

Laporan Tugas Besar Pemrograman Berorientasi Objek

“Aplikasi Kasir Toko Aksesoris HP”

diajukan untuk memenuhi tugas besar mata kuliah pemrograman berorientasi objek



Nama : Agni Pangestu
Nrp : 152019038
Kelas : A

**PROGRAM STUDI INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
BANDUNG**

2021

Kata Pengantar

Segala puji bagi Allah SWT yang telah memberikan kami kemudahan sehingga kami dapat menyelesaikan laporan ini dengan tepat waktu. Tanpa pertolongan-Nya tentunya kami tidak akan sanggup untuk menyelesaikan makalah ini dengan baik. Shalawat serta salam semoga tercurah limpahkan kepada baginda nabi yaitu Nabi Muhammad SAW, kepada para sahabatnya, dan kepada kita sebagai umatnya.

Puji syukur kepada Allah SWT atas limpahan nikmat sehat-Nya, baik itu berupa sehat fisik maupun akal pikiran, sehingga penulis mampu untuk menyelesaikan Makalah ini yang berjudul “Makalah Eksepsi Java”.

Saya tentu menyadari bahwa makalah ini masih jauh dari kata sempurna dan masih terdapat banyak kesalahan serta kekurangan di dalamnya. Untuk itu, diharapkan kritik serta saran dari pembaca, supaya makalah ini dapat menjadi makalah yang lebih baik lagi. Kemudian apabila terdapat banyak kesalahan pada makalah ini, penulis mohon maaf yang sebesar-besarnya. Demikian, semoga makalah ini dapat bermanfaat. Terima kasih.

Bandung, 3 Juni 2021

Penulis

Daftar Isi

Kata Pengantar	ii
Daftar Isi	iii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan.....	1
1.4 Batasan Masalah.....	1
BAB II LANDASAN TEORI.....	3
2.1 Pemrograman berorientasi objek.....	3
2.2 NetBeans	3
2.3 Heidi SQL	4
BAB III METODE PENGKERJAAN	5
3.1 Studi Kasus.....	5
3.2 Fitur Aplikasi.....	5
3.3 Alur Proses	5
3.4 Class Diagram	6
3.5 Database	7
3.6 Skema Aplikasi	8
3.7 Mockup UI	9
3.8 Use Case Diagram.....	11
BAB IV IMPLEMENTASI	12
3.1 Package com.....	12
3.2 Package db.....	20
3.3 Package exec	21
3.4 Package view.....	28
3.5 GUI FormLogin.....	33
3.6 GUI FormData.....	33
BAB V PENUTUP	35
2.1 Kesimpulan.....	35
2.2 Saran.....	35

BAB I PENDAHULUAN

1.1 Latar Belakang

Seiring berkembangnya zaman, teknologi pasti akan sangat dibutuhkan pada bidang manapun. Bisa dilihat dari beberapa tahun yang lalu sampai tahun ini sudah sampai mana perkembangan teknologi terus berkembang. Mulai dari media pembelajaran, media informasi dan media hiburan. Setiap tahun ke tahun selalu akan ada perkembangan teknologi.

Transaksi Manual masih banyak yang menggunakan metode bon manual yang di tulis menggunakan tangan. Dengan teknologi yang ada, maka Transaksi bisa mempermudah penjualan agar data transaksi yang sudah dilakukan bisa terdata lebih baik.

Berdasarkan latar belakang masalah diatas maka saya membuat “Aplikasi Transaksi Toko Aksesoris HP”.

Aplikasi ini dikembangkan memanfaatkan teknologi informasi berbasis GUI (Graphic User Interface) yaitu menggunakan netbeans dan database Mysql dengan menggunakan bantuan HeidiSQL.

1.2 Rumusan Masalah

1. Bagaimana cara kerja sistem “Aplikasi Transaksi Toko Aksesoris HP” dengan menggunakan Heidi SQL dan NetBeans?
2. Bagaimana cara kerja “Aplikasi Transaksi Toko Aksesoris HP”?

1.3 Tujuan

1. Mengetahui cara kerja sistem “Aplikasi Transaksi Toko Aksesoris HP” dengan menggunakan Heidi SQL dan NetBeans.
2. Mengetahui cara kerja “Aplikasi Transaksi Toko Aksesoris HP”

1.4 Batasan Masalah

1. Transaksi yang dilakukan belum bisa memilih barang yang berbeda dalam satu transaksi. Jadi jika barangnya berbeda, transaksinya yang dilakukan pun berbeda.
2. Aplikasi hanya bisa di pakai oleh Pegawai atau Owner dari toko.
3. Aplikasi ini belum bisa mencetak struk transaksi.

1.5 Sistematika Penulisan Laporan

Sistematika penulisan laporan ini mengacu pada subbab berikut ini :

1. BAB I, PENDAHULUAN

Pada bab ini menjelaskan tentang tujuan latar belakang pembuatan aplikasi, rumusan masalah, tujuan penulisan laporan dan batasan masalah dari sistem yang penulis buat.

2. BAB II, LANDASAN TEORI

Bab ini menjelaskan tentang teori-teori penunjang yang digunakan dalam pembuatan laporan ini.

3. BAB III, PERANCANGAN

Pada bagian bab ini menjelaskan tentang studi kasus, alur sistem dan database sistem, sitemaps aplikasi dan mockup user interface.

4. BAB IV, IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem menjadi program aplikasi berbasis GUI dengan menggunakan bahasa pemrograman Java.

5. BAB V, PENUTUP

Bab ini berisi tentang kesimpulan dari pembuatan sistem aplikasi dan saran – saran dari penulis tentang aplikasi ini.

BAB II LANDASAN TEORI

2.1 Pemrograman berorientasi objek

OOP/PBO merupakan paradigma pemrograman yang populer saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. Object Oriented Programming yang berarti pula Pemrograman Berorientasi Objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970.

Pemrograman Berorientasi Objek (Object Oriented Programming/OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class-class atau object-object. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya.

Masing-masing object harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan Object yang lain. Pemrograman berorientasi objek berbeda dengan pemrograman prosedural yang hanya menggunakan satu halaman kebawah untuk mengerjakan banyak perintah atau statement. Penggunaan pemrograman berorientasi objek sangat banyak sekali, contoh : java, php dan lainnya.

2.2 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 :

- 1) Smart code completion : mengusulkan nama variable dari suatu tipe, melengkapi keyword, dan mengusulkan tipe parameter dari method
- 2) Menggunakan code generator : dengan menggunakan fitur ini kita dapat meng-generate constructor, setter and getter method, dll.
- 3) Error stripe : fitur yang menandai baris yang error dengan menghighlight merah.
- 4) Bookmarking : fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi

- 5) Go to commands : fitur yang digunakan untuk jump ke deklarasi variable, source code atau file yang ada pada project yang sama.

Database yang didukung Netbeans :

- a. JDBC merupakan spesifikasi standar dari javasoft API yang memungkinkan program java untuk mengakses sistem database manajemen.
- b. JDBC API terdiri dari satu set interface dan kelas yang ditulis dengan bahasa pemrograman java.
- c. ojdbc (oracle java database connectivity) adalah driver yang menghubungkan program java dan oracle
- d. mysql-connector-java adalah driver yang menghubungkan program java dan oracle
- e. JDBC-ODBC
- f. 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.

2.3 Heidi SQL

HeidiSQL adalah interface gratis untuk MySQL server. Tool ini sangat ringan dan berjalan tanpa masalah bahkan di Windows 7. Tool ini merupakan alternatif bagi pengguna yang ingin menggunakan MySQL dengan tampilan GUI (Graphical User Interface).

Dibandingkan dengan tool serupa yang komersil, HeidiSQL memang belum seberapa. Tapi utilitas terkait dengan administrasi dasar, manajemen user, ekspor-impor, DDL, DML dan DAL telah didukung. Hanya saja, ekspor baru mendukung format CSV, XML dan SQL script. Belum ada dukungan ekspor ke Excel atau format lainnya yang lebih readable dan editable.

Data Object editor, seperti table editor, view editor dan database creator, telah sangat bagus dan mudah digunakan. Dilengkapi dengan warna-warna berbeda sesuai tipe field pada table editor.

BAB III METODE Pengerjaan

3.1 Studi Kasus

Dalam aplikasi ini hanya terdapat satu user yaitu admin. Fungsi dari admin untuk mengatur database baik itu menambahkan, mengubah dan menghapus data. User admin juga bisa melakukan proses transaksi untuk pembelian barang. Kemudian data transaksi akan masuk ke database dan akan di tampilkan di tabel riwayat transaksi.

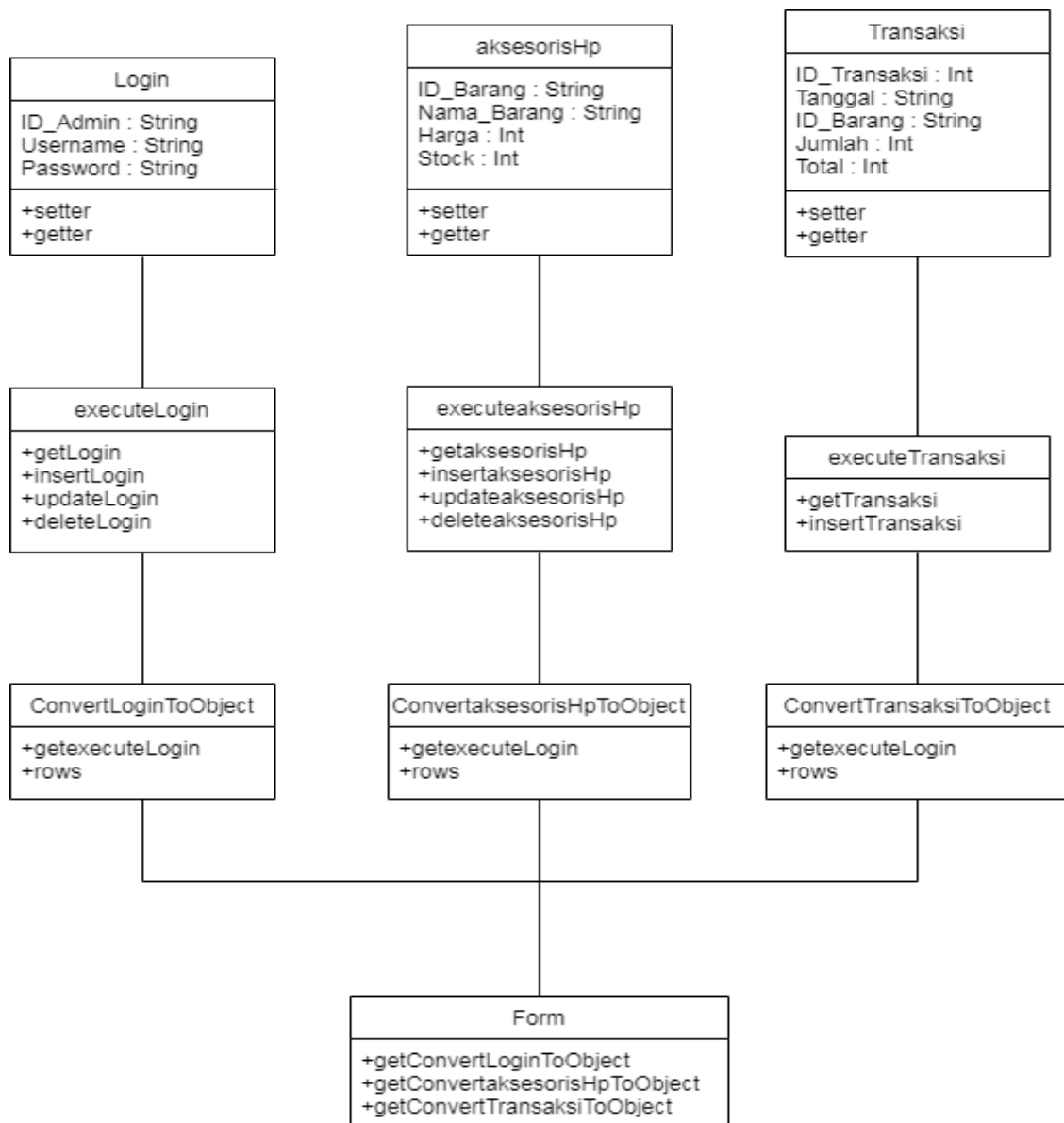
3.2 Fitur Aplikasi

Fitur yang tersedia pada aplikasi ini ada bermacam-macam, mulai dari Create, Read, Update, dan Delete pada suatu tabel database. Kemudian ada fitur transaksi dan hasil proses transaksi tersebut dimasukan ke database dan akan di tampilkan di tabel riwayat transaksi.

3.3 Alur Proses

Proses aplikasi ini dimulai dengan user login ke form login, user memasukan username dan password yang sudah dibuat di database. Lalu memilih button untuk mengarahkan ke panel admin, barang atau transaksi. Pada panel admin, user bisa mengedit database admin untuk login dengan CRUD yang dibuat. Pada panel barang, user bisa mengedit database barang yang dijual oleh Toko Aksesoris HP ini dengan CRUD yang dibuat. Lalu untuk panel transaksi, user bisa melakukan transaksi dengan memilih barang yang akan dijual ke pelanggan dan melakukan perhitungan transaksi. Lalu hasil transaksi akan di update ke tabel transaksi.

3.4 Class Diagram



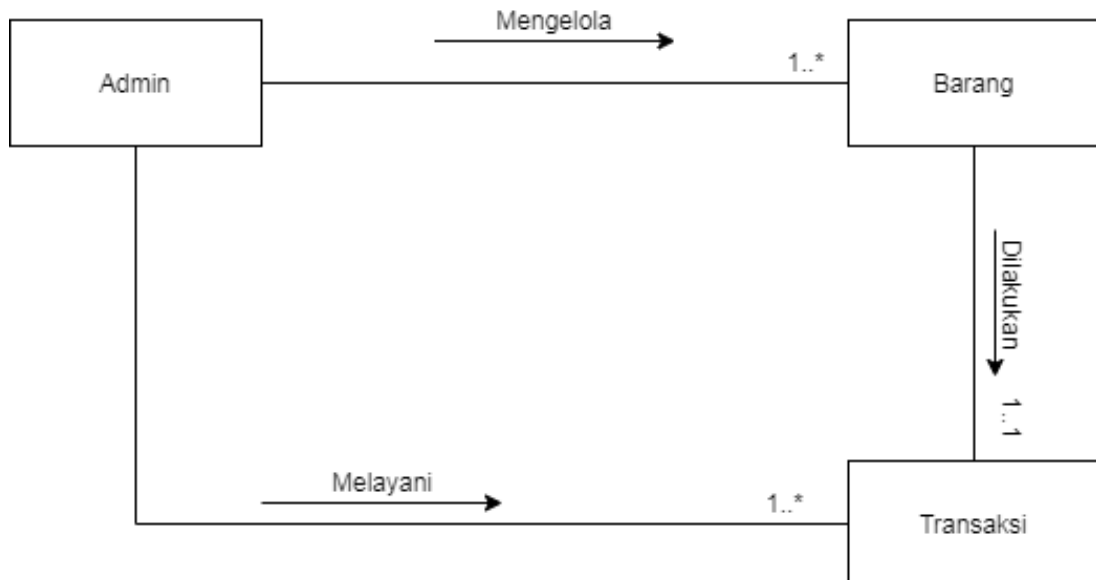
Keterangan

Class diagram terdiri dari 3 tabel database, isinya berupa data yang diinputkan pada awal pembuatan database sesuai dengan apa yang diinputkan user. Kemudian ke 3 tabel tersebut dikirimkan ke ConnectionManager untuk disambungkan dengan JDBC NetBeans, setelah dari ConnectionManager kemudian dikirimkan ke masing-masing execute yang berfungsi untuk CRUD database, lalu dikirimkan lagi ke masing - masing Convert untuk mendapatkan rows tabel, akhirnya semua tabel melewati proses akhir yaitu dikirimkan ke Form.

3.5 Database

1. ERD

Entity Relationship Diagram (ERD) adalah sebuah model diagram yang menjelaskan hubungan antar objek pada basis data yang mempunyai hubungan antar relasi.



2. TRD

TRD (Table Relationship Diagram) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi yang berbentuk tabel dan memiliki atributnya masing-masing yang saling terhubung dengan atribut di tabel yang lain.

gidaw_case admin
Id_Admin : varchar(4)
Username : varchar(10)
Password : varchar(10)

gidaw_case aksesoris_hp
ID_Barang : varchar(5)
Nama_Barang : varchar(30)
Harga : int(15)
Stock : int(3)

gidaw_case transaksi
ID_Transaksi : int(4)
Tanggal : varchar(50)
ID_Barang : varchar(5)
Jumlah : int(3)
Total : int(15)



3. DDS

Kamus data (data dictionary) adalah suatu penjelasan tertulis tentang suatu data yang berada di dalam database atau suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga user dan analis sistem mempunyai pengertian yang sama tentang input, output, dan komponen data store.

1 admin

Pembuatan: 08 Jun 2021 pada 21.39

Kolom	Jenis	Atribut	Tak Ternak	laBawaan	Ekstra	Tautan ke	Komentar	MIME
Id Admin	varchar(4)		Tidak				id unik dari table admin	
Username	varchar(10)		Tidak				username untuk login	
Password	varchar(10)		Tidak				password untuk login	

DDS Admin

2 aksesoris_hp

Pembuatan: 08 Jun 2021 pada 21.39

Kolom	Jenis	Atribut	Tak Ternak	laBawaan	Ekstra	Tautan ke	Komentar	MIME
ID Barang	varchar(5)		Tidak				id unik barang	
Nama Barang	varchar(30)		Tidak				nama barang	
Harga	int(15)		Tidak				harga barang	
Stock	int(3)		Tidak				stock barang	

DDS aksesoris_hp

3 transaksi

Pembuatan: 08 Jun 2021 pada 21.39

Kolom	Jenis	Atribut	Tak Ternak	laBawaan	Ekstra	Tautan ke	Komentar	MIME
ID Transaksi	int(4)		Tidak		auto increment		id unik transaksi	
Tanggal	varchar(50)		Tidak				tanggal transaksi	
ID Barang	varchar(5)		Tidak			-> aksesoris_hp.ID_Barang ON UPDATE RESTRICT ON DELETE RESTRICT	id unik barang yang dipilih	
Jumlah	int(3)		Tidak				jumlah barang yang ingin dibeli	
Total	int(15)		Tidak				total transaksi	

DDS Transaksi

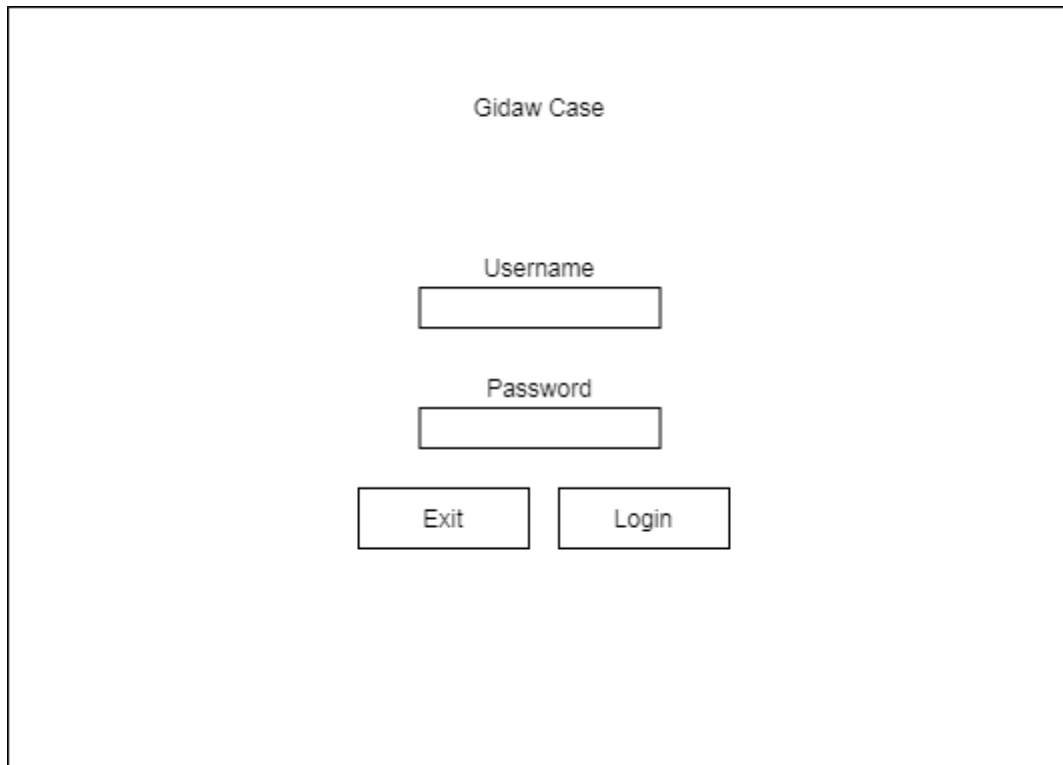
3.6 Skema Aplikasi



Keterangan

Dimulai dari user memasukan username dan password di form login, bila user masuk ke form admin akan keluar panel awal. Lalu jika menekan button admin maka akan ke panel yang isinya table admin dan memiliki fungsi CRUD tabel, jika menekan button barang maka akan ke panel yang isinya table barang dan memiliki fungsi CRUD tabel. Jika menekan tombol transaksi, maka panel transaksi akan muncul. Disini lah transaksi dilakukan dimana data dari transaksi akan di sim[an di tabel riwayat transaksi. Terakhir, terdapat tombol logout untuk kembali ke form login.

3.7 Mockup UI



A mockup of a login interface. It features a title 'Gidaw Case' at the top center. Below it are two input fields: 'Username' and 'Password'. At the bottom, there are two buttons: 'Exit' and 'Login'.

Gidaw Case

Username

Password

Exit Login

Mockup Login

Logo	Tombol Admin	Tombol Barang	Tombol Transaksi	Tombol Log Out
------	--------------	---------------	------------------	----------------

Setting Admin

ID_Admin

Username

Password

Tabel Admin

Mockup Admin

Logo	Tombol Admin	Tombol Barang	Tombol Transaksi	Tombol Log Out
------	--------------	---------------	------------------	----------------

Setting Admin

ID_Admin

Nama Barang

Harga

Stock

Tabel Barang

Mockup Barang

Logo	Tombol Admin	Tombol Barang	Tombol Transaksi	Tombol Log Out
------	--------------	---------------	------------------	----------------

Setting Admin

Tanggal

ID Barang

Nama Barang

Harga

Jumlah

Total

List Barang

Tabel Barang

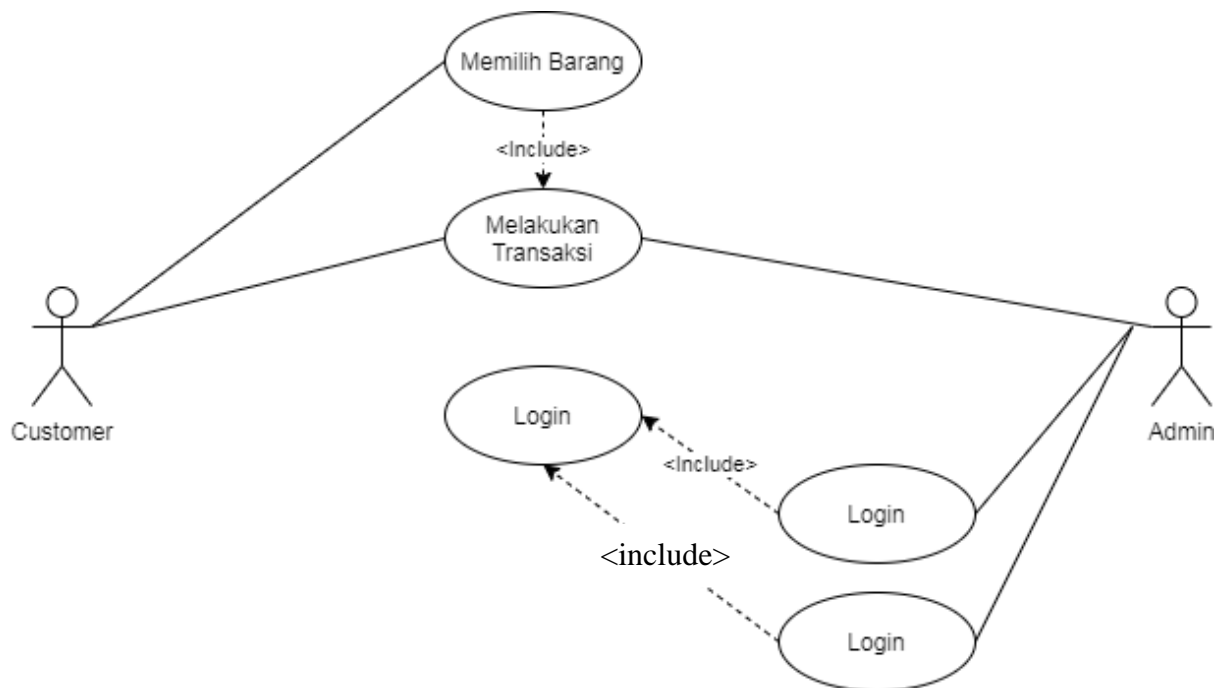
Riwayat Transaksi

Tabel Barang

Mockup Transaksi

3.8 Use Case Diagram

Use Case diagram adalah suatu urutan interaksi yang saling berkaitan antara sistem dan aktor. Use case dijalankan melalui cara menggambarkan tipe interaksi antara user suatu program (sistem) dengan sistemnya sendiri.



BAB IV IMPLEMENTASI

3.1 Package com

1. Class login

```
package com;

public class Login {
    private String ID_Admin;
    private String Username;
    private String Password;

    public Login() {
    }

    public Login(String Username, String Password) {
        this.Username = Username;
        this.Password = Password;
    }

    public Login(String ID_Admin, String Username, String Password) {
        this.ID_Admin = ID_Admin;
        this.Username = Username;
        this.Password = Password;
    }

    public String getID_Admin() {
        return ID_Admin;
    }

    public void setID_Admin(String ID_Admin) {
        this.ID_Admin = ID_Admin;
    }
}
```

```

public String getUsername() {
    return Username;
}

public void setUsername(String Username) {
    this.Username = Username;
}

public String getPassword() {
    return Password;
}

public void setPassword(String Password) {
    this.Password = Password;
}
}

```

2. Class aksesorisHp

```

package com;

public class aksesorisHP {
    private String ID_Barang;
    private String Nama_Barang;
    private int Harga;
    private int Stock;

    public aksesorisHP() {
    }

    public aksesorisHP(String ID_Barang) {
        this.ID_Barang = ID_Barang;
    }

    public aksesorisHP(String ID_Barang, String Nama_Barang, int Harga, int Stock)
    {

```



```
this.ID_Barang = ID_Barang;
this>Nama_Barang = Nama_Barang;
this.Harga = Harga;
this.Stock = Stock;
}

public String getID_Barang() {
    return ID_Barang;
}

public void setID_Barang(String ID_Barang) {
    this.ID_Barang = ID_Barang;
}

public String getNama_Barang() {
    return>Nama_Barang;
}

public void setNama_Barang(String>Nama_Barang) {
    this>Nama_Barang =>Nama_Barang;
}

public int getHarga() {
    return Harga;
}

public void setHarga(int Harga) {
    this.Harga = Harga;
}

public int getStock() {
    return Stock;
}
```

```

    public void setStock(int Stock) {
        this.Stock = Stock;
    }
}

```

3. Class Transaksi

```

package com;

public class Transaksi {

    private int ID_Transaksi;
    private String Tanggal;
    private String ID_Barang;
    private int Jumlah;
    private int Total;

    public Transaksi() {

    }

    public Transaksi(String Tanggal, String ID_Barang, int Jumlah, int Total) {
        this.Tanggal = Tanggal;
        this.ID_Barang = ID_Barang;
        this.Jumlah = Jumlah;
        this.Total = Total;
    }

    public Transaksi(int ID_Transaksi, String Tanggal, String Nama_Barang, int
    Jumlah, int Total) {
        this.ID_Transaksi = ID_Transaksi;
        this.Tanggal = Tanggal;
        this.ID_Barang = ID_Barang;
        this.Jumlah = Jumlah;
        this.Total = Total;
    }
}

```

```
public int getID_Transaksi() {  
    return ID_Transaksi;  
}
```

```
public void setID_Transaksi(int ID_Transaksi) {  
    this.ID_Transaksi = ID_Transaksi;  
}
```

```
public String getTanggal() {  
    return Tanggal;  
}
```

```
public void setTanggal(String Tanggal) {  
    this.Tanggal = Tanggal;  
}
```

```
public String getID_Barang() {  
    return ID_Barang;  
}
```

```
public void setID_Barang(String ID_Barang) {  
    this.ID_Barang = ID_Barang;  
}
```

```
public int getJumlah() {  
    return Jumlah;  
}
```

```
public void setJumlah(int Jumlah) {  
    this.Jumlah = Jumlah;  
}
```

```
public int getTotal() {  
    return Total;  
}
```

```

    }

    public void setTotal(int Total) {
        this.Total = Total;
    }
}

```

4. Class ConvertLoginToObject

```

package com;

import exec.ExecuteLogin;
import java.util.ArrayList;
import java.util.List;

public class ConvertLoginToObject {
    public String[][] getAllLogin(){
        List<Login> myLg = new ArrayList<Login>();
        ExecuteLogin eLg = new ExecuteLogin();
        myLg = eLg.getAllLogin();
        String[][] dataLogin = new String[myLg.size()][3];
        int i=0;
        for(Login lg : myLg){
            dataLogin[i][0] = lg.getID_Admin();
            dataLogin[i][1] = lg.getUsername();
            dataLogin[i][2] = lg.getPassword();
            i++;
        }
        return dataLogin;
    }
}

```

5. Class ConvertaksesorisHpToObject

```
package com;

import exec.ExecuteaksesorisHP;
import java.util.ArrayList;
import java.util.List;

public class ConvertaksesorisHPToObject {
    public String[][] getAllaksesorisHP(){
        List<aksesorisHP> myAh = new ArrayList<aksesorisHP>();
        ExecuteaksesorisHP eAh = new ExecuteaksesorisHP();
        myAh = eAh.getAllaksesorisHP();
        String[][] dataAksesorisHP = new String[myAh.size()][4];
        int i=0;
        for(aksesorisHP ah : myAh){
            dataAksesorisHP[i][0] = ah.getID_Barang();
            dataAksesorisHP[i][1] = ah.getNama_Barang();
            dataAksesorisHP[i][2] = ""+ah.getHarga();
            dataAksesorisHP[i][3] = ""+ah.getStock();
            i++;
        }
        return dataAksesorisHP;
    }
}
```

6. Class ConvertTransaksiToObject

```
package com;

import exec.ExecuteTransaksi;
import java.util.ArrayList;
import java.util.List;

public class ConvertTransaksiToObject {
    public String[][] getAllTransaksi(){
```

```
List<Transaksi> myTs = new ArrayList<Transaksi>();
ExecuteTransaksi eTs = new ExecuteTransaksi();
myTs = eTs.getAllTransaksi();
String[][] dataTransaksi = new String[myTs.size()][5];
int i=0;
for(Transaksi ts : myTs){
    dataTransaksi[i][0] = ""+ts.getID_Transaksi();
    dataTransaksi[i][1] = ts.getTanggal();
    dataTransaksi[i][2] = ts.getID_Barang();
    dataTransaksi[i][3] = ""+ts.getJumlah();
    dataTransaksi[i][4] = ""+ts.getTotal();
    i++;
}
return dataTransaksi;
}
}
```

3.2 Package db

1. Class ConnectionManager

```
package db;

import java.sql.Connection;
import java.sql.DriverManager;

public class connectionManager {
    private Connection con;
    private String driver = "com.mysql.jdbc.Driver";
    private String url = "jdbc:mysql://localhost:3306/gidaw_case"; // myDB --> nama
    database kita
    private String username = "root"; // user name DMBS
    private String password = ""; // pswd DMBS
    public Connection logOn(){
        try {
            //Load JDBC Driver
            Class.forName( driver ).newInstance();
            //Buat object Connection
            con = DriverManager.getConnection( url, username, password );
        }
        catch(Exception ex){
            ex.printStackTrace();
        }
        return con;
    }
    public void logOff(){
        try {
            //Tutup Koneksi
            con.close();
        }
        catch(Exception ex){
            ex.printStackTrace();
        }
    }
}
```

3.3 Package exec

1. executeLogin

```
package exec;

import com.Login;
import db.connectionManager;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ExecuteLogin {
    // Read -> Select
    public List<Login> getAllLogin(){
        String query = "select * from admin";
        connectionManager conMan = new connectionManager();
        List<Login> lslogin = new ArrayList<Login>();
        Connection conn = conMan.logOn();
        try {
            Statement stm = conn.createStatement();
            ResultSet rs = stm.executeQuery(query);
            while(rs.next()){
                Login lg = new Login();
                lg.setID_Admin(rs.getString("ID_Admin"));
                lg.setUsername(rs.getString("Username"));
                lg.setPassword(rs.getString("Password"));
                lslogin.add(lg);
            }
        } catch (SQLException ex) {
            Logger.getLogger(ExecuteLogin.class.getName()).log(Level.SEVERE, null,
                ex);
        }
        conMan.logOff();
        return lslogin;
    }

    // Insert -> Create / Membuat
    public int insertData(Login lg){
        int hasil = 0;
        String query = "Insert into admin(ID_Admin, Username, Password)values"
            + "("+lg.getID_Admin()+",""+lg.getUsername()+",""+lg.getPassword()+")";
        connectionManager conMan = new connectionManager();
```



```

Connection conn = conMan.logOn();
try {
    Statement stm = conn.createStatement();
    hasil = stm.executeUpdate(query);
} catch (SQLException ex) {
    Logger.getLogger(ExecuteLogin.class.getName()).log(Level.SEVERE, null,
        ex);
}
conMan.logOff();
return hasil;
}

```

// Delete -> Hapus

```

public int hapusData(String ID_Admin){
    String query = "Delete from admin where ID_Admin='"+ID_Admin+"'";
    int hasil = 0;
    connectionManager conMan = new connectionManager();
    Connection conn = conMan.logOn();
    try {
        Statement stm = conn.createStatement();
        hasil = stm.executeUpdate(query);
    } catch (SQLException ex) {
        Logger.getLogger(ExecuteLogin.class.getName()).log(Level.SEVERE, null,
            ex);
    }
    conMan.logOff();
    return hasil;
}

```

// Update -> Ubah

```

public int ubahData(Login lg){
    int hasil = 0;
    connectionManager conMan = new connectionManager();
    Connection conn = conMan.logOn();
    String query = "Update admin set Username='"+lg.getUsername()+"',
        Password='"+lg.getPassword()+"' "+ " where ID_Admin='"+
        +lg.getID_Admin()+"'";
    try {
        Statement stm = conn.createStatement();
        hasil = stm.executeUpdate(query);
    } catch (SQLException ex) {
        Logger.getLogger(ExecuteLogin.class.getName()).log(Level.SEVERE, null,
            ex);
    }
    return hasil;
}

```

```

//Login
public int fungsiLogin(Login login){
    /*
        hasil = 0 : uname / pass invalid
        hasil = -1 : pass invalid
        hasil = 1 : pass
    */
    int hasil = 0;
    String query = "SELECT * FROM admin Where Username = '"+
login.getUsername()+"'";
    connectionManager conMan = new connectionManager();
    Connection con = conMan.logOn();
    try {
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(query);
        while (rs.next()) {
            if (!rs.getString("password").equals(login.getPassword())){
                hasil = -1;
            } else if (rs.getString("password").equals(login.getPassword())) {
                login.setID_Admin(rs.getString("id_admin"));
                login.setUsername(rs.getString("username"));
                login.setPassword(rs.getString("password"));
                hasil = 1;
            } else {
                hasil = 0;
            }
        }
    } catch (SQLException ex) {
        java.util.logging.Logger.getLogger(ExecuteLogin.class.getName()).
            log(Level.SEVERE, null, ex);
    }
    return hasil;
}
}

```

2. executeaksesorisHp

```

package exec;

import com.aksesorisHP;
import db.connectionManager;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

```

```

import java.util.logging.Level;
import java.util.logging.Logger;

public class ExecuteaksesorisHP {
    // Read -> select
    public List<aksesorisHP> getAllaksesorisHP(){
        String query = "select * from aksesoris_hp";
        connectionManager conMan = new connectionManager();
        List<aksesorisHP> lsAksesorisHP = new ArrayList<aksesorisHP>();
        Connection conn = conMan.logOn();
        try {
            Statement stm = conn.createStatement();
            ResultSet rs = stm.executeQuery(query);
            while(rs.next()){
                aksesorisHP ah = new aksesorisHP();
                ah.setID_Barang(rs.getString("ID_Barang"));
                ah.setNama_Barang(rs.getString("Nama_Barang"));
                ah.setHarga(rs.getInt("Harga"));
                ah.setStock(rs.getInt("Stock"));
                lsAksesorisHP.add(ah);
            }
        } catch (SQLException ex) {
            Logger.getLogger(ExecuteaksesorisHP.class.getName()).log(Level.SEVERE,
                null, ex);
        }
        conMan.logOff();
        return lsAksesorisHP;
    }

    //Insert Data
    public int insertData(aksesorisHP ah){
        int hasil = 0;
        String query = "insert into aksesoris_hp(ID_Barang, Nama_Barang, Harga,
            Stock)values"+"(""+ah.getID_Barang()+""+""+ah.getNama_Barang()+""+""
            +ah.getHarga()+""+""+ah.getStock()+""+"";
        connectionManager conMan = new connectionManager();
        Connection conn = conMan.logOn();
        try {
            Statement stm = conn.createStatement();
            hasil = stm.executeUpdate(query);
        } catch (SQLException ex) {

            Logger.getLogger(ExecuteaksesorisHP.class.getName()).log(Level.SEVERE, null,
                ex);
        }
        conMan.logOff();
    }
}

```

```

        return hasil;
    }

    //Delete Data
    public int hapusData(String delID_Barang){
        String query = "Delete from aksesoris_hp where ID_Barang='"+
            delID_Barang+"'";
        int hasil = 0;
        connectionManager conMan = new connectionManager();
        Connection conn = conMan.logOn();
        try {
            Statement stm = conn.createStatement();
            hasil = stm.executeUpdate(query);
        } catch (SQLException ex) {

            Logger.getLogger(ExecuteaksesorisHP.class.getName()).log(Level.SEVERE, null,
                ex);
        }
        conMan.logOff();
        return hasil;
    }

    //Update Data
    public int ubahData(aksesorisHP ah){
        int hasil = 0;
        connectionManager conMan = new connectionManager();
        Connection conn = conMan.logOn();
        String query = "Update aksesoris_hp set
            Nama_Barang='"+ah.getNama_Barang()+"',"
            + " Harga='"+ah.getHarga()+"', Stock='"+ah.getStock()+" where
            ID_Barang='"+ah.getID_Barang()+"'";
        try {
            Statement stm = conn.createStatement();
            hasil = stm.executeUpdate(query);
        } catch (SQLException ex) {

            Logger.getLogger(ExecuteaksesorisHP.class.getName()).log(Level.SEVERE, null,
                ex);
        }
        return hasil;
    }
}

```

3. executeTransaksi

```

package exec;

import com.Transaksi;
import db.connectionManager;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;

public class ExecuteTransaksi {
    // Read -> Select
    public List<Transaksi> getAllTransaksi(){
        String query = "select * from transaksi inner join aksesoris_hp on
            aksesoris_hp.ID_Barang = transaksi.ID_Barang";
        connectionManager conMan = new connectionManager();
        List<Transaksi> lstrans = new ArrayList<Transaksi>();
        Connection conn = conMan.logOn();
        try {
            Statement stm = conn.createStatement();
            ResultSet rs = stm.executeQuery(query);
            while(rs.next()){
                Transaksi ts = new Transaksi();
                ts.setID_Transaksi(rs.getInt("ID_Transaksi"));
                ts.setTanggal(rs.getString("Tanggal"));
                ts.setID_Barang(rs.getString("ID_Barang"));
                ts.setJumlah(rs.getInt("Jumlah"));
                ts.setTotal(rs.getInt("Total"));
                lstrans.add(ts);
            }
        } catch (SQLException ex) {
            Logger.getLogger(ExecuteTransaksi.class.getName()).log(Level.SEVERE,
                null, ex);
        }
        conMan.logOff();
        return lstrans;
    }

    //Insert Data
    public int insertData(Transaksi ts){
        int hasil = 0;
    }

```

```

String query = "insert into transaksi(ID_Transaksi, Tanggal, ID_Barang,
    Jumlah, Total) values" + "("+ts.getID_Transaksi()+", '"+ts.getTanggal()+",
    '"+ts.getID_Barang()+", "+ ""'+ts.getJumlah()+", "+ts.getTotal()+)";
connectionManager conMan = new connectionManager();
Connection conn = conMan.logOn();
try {
    Statement stm = conn.createStatement();
    hasil = stm.executeUpdate(query);
} catch (SQLException ex) {
    Logger.getLogger(ExecuteTransaksi.class.getName()).log(Level.SEVERE,
        null, ex);
}
conMan.logOff();
return hasil;
}
}

```

3.4 Package view

1. JFrame FormLogin

```
package view;
```

```
import com.Login;
import exec.ExecuteLogin;
import javax.swing.JOptionPane;
```

```
/**
 *
 * @author Agni Pangestu
 */
```

```
public class FormLogin extends javax.swing.JFrame {
```

```
    /**
     * Creates new form FormLogin
     */
```

```
    public FormLogin() {
        initComponents();
    }
```

```
private void btn_loginActionPerformed(java.awt.event.ActionEvent evt) {
    String username, password;
    int hasil = 0;
    username = txtUsername.getText();
    password = txtPassword.getText();

    Login login = new Login();
    login.setUsername(username);
    login.setPassword(password);

    ExecuteLogin eLogin = new ExecuteLogin();
    hasil = eLogin.fungsiLogin(login);

    switch (hasil) {
        case 1:
            JOptionPane.showMessageDialog(this, "Selamat Datang, " + login.getUsername(), "Info", JOptionPane.INFORMATION_MESSAGE);
            new FormData().setVisible(true);
            this.dispose();
            break;
        case 0:
            JOptionPane.showMessageDialog(this, "Username Tidak Ditemukan!");
            break;
        case -1:
            JOptionPane.showMessageDialog(this, "Password Salah");
            break;
        default:
            JOptionPane.showMessageDialog(this, "Data Tidak Valid!");
            break;
    }
}
```

```
private void btn_exitActionPerformed(java.awt.event.ActionEvent evt) {
    int exit = JOptionPane.showConfirmDialog(this, "Keluar Aplikasi?", "Exit App", JOptionPane.YES_NO_OPTION);
    if (exit == JOptionPane.YES_NO_OPTION) {
        System.exit(0);
    }
}
```

2. Jframe FormData

```

package view;

import com.ConvertaksesorisHPTtoObject;
import com.aksesorisHP;
import com.ConvertLoginToObject;
import com.Login;
import com.ConvertTransaksiToObject;
import com.Transaksi;
import java.awt.CardLayout;
import javax.swing.JOptionPane;

/**
 *
 * @author Agni Pangestu
 */
public class FormData extends javax.swing.JFrame {

    /**
     * Creates new form FormData
     */
    public FormData() {
        initComponents();
        setTanggal();
        setDataAdmin();
        setDataAksesorisHP();
        setDataBarang();
        setDataTransaksi();
    }

    private void setDataAdmin() {
        ConvertLoginToObject clto = new ConvertLoginToObject();
        String[][] dataLogin = clto.getAllLogin();
        tbl_admin.setModel(new javax.swing.table.DefaultTableModel(
            dataLogin, new String[] {
                "ID_Admin", "Username", "Password"
            }
        ));
    }

    private void setDataAksesorisHP() {
        ConvertaksesorisHPTtoObject cato = new ConvertaksesorisHPTtoObject();
        String[][] dataAksesorisHP = cato.getAllaksesorisHP();
        tbl_aksesoris.setModel(new javax.swing.table.DefaultTableModel(
            dataAksesorisHP, new String[] {
                "ID_Barang", "Nama_Barang", "Harga", "Stock"
            }
        ));
    }

    private void setDataBarang() {
        ConvertaksesorisHPTtoObject cato = new ConvertaksesorisHPTtoObject();
        String[][] dataAksesorisHP = cato.getAllaksesorisHP();
        tbl_aksesoris.setModel(new javax.swing.table.DefaultTableModel(
            dataAksesorisHP, new String[] {
                "ID_Barang", "Nama_Barang", "Harga", "Stock"
            }
        ));
        tbl_barang.setModel(new javax.swing.table.DefaultTableModel(
            dataAksesorisHP, new String[] {
                "ID_Barang", "Nama_Barang", "Harga", "Stock"
            }
        ));
    }

    private void setDataTransaksi() {
        ConvertTransaksiToObject ctto = new ConvertTransaksiToObject();
        String[][] dataTransaksi = ctto.getAllTransaksi();
        tbl_transaksi.setModel(new javax.swing.table.DefaultTableModel(
            dataTransaksi, new String[] {
                "ID_Transaksi", "Tanggal", "ID_Barang", "Jumlah", "Total"
            }
        ));
    }

    public void setTanggal() {
        java.util.Date skrg = new java.util.Date();
        java.text.SimpleDateFormat kal = new java.text.SimpleDateFormat("dd/MM/yyyy");
        txt_tanggal.setText(kal.format(skrg));
    }
}

```



```

private void tbl_aksesorisMouseClicked(java.awt.event.MouseEvent evt) {
    int row = tbl_aksesoris.getSelectedRow();
    String ID_Barang = tbl_aksesoris.getValueAt(row, 0).toString();
    String Nama_Barang = tbl_aksesoris.getValueAt(row, 1).toString();
    String Harga = tbl_aksesoris.getValueAt(row, 2).toString();
    String Stock = tbl_aksesoris.getValueAt(row, 3).toString();
    txt_id.setText(ID_Barang);
    txt_nama.setText(Nama_Barang);
    txt_harga.setText(Harga);
    txt_stock.setText(Stock);
}

private void btn_aksesorisActionPerformed(java.awt.event.ActionEvent evt) {
    cl = (CardLayout) pnl_utama.getLayout();
    cl.show(pnl_utama, "cardAksesoris");
    setDataBarang();
}

private void btn_createAksesorisActionPerformed(java.awt.event.ActionEvent evt) {
    String ID_Barang = txt_id.getText();
    String Nama_Barang = txt_nama.getText();
    String strHarga = txt_harga.getText();
    String strStock = txt_stock.getText();
    int Harga = Integer.parseInt(strHarga);
    int Stock = Integer.parseInt(strStock);
    aksesorisHP ah = new aksesorisHP(ID_Barang, Nama_Barang, Harga, Stock);
    exec.ExecuteaksesorisHP eAh = new exec.ExecuteaksesorisHP();
    int hasil = eAh.insertData(ah);
    if(hasil > 0){
        JOptionPane.showMessageDialog(null, "Data berhasil disimpan");
        setDataAksesorisHP();
    }
    else{
        JOptionPane.showMessageDialog(null, "Data gagal disimpan");
    }
}

private void btn_updateAksesorisActionPerformed(java.awt.event.ActionEvent evt) {
    String ID_Barang = txt_id.getText();
    String Nama_Barang = txt_nama.getText();
    String strHarga = txt_harga.getText();
    String strStock = txt_stock.getText();
    int Harga = Integer.parseInt(strHarga);
    int Stock = Integer.parseInt(strStock);
    aksesorisHP ah = new aksesorisHP(ID_Barang, Nama_Barang, Harga, Stock);
    exec.ExecuteaksesorisHP eAh = new exec.ExecuteaksesorisHP();
    int hasil = eAh.ubahData(ah);
    if(hasil > 0){
        JOptionPane.showMessageDialog(null, "Data berhasil diubah");
        setDataAksesorisHP();
    }
    else{
        JOptionPane.showMessageDialog(null, "Data gagal diubah");
    }
}

private void btn_deleteaksesorisActionPerformed(java.awt.event.ActionEvent evt) {
    String ID_Barang = txt_id.getText();
    exec.ExecuteaksesorisHP eAh = new exec.ExecuteaksesorisHP();
    int hasil = eAh.hapusData(ID_Barang);
    if(hasil > 0){
        JOptionPane.showMessageDialog(null, "Data berhasil dihapus");
        setDataAksesorisHP();
    }
    else{
        JOptionPane.showMessageDialog(null, "Data gagal dihapus");
    }
}

private void btn_adminActionPerformed(java.awt.event.ActionEvent evt) {
    cl = (CardLayout) pnl_utama.getLayout();
    cl.show(pnl_utama, "cardAdmin");
}

```

```

private void btn_transaksiActionPerformed(java.awt.event.ActionEvent evt) {
    cl = (CardLayout) pnl_utama.getLayout();
    cl.show(pnl_utama, "cardTransaksi");
    setDataBarang();
}

private void btn_logoutActionPerformed(java.awt.event.ActionEvent evt) {
    new FormLogin().setVisible(true);
    dispose();
}

private void tbl_adminMouseClicked(java.awt.event.MouseEvent evt) {
    int row = tbl_admin.getSelectedRow();
    String ID_Admin = tbl_admin.getValueAt(row, 0).toString();
    String Username = tbl_admin.getValueAt(row, 1).toString();
    String Password = tbl_admin.getValueAt(row, 2).toString();
    txt_id_admin.setText(ID_Admin);
    txt_username.setText(Username);
    txt_password.setText>Password);
}

private void btn_updateAdminActionPerformed(java.awt.event.ActionEvent evt) {
    String ID_Admin = txt_id_admin.getText();
    String Username = txt_username.getText();
    String Password = txt_password.getText();
    Login lg = new Login(ID_Admin, Username, Password);
    exec.ExecuteLogin eLg = new exec.ExecuteLogin();
    int hasil = eLg.ubahData(lg);
    if(hasil >0){
        JOptionPane.showMessageDialog(null, "Data berhasil diubah");
        setDataAdmin();
    }
    else{
        JOptionPane.showMessageDialog(null, "Data gagal diubah");
    }
}

private void btn_createAdminActionPerformed(java.awt.event.ActionEvent evt) {
    String ID_Admin = txt_id_admin.getText();
    String Username = txt_username.getText();
    String Password = txt_password.getText();
    Login lg = new Login(ID_Admin, Username, Password);
    exec.ExecuteLogin eLg = new exec.ExecuteLogin();
    int hasil = eLg.insertData(lg);
    if(hasil >0){
        JOptionPane.showMessageDialog(null, "Data berhasil disimpan");
        setDataAdmin();
    }
    else{
        JOptionPane.showMessageDialog(null, "Data gagal disimpan");
    }
}

private void btn_deleteAdminActionPerformed(java.awt.event.ActionEvent evt) {
    String ID_Admin = txt_id_admin.getText();
    exec.ExecuteLogin eLg = new exec.ExecuteLogin();
    int hasil = eLg.hapusData(ID_Admin);
    if(hasil >0){
        JOptionPane.showMessageDialog(null, "Data berhasil dihapus");
        setDataAdmin();
    }
    else{
        JOptionPane.showMessageDialog(null, "Data gagal dihapus");
    }
}

private void tbl_barangMouseClicked(java.awt.event.MouseEvent evt) {
    int row = tbl_barang.getSelectedRow();
    String ID_Barang = tbl_barang.getValueAt(row, 0).toString();
    String Nama_Barang = tbl_barang.getValueAt(row, 1).toString();
    String Harga = tbl_barang.getValueAt(row, 2).toString();
    String Stock = tbl_barang.getValueAt(row, 3).toString();
    txt_idBarang.setText(ID_Barang);
    txt_namaBarang.setText(Nama_Barang);
    txt_harga2.setText(Harga);
}

```

```

] private void btn_hitungActionPerformed(java.awt.event.ActionEvent evt) {
    int jumlah = Integer.parseInt(txt_jumlah.getText());
    int harga = Integer.parseInt(txt_harga2.getText());
    int total = jumlah * harga;
    txt_total.setText(Integer.toString(total));
- }

] private void btn_pesananActionPerformed(java.awt.event.ActionEvent evt) {
    String Tanggal = txt_tanggal.getText();
    String ID_Barang = txt_idBarang.getText();
    String strJumlah = txt_jumlah.getText();
    String strTotal = txt_total.getText();
    int Jumlah = Integer.parseInt(strJumlah);
    int Total = Integer.parseInt(strTotal);
    Transaksi ts = new Transaksi(Tanggal, ID_Barang, Jumlah, Total);
    exec.ExecuteTransaksi eTs = new exec.ExecuteTransaksi();
    int hasil = eTs.insertData(ts);
    if(hasil > 0){
        JOptionPane.showMessageDialog(null, "Pesanan berhasil");
        setDataTransaksi();
        setDataBarang();
    }
    else{
        JOptionPane.showMessageDialog(null, "Pesanan gagal");
    }
- }

```

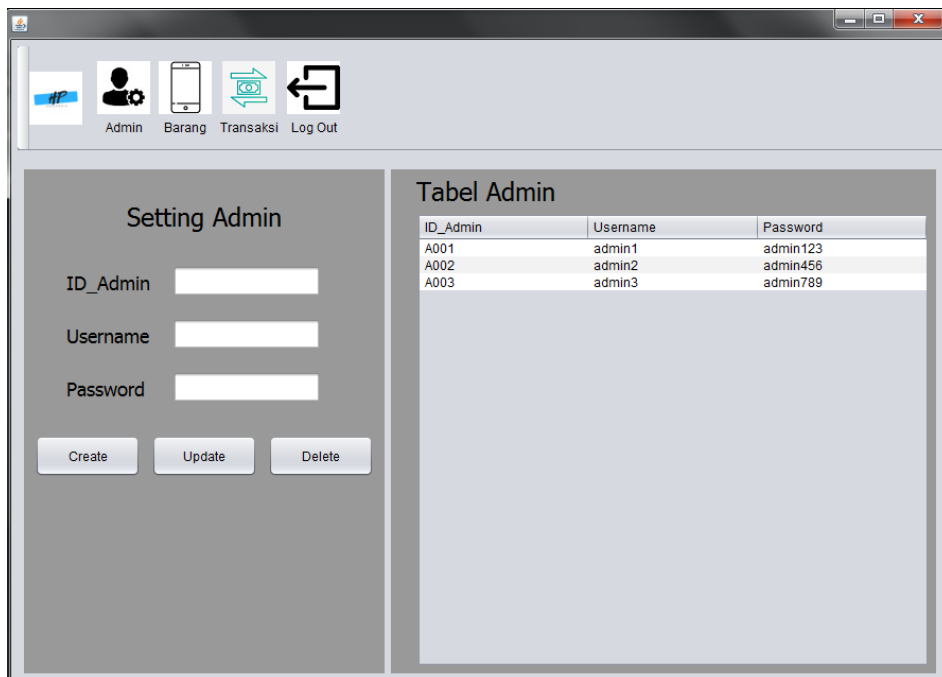
3.5 GUI FormLogin



The screenshot shows a login window titled "GIDAW CASE". It features a "Username" label above a text input field, and a "Password" label above a password input field. Below the input fields are two buttons: "Exit" and "Login".

FormLogIn

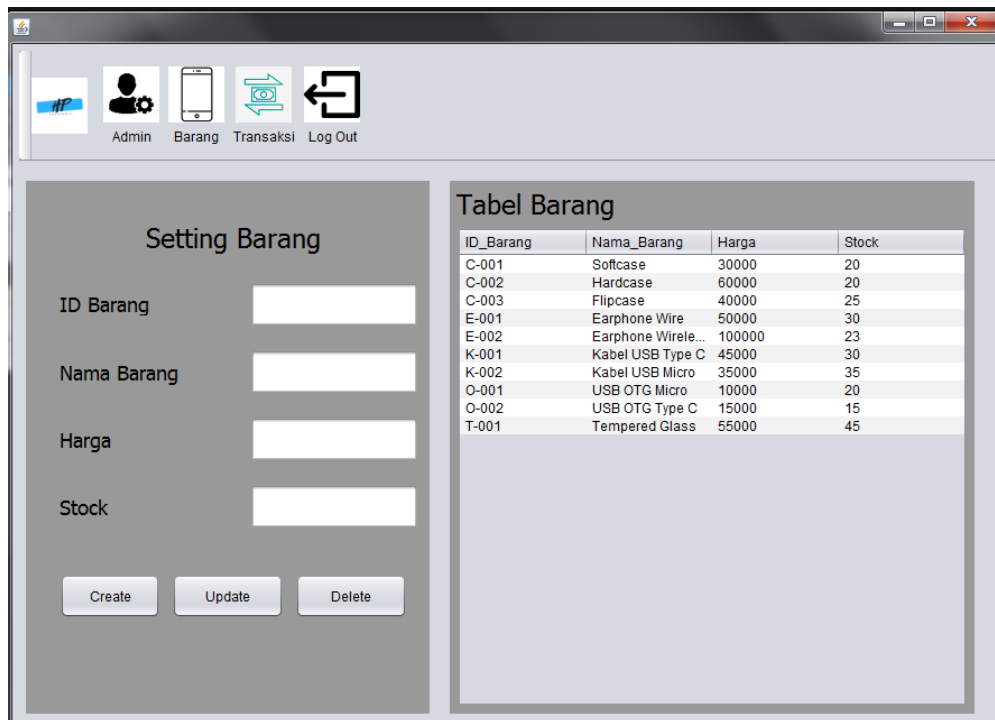
3.6 GUI FormData



The screenshot shows an admin panel titled "GIDAW CASE". At the top, there is a navigation bar with icons and labels for "Admin", "Barang", "Transaksi", and "Log Out". The main area is divided into two sections. The left section, titled "Setting Admin", contains three input fields for "ID_Admin", "Username", and "Password", and three buttons: "Create", "Update", and "Delete". The right section, titled "Tabel Admin", contains a table with three columns: "ID_Admin", "Username", and "Password".

ID_Admin	Username	Password
A001	admin1	admin123
A002	admin2	admin456
A003	admin3	admin789

FormData Panel Admin



Setting Barang

ID Barang

Nama Barang

Harga

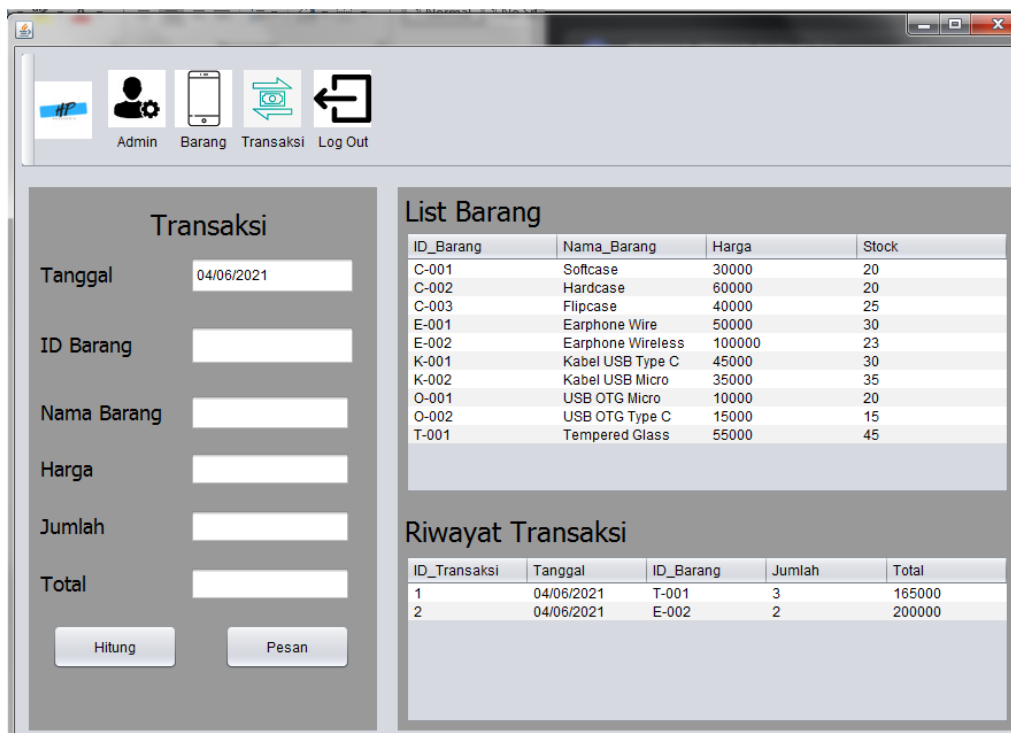
Stock

Create Update Delete

Tabel Barang

ID_Barang	Nama_Barang	Harga	Stock
C-001	Softcase	30000	20
C-002	Hardcase	60000	20
C-003	Flipcase	40000	25
E-001	Earphone Wire	50000	30
E-002	Earphone Wirele...	100000	23
K-001	Kabel USB Type C	45000	30
K-002	Kabel USB Micro	35000	35
O-001	USB OTG Micro	10000	20
O-002	USB OTG Type C	15000	15
T-001	Tempered Glass	55000	45

FormData Panel Barang



Transaksi

Tanggal

ID Barang

Nama Barang

Harga

Jumlah

Total

Hitung Pesan

List Barang

ID_Barang	Nama_Barang	Harga	Stock
C-001	Softcase	30000	20
C-002	Hardcase	60000	20
C-003	Flipcase	40000	25
E-001	Earphone Wire	50000	30
E-002	Earphone Wireless	100000	23
K-001	Kabel USB Type C	45000	30
K-002	Kabel USB Micro	35000	35
O-001	USB OTG Micro	10000	20
O-002	USB OTG Type C	15000	15
T-001	Tempered Glass	55000	45

Riwayat Transaksi

ID_Transaksi	Tanggal	ID_Barang	Jumlah	Total
1	04/06/2021	T-001	3	165000
2	04/06/2021	E-002	2	200000

FormData Panel Transaksi

BAB V PENUTUP

2.1 Kesimpulan

Pada aplikasi ini telah berhasil menyambungkan netbeans ke MySQL. Adapun fitur fitur yang berhasil dibuat adalah fitur create, read, update, dan delete pada tabel yang ada di database aksesoris Hp, kemudian berhasil membuat inisialisasi login untuk masuk ke aplikasi, lalu membuat perhitungan terkait untuk transaksi aplikasi ini.

2.2 Saran

- Aplikasi ini seharusnya bisa memilih banyak barang dalam satu kali transaksi jadi jika ingin membeli barang yang berbeda tidak perlu melakukan 2 kali transaksi.
- Aplikasi ini seharusnya bisa mencetak struk dari transaksi yang dilakukan