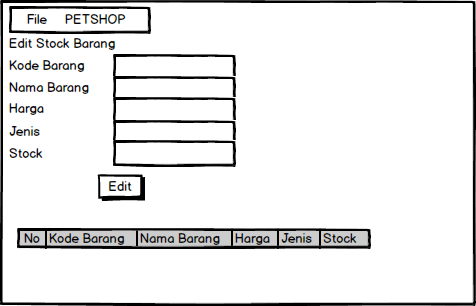
Gambar 3.8 View Stock barang



Gambar 3.9 Search Stock barang



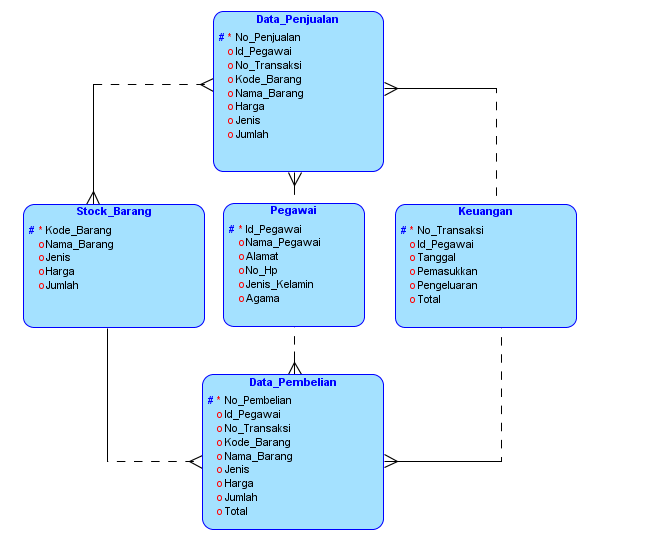
Gambar 3.10 Delete Stock barang



Gambar 3.11 Edit Stock barang

1. View : Untuk melihat database stock barang
2. Edit : Untuk mengedit database stock barang berdasarkan kode barang
3. Delete : Untuk menghapus database stock barang berdasarkan kode barang
4. Search : Untuk mencari database stock barang berdasarkan kode barang
5. Add : Untuk menambahkan database stock barang
6. **Entity Relationship Diagram**

Entity Relationship diagram (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analys dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database.



Gambar 3.12 Entity Relationship Diagram

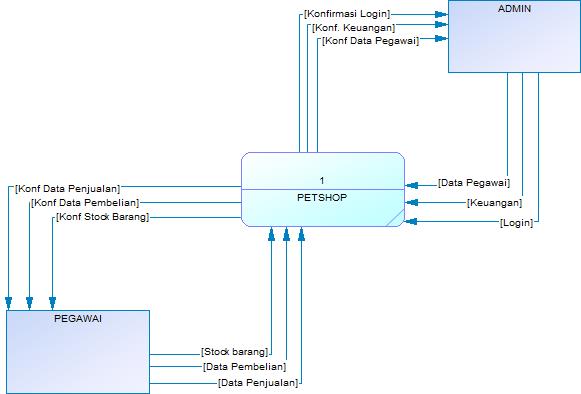
Pada Entity Flow Diagram terdapat 5 Entity dengan masing- masing attribut sebagai berikut

Tabel 3.1 Entity Relationship Diagram

|  |  |
| --- | --- |
| **Entity** | **Atribut** |
| Data\_Pegawai | Id\_Pegawai Nama\_Pegawai Alamat  No\_HP Jenis\_Kelamin Agama |
| Data\_Pembelian | Id\_Pembelian  Id\_Pegawai  No\_Transaksi Kode\_Barang Nama\_Barang Harga  Jenis  Jumlah Total |
| Data\_Penjualan | No\_Penjual  Id\_Pegawai  No\_Transaksi Kode\_Barang Nama\_Barang  Harga Jumlah  Jenis  Jumlah  Total |
| Stock\_Barang | Kode\_Barang Nama\_Barang Harga Jenis  Jumlah |
| Keuangan | No\_Transaksi  Id\_Pegawai  Tanggal  Pemasukkan  Pengeluaran  Total |

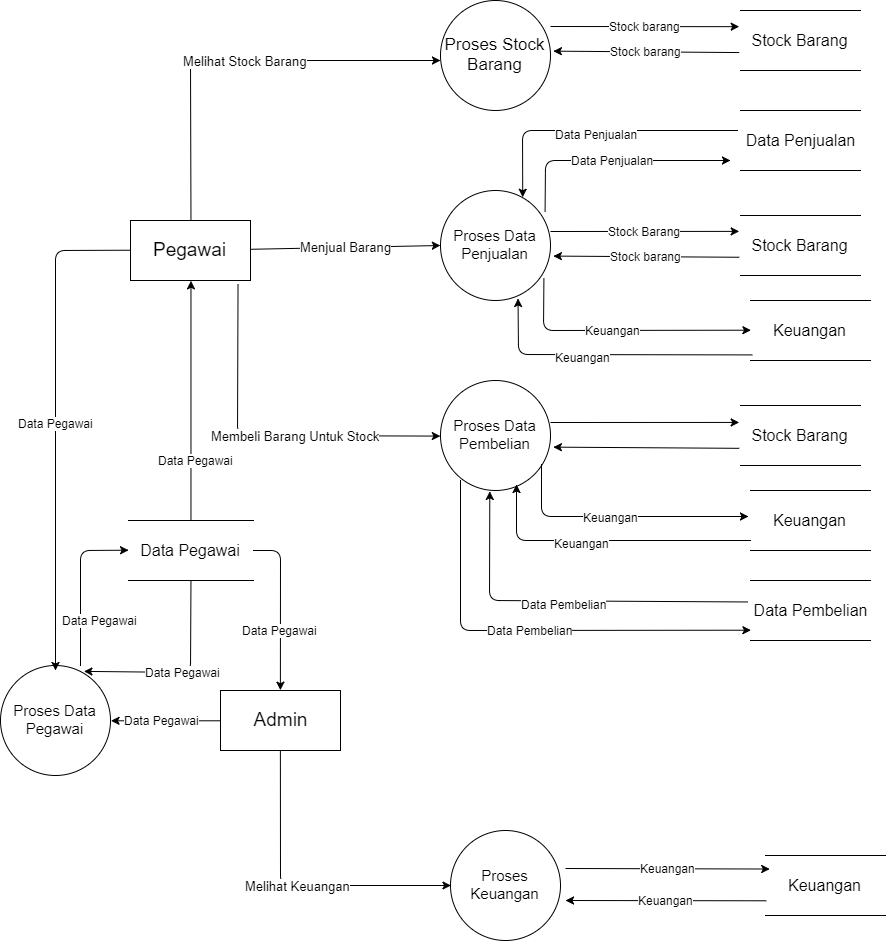
1. **Data Flow Diagram**

DFD berfungsi untuk menggambarkan suatu sistem yang lama kesistem yang baru yang akan dikembanghkan dimana sata tersebut mengalir dan menulis informasi serta menggambatkan aliean indirmasi dari suatu proses keproses lainnya, berikut ini adalah data flow diagram untuk Petshop yang dikembangkan :

****

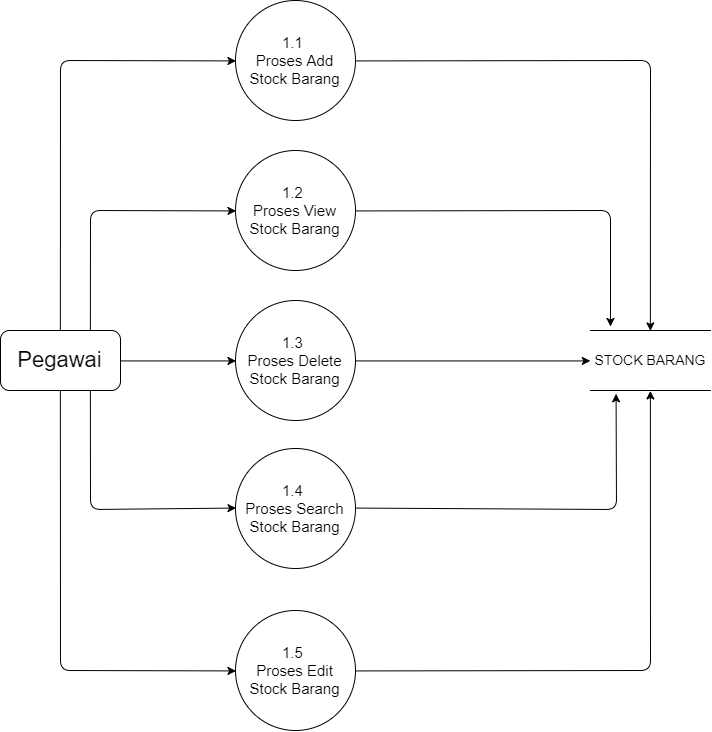
Gambar 3.13 DFD Lv 0

Gambar diatas menjeleaskan proses dari *DFD* level 0



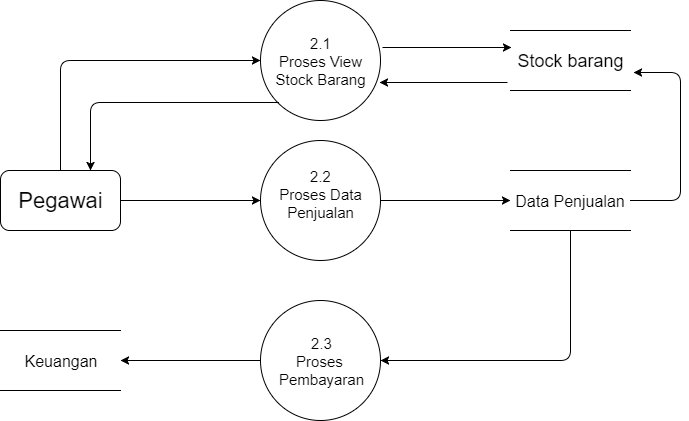
Gambar 3.14 DFD Lv 1

Gambar diatas menjeleaskan proses dari *DFD* level 1



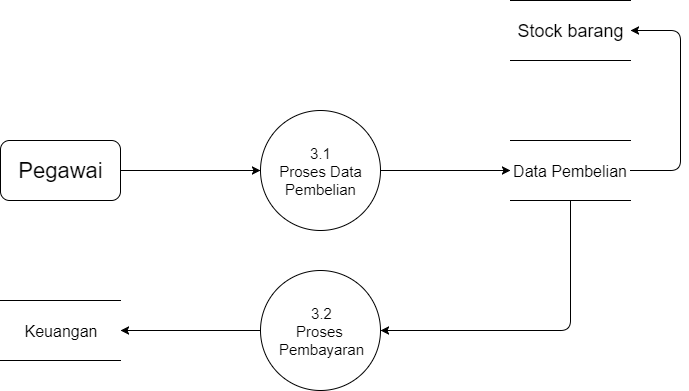
Gambar 3.15 DFD Lv 2.1

Gambar diatas menjeleaskan proses dari *DFD* level 2 Proses Stock Barang



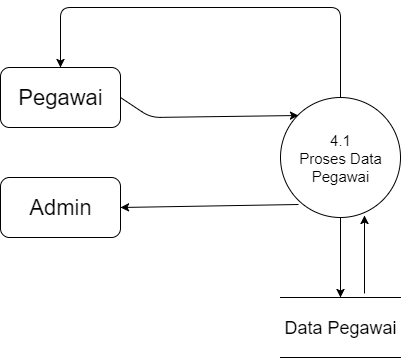
Gambar 3.16 DFD Lv 2.2

Gambar diatas menjeleaskan proses dari *DFD* level 2 Proses Data Penjualan



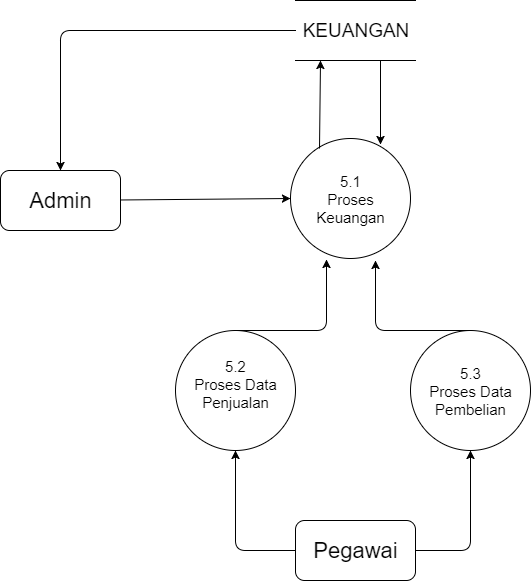
Gambar 3.17 DFD Lv 2.3

Gambar diatas menjeleaskan proses dari *DFD* level 2 Proses Data Pembelian



Gambar 3.18 DFD Lv 2.4

Gambar diatas menjeleaskan proses dari *DFD* level 2 Proses Admin

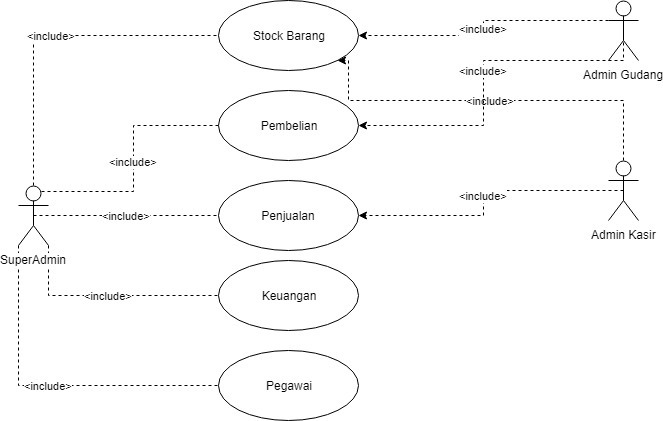


Gambar 3.19 DFD Lv 2.5

Gambar diatas menjeleaskan proses dari *DFD* level 2 Proses Keuangan

1. **Use Case**

Diagram **use case** mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Diagram **use case** digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.



Gambar 3.20 Use Case Petshop

1. **Skenario Use Case**

| ***Use case Name*** | ***Login*** | |
| --- | --- | --- |
| ***Number*** | UCA-01 | |
| ***Actor*** | *Superadmin, Admin* Kasir, *Admin* Gudang | |
| ***Description*** | Proses untuk mengakses *application Petshop* | |
| ***Main Success Scenario*** | ***User*** | ***System*** |
|  | 1. Pengguna menjalankan applikasi *petshop* |  |
|  |  | 1. Memperlihatkan menu *form login* |
|  | 1. *User* melakukan *log in* pada sistem agar dapat mengakses sistem. *Log in* dilakukan dengan cara memasukkan *username* dan *password user* pada sistem, kemudian *user* menekan tombol “log in”. |  |
|  |  | 1. Sistem melakukan verifikasi apakah *user* sudah terdaftar pada aplikasi atau belum. |
|  |  | 1. *Redirect* ke halaman awal sesuai dengan *user role.* |

| ***Use case Name*** | ***Stock* Barang** | |
| --- | --- | --- |
| ***Number*** | UCA-02 | |
| ***Actor*** | *Superadmin*, *Admin* Gudang | |
| ***Description*** | Proses untuk mengakses menu stock barang pada aplikasi *petshop.* | |
| ***Main Success Scenario*** | ***User*** | ***System*** |
|  | 1. Pengguna berhasil *log in.* |  |
|  | 1. *Log in* sebagai *Superadmin* |  |
|  | 1. *Log in* sebagai *Admin* Gudang |  |
|  |  | 1. Jika sebagai *superadmin* pada tampilan home akan menampilkan menu *stock* barang, pembelian, penjualan, keuangan dan pegawai. |
|  |  | 1. Jika sebagai *admin* gudang pada tampilan home akan menampilkan menu *stock* barang dan pembelian. |
|  | 1. Pengguna memilih menu *stock* barang. |  |
|  |  | 1. Jika sebagai *superadmin* pada tampilan *stock* barang akan memunculkan list barang yang tersedia dan berbagai tombol untuk menghapus dan mencari data pembelian. |
|  |  | 1. Jika sebagai *admin* gudang pada tampilan *stock* barang, memunculkan list barang yang tersedia dan mencari data barang. |
|  | 1. Jika *superadmin* mengetikkan id barang pada *textbox* dan melakukan aksi pencarian. |  |
|  |  | 1. Memunculkan data barang sesuai dengan id barang yang dicari. |
|  | 1. Jika *superadmin* mengetikkan id barang pada *textbox* dan melakukan aksi hapus. |  |
|  |  | 1. Maka data barang yang sama dengan id diketikan oleh pengguna akan terhapus. |
|  | 1. Jika *admin* gudangmengetikkan id barang pada *textbox* dan melakukan aksi pencarian. |  |
|  |  | 1. Maka memunculkan data barang yang dicari. |

| ***Use case Name*** | **Pembelian** | |
| --- | --- | --- |
| ***Number*** | UCA-03 | |
| ***Actor*** | *Superadmin*, *Admin* Gudang | |
| ***Description*** | Proses untuk mengakses menu pembelian pada aplikasi *petshop.* | |
| ***Main Success Scenario*** | ***User*** | ***System*** |
|  | 1. Pengguna berhasil *log in.* |  |
|  | 1. *Log in* sebagai *Superadmin* |  |
|  | 1. *Log in* sebagai *Admin* Gudang |  |
|  |  | 1. Jika sebagai *superadmin* pada tampilan home akan menampilkan menu *stock* barang, pembelian, penjualan, keuangan dan pegawai. |
|  |  | 1. Jika sebagai *admin* gudang pada tampilan home akan menampilkan menu *stock* barang dan pembelian. |
|  | 1. Pengguna memilih menu pembelian |  |
|  |  | 1. Jika sebagai *superadmin* pada tampilan pembelian akan memunculkan list data pembelian dan berbagai tombol untuk menghapus dan mencari data pembelian. |
|  |  | 1. Jika sebagai *admin* gudang pada tampilan pembelian terdapat inputan inputan untuk menambah data pembelian, memunculkan list data pembelian dan dapat menambahkan catatan jikalau ada stock barang yang tinggal sedikit. |
|  | 1. Jika *superadmin* mengetikkan id pembelian pada *textbox* dan melakukan aksi pencarian. |  |
|  |  | 1. Memunculkan data pembelian sesuai dengan id pembelian yang dicari. |
|  | 1. Jika *superadmin* mengetikkan id pembelian pada *textbox dan melakukan aksi hapus.* |  |
|  |  | 1. Maka data pembelian yang sama dengan id diketikan oleh pengguna akan terhapus. |
|  | 1. Jika *admin* gudang mengisi data – data pembelian dan melakukkan aksi penambahan data. |  |
|  |  | 1. Maka data pembelian akan bertambah. |

| ***Use case Name*** | **Penjualan** | |
| --- | --- | --- |
| ***Number*** | UCA-04 | |
| ***Actor*** | *Superadmin*, *Admin* Kasir | |
| ***Description*** | Proses untuk mengakses menu penjualan pada aplikasi *petshop.* | |
| ***Main Success Scenario*** | ***User*** | ***System*** |
|  | 1. Pengguna berhasil *log in.* |  |
|  | 1. *Log in* sebagai *Superadmin* |  |
|  | 1. *Log in* sebagai *Admin* Kasir |  |
|  |  | 1. Jika sebagai *superadmin* pada tampilan home akan menampilkan menu *stock* barang, pembelian, penjualan, keuangan dan pegawai. |
|  |  | 1. Jika sebagai *admin* kasir pada tampilan home akan menampilkan menu *stock* barang dan penjualan. |
|  | 1. Pengguna memilih menu penjualan. |  |
|  |  | 1. Jika sebagai *superadmin* pada tampilan penjualan akan memunculkan list data penjualan dan berbagai tombol untuk menghapus dan mencari data penjualan. |
|  |  | 1. Jika sebagai *admin* kasir pada tampilan penjualan terdapat inputan inputan untuk menambah data penjualan, memunculkan list data penjualan dan dapat mencari data penjualan. |
|  | 1. Jika *superadmin* mengetikkan id penjualan pada *textbox* dan melakukan aksi pencarian. |  |
|  |  | 1. Memunculkan data penjualan sesuai dengan id yang dicari. |
|  | 1. Jika *superadmin* mengetikkan id penjualan pada *textbox* dan melakukan aksi hapus. |  |
|  |  | 1. Maka data penjualan yang sama dengan id diketikan oleh pengguna akan terhapus. |
|  | 1. Jika *admin* kasir mengisi data – data penjualan dan melakukkan aksi penambahan data. |  |
|  |  | 1. Maka data penjualan akan bertambah. |
|  | 1. Jika *admin* kasirmengetikkan id penjualan pada *textbox* dan melakukan aksi pencarian. |  |
|  |  | 1. Memunculkan data penjualan sesuai dengan id yang dicari. |

| ***Use case Name*** | **Keuangan** | |
| --- | --- | --- |
| ***Number*** | UCA-05 | |
| ***Actor*** | *Superadmin* | |
| ***Description*** | Proses untuk mengakses menu keuangan pada aplikasi *petshop.* | |
| ***Main Success Scenario*** | ***User*** | ***System*** |
|  | 1. Pengguna berhasil *log in.* |  |
|  | 1. *Log in* sebagai *Superadmin* |  |
|  |  | 1. Jika sebagai *superadmin* pada tampilan home akan menampilkan menu *stock* barang, pembelian, penjualan, keuangan dan pegawai. |
|  | 1. Pengguna memilih menu keuangan. |  |
|  |  | 1. Jika sebagai *superadmin* pada tampilan penjualan akan memunculkan list keuangan dari penjualan dan pembelian dan berbagai tombol untuk menghapus dan mencari laporan keuangan. |
|  | 1. Jika *superadmin* mengetikkan id keuangan pada *textbox* dan melakukan aksi pencarian. |  |
|  |  | 1. Memunculkan data keuangan sesuai dengan id yang dicari. |
|  | 1. Jika *superadmin* mengetikkan id keuangan pada *textbox* dan melakukan aksi hapus. |  |
|  |  | 1. Maka data keuangan yang sama dengan id diketikan oleh pengguna akan terhapus. |

| ***Use case Name*** | **Pegawai** | |
| --- | --- | --- |
| ***Number*** | UCA-06 | |
| ***Actor*** | *Superadmin* | |
| ***Description*** | Proses untuk mengakses menu penjualan pada aplikasi *petshop.* | |
| ***Main Success Scenario*** | ***User*** | ***System*** |
|  | 1. Pengguna berhasil *log in.* |  |
|  | 1. *Log in* sebagai *Superadmin* |  |
|  |  | 1. Jika sebagai *superadmin* pada tampilan home akan menampilkan menu *stock* barang, pembelian, penjualan, keuangan dan pegawai. |
|  | 1. Pengguna memilih menu Pegawai. |  |
|  |  | 1. Jika sebagai *superadmin* pada tampilan data pegawai akan memunculkan list data pegawai dan berbagai tombol untuk menghapus dan mencari data penjualan. |
|  |  | 1. Terdapat inputan inputan untuk menambah pegawai. |
|  | 1. Jika *superadmin* mengetikkan id pegawai pada *textbox* dan melakukan aksi pencarian. |  |
|  |  | 1. Memunculkan data pegawai sesuai dengan id yang dicari. |
|  | 1. Jika *superadmin* mengetikkan id pegawai pada *textbox* dan melakukan aksi hapus. |  |
|  |  | 1. Maka data pegawai yang sama dengan id diketikan oleh pengguna akan terhapus. |
|  | 1. Jika pengguna mengisi inputan inputan untuk menambah pegawai dan melakukan aksi tambah |  |
|  |  | 1. Maka data pegawai akan bertambah. |

5. PENUTUP

Demikian proposal ini kami sampaikan, semoga dapat menjadi pertimbangan dan bermanfaat bagi semua pihak yang berkepentingan. Atas diterimanya proposal ini, kami mengucapkan terimakasih