

LAPORAN PRAKTIKUM BAHASA PEMROGRAMAN 1

MODUL 9 : iREPORT

Nama : Ilham Hafidz

NIM : 20220810052

Kelas : TINFC-2022-01

TUGAS

Jawab :

File Connect.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import javax.swing.JOptionPane;

public class Connect {
    Connection con;

    public Connect(){
        String id, pass, driver, url;
        id="root";
        pass="";
        driver="com.mysql.jdbc.Driver";
        url="jdbc:mysql://localhost:3306/alope_topup_ff";

        try{
            Class.forName(driver).newInstance();
            con= DriverManager.getConnection(url, id, pass);
            JOptionPane.showMessageDialog(null, "Koneksi Berhasil");
        } catch(Exception e){
            System.out.println(""+e.getLocalizedMessage());
        }
    }

    public static void main(String[] args){
        Connect k= new Connect();
    }
}
```

Analisis

1. Ini merupakan file untuk koneksi project Netbeans Java ke database nya.
2. Pada constructor methodnya (method yang akan dijalankan ketika class diinstance) :
 - i. Inisialisasi variabel id dengan nilai "root". "root" ini merupakan username untuk mengakses SQL phpmyadmin-nya sebagai DBMS.
 - ii. Inisialisasi variabel pass dengan nilai string kosong (""). Ini

merupakan password untuk login ke phpMyAdmin, default passwordnya yaitu string kosong ("")

- iii. Inisialisasi variabel driver dengan nilai "com.mysql.jdbc.Driver". Ini adalah Driver yang digunakan untuk menghubungkan Java Project dengan MySQLnya. Pada project ini saya menggunakan jdbcDriver.
- iv. Inisialisasi variabel url dengan nilai "jdbc:mysql://localhost:3306/alope_topup_ff". Ini merupakan url untuk mengakses database yang ada di phpMyAdminnya.
- v. Membuat sebuah fungsi try catch, dimana jika tidak ada error maka jalankan statement pada try tpi jika ada error jalankan statement yang berada didalam catch :
Statement Try nya adalah Untuk melakukan koneksi dengan menggunakan fungsi DriverManager.getConnection dengan mengirimkan variabel url, id dan pass sebagai parameternya.
Statement Catch nya adalah untuk print message error nya dengan Exeption (handling Error)

File Auth.java

```
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class Auth extends javax.swing.JFrame {
    Connect connect;

    Statement st;
    ResultSet rs;

    public Auth() {
        connect = new Connect();

        initComponents();
    }

    private void resetForm(){
        textFieldRegisterNamaLengkap.setText("");
        textFieldRegisterUsername.setText("");
    }
}
```

```

        textFieldRegisterPassword.setText("");
        textFieldRegisterEmail.setText("");
    }

// initComponents() JFrameform

private void buttonLoginActionPerformed(java.awt.event.ActionEvent evt) {
    String sql =
        "SELECT * FROM accounts WHERE username="
        + textFieldLoginUsername.getText() + " "
        + "AND password ="
        + textFieldLoginPassword.getText() + "";
    try {
        st= connect.con.createStatement();
        rs= st.executeQuery(sql);

        if(rs.next()){
            JOptionPane.showMessageDialog(null, "LOGIN Berhasil!");

            if(rs.getString("role").equals("admin")){
                new CMSAdmin().setVisible(true);
            } else{
                new Beranda().setVisible(true);
            }
        } else{
            JOptionPane.showMessageDialog(null, "Username atau Password salah!");
        }
    } catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}

private void buttonDaftarActionPerformed(java.awt.event.ActionEvent evt) {
    int confirmRegistrasion =
    JOptionPane.showConfirmDialog(
        null,
        "Apakah data anda sudah benar?",
        "Konfirmasi Registrasi",
        JOptionPane.YES_NO_OPTION

```

```

);

if(confirmRegistrasi == 0){
    String sql= "INSERT INTO accounts VALUES(NULL, ""
    + textFieldRegisterNamaLengkap.getText() + "", ""
    + textFieldRegisterUsername.getText() + "", ""
    + textFieldRegisterEmail.getText() + "", ""
    + textFieldRegisterPassword.getText() + "", 'user')";

    try{
        st= connect.con.createStatement();
        st.execute(sql);

        JOptionPane.showMessageDialog(null, "Registrasi Berhasil, Silahkan Login!");
        resetForm();
    } catch(Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
} else{
    JOptionPane.showMessageDialog(
        null,
        "Registrasi Dibatalkan!",
        "BATAL",
        JOptionPane.WARNING_MESSAGE
    );
}

private void buttonBatalActionPerformed(java.awt.event.ActionEvent evt) {
    resetForm();
    JOptionPane.showMessageDialog(
        null,
        "Registrasi Dibatalkan!",
        "BATAL",
        JOptionPane.WARNING_MESSAGE
    );
}

public static void main(String args[]) {

```

```

java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Auth().setVisible(true);
    }
});
}

// Variables declaration JFrameForm
}

```

Analisis

1. Ini merupakan file untuk membuat form untuk login dan register
2. Membuat sebuah method resetForm() yang merupakan fungsi untuk mereset inputan user dengan menggunakan fungsi setText("") pada setiap variabel inputan formnya.
3. Pada fungsi buttonLoginActionPerformed (fungsi yang akan handle jika buttonLogin melakukan aksi / di klik), statementnya adalah :
 - i. Inisialisasi variabel sql dengan nilai string SQL untuk "SELECT" data yang (WHERE) username dan passwordnya sesuai dengan inputan user pada textfield textfieldLoginUsername dan textfieldLoginPassword
 - ii. Membuat sebuah fungsi try catch, dimana jika tidak ada error maka jalankan statement pada try tpi jika ada error jalankan statement yang berada didalam catch
Statement try nya adalah:
 - a) Eksekusi / jalankan query nya dengan fungsi executeQuery dengan mengirimkan variabel sql sebagai argument.
 - b) Melakukan pengecekan jika hasil dari query get datanya ada, maka Tampilkan popup bahwa login berhasil dan lakukan pengecekan kembali jika nilai dari "role" nya adalah admin tampilkan form admin, tetapi jika bukan maka tampilkan form beranda.
 - c) Jika hasil query get datanya tidak ada, maka tampilkan popup message "username atau password salah"
 Statement catch nya adalah : Untuk menampilkan message error dengan memanfaatkan Exception (class untuk error handling)

4. Pada fungsi `buttonDaftarActionPerformed` (Fungsi untuk handle ketika button Daftar melakukan aksi / di klik), statementnya adalah :
- Inisialisasi variabel `confirmRegistrasi` dengan nilai dari return data `Popup Confirm JOptionPane`.
 - Melakukan pengecekan jika `confirmRegistrasi` bernilai 0 (user klik yes) maka :
 - Inisialisasi variabel `sql` dengan nilai SQL untuk INSERT data pada tabel `accounts`.
 - Membuat sebuah fungsi `try catch`, dimana jika tidak ada error maka jalankan statement pada `try` tpi jika ada error jalankan statement yang berada didalam `catch`
Statement `try` nya adalah : untuk mengeksekusi query dari variabel `sql` dan menampilkan popup message bahwa registrasi berhasil.
Statement `catch` nya adalah : Untuk menampilkan message error dengan memanfaatkan `Exception` (class untuk error handling)
Jika bukan 0 maka menampilkan popup registrasi dibatalkan dengan `JOptionPane.showMessageDialog()`
5. Pada fungsi `buttonBatalActionPerformed` (Fungsi untuk handle ketika button batal melakukan aksi / di klik), statement fungsinya adalah :
- Menjalankan fungsi `resetForm`
 - Menampilkan `messageDialog` dengan `JOptionPane` bahwa “registrasi dibatalkan” dengan type message warning menggunakan `JOptionPane.WARNING_MESSAGE`

File Beranda.java

```
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class Beranda extends javax.swing.JFrame {
    Connect connect;
```

```

UserSession userSession;

Statement st;
ResultSet rs;

String OpsiPilihDiamond="";
int idr = 0;

public Beranda() {
    connect = new Connect();
    userSession =new UserSession();

    initComponents();
}

void reset(){
    textFieldEmail.setText("");
    textFieldUserID.setText("");
    textFieldKonfirmasiEmail.setText("");
}

// initComponents() JFrameform

private void radioButton5DMActionPerformed(java.awt.event.ActionEvent evt) {
    OpsiPilihDiamond = "5";
    idr = 1;
}

private void buttonBeliMouseClicked(java.awt.event.MouseEvent evt) {
    int confirm = JOptionPane.showConfirmDialog(null, "Apakah data anda sudah benar", "Konfirmasi
Pembelian", JOptionPane.YES_NO_OPTION);

    if(confirm == 0){

        String sqlGetUserId = "SELECT * FROM accounts WHERE email=" + textFieldEmail.getText()
+"";
        int account_id = 0;

        try{
            st= connect.con.createStatement();

```

```

rs= st.executeQuery(sqlGetUserId);

while(rs.next()){
    account_id= rs.getInt(1);
}

String sql = "INSERT INTO transactions VALUES (NULL, "
    + textFieldUserID.getText() + ", "
    + comboBoxServer.getSelectedItemAt() + ", "
    + OpsiPilihDiamond + ", "
    + idr + ", NULL, "
    + account_id + ", 'proses')";

st.execute(sql);

JOptionPane.showMessageDialog(null, "Topup Berhasil!");
} catch(SQLException e) {
    JOptionPane.showMessageDialog(null, e);
}

} else{
    JOptionPane.showMessageDialog(null, "Topup Dibatalkan!");
}
}

private void radioButton50DMActionPerformed(java.awt.event.ActionEvent evt) {
    OpsiPilihDiamond = "50";
    idr = 7;
}

private void radioButton80DMActionPerformed(java.awt.event.ActionEvent evt) {
    OpsiPilihDiamond = "80";
    idr = 11;
}

private void radioButton100DMActionPerformed(java.awt.event.ActionEvent evt) {
    OpsiPilihDiamond = "100";
    idr = 14;
}

```



```

private void radioButton160DMActionPerformed(java.awt.event.ActionEvent evt) {
    OpsiPilihDiamond = "160";
    idr = 22;
}

private void radioButton210DMActionPerformed(java.awt.event.ActionEvent evt) {
    OpsiPilihDiamond = "210";
    idr = 27;
}

private void radioButton280DMActionPerformed(java.awt.event.ActionEvent evt) {
    OpsiPilihDiamond = "280";
    idr = 36;
}

private void radioButton1075DMActionPerformed(java.awt.event.ActionEvent evt) {
    OpsiPilihDiamond = "1075";
    idr = 135;
}

private void buttonKonfirmasiActionPerformed(java.awt.event.ActionEvent evt) {
    Object header[] = {"User ID", "Server", "Diamond", "Harga", "Status"};
    DefaultTableModel data = new DefaultTableModel(null, header);
    tabelDataTransaksi.setModel(data);

    String sqlGetUserByEmail = "SELECT * FROM accounts WHERE email='" +
    textFieldKonfirmasiEmail.getText() + "'";
    int account_id = 0;

    try {
        st = connect.con.createStatement();
        rs = st.executeQuery(sqlGetUserByEmail);

        while(rs.next()){
            account_id = rs.getInt(1);
        }

        try {
            String sqlGetTransaction = "SELECT * FROM transactions WHERE account_id="

```

```

+account_id;

        st= connect.con.createStatement();
        rs= st.executeQuery(sqlGetTransaction);

        while(rs.next()){
            String col1= rs.getString(2);
            String col2= rs.getString(3);
            String col3= rs.getString(4);
            String col4= "Rp " + rs.getString(5) + ".000";
            String col5= rs.getString(8);

            String k[]= {col1, col2, col3, col4, col5};
            data.addRow(k);
        }

    } catch(SQLException e) {
        JOptionPane.showMessageDialog(null, e);
    }
} catch(SQLException e) {
    JOptionPane.showMessageDialog(null, e);
}

reset();
}

private void buttonLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    new Beranda().setVisible(false);
    new Auth().setVisible(true);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Beranda().setVisible(true);
        }
    });
}

// Variables declaration JFrameForm
}

```

Analisis

1. Ini adalah file form untuk user melakukan top up dan melihat data transaksi user tersebut
2. Inisialisasi variabel OpsiPilihDiamond dengan nilai string kosong (“”)
3. Inisialisasi variabel idr dengan nilai 0
4. Membuat fungsi reset sebagai fungsi untuk mereset form inputan user
5. Fungsi radioButton5DMActionPerformed merupakan fungsi yang akan dijalankan / handle radioButton5DM ketika di klik, statementnya adalah untuk mengubah nilai dari variabel OpsiPilihDiamond menjadi “5” dan idr menjadi 1.
6. Fungsi radioButton50DMActionPerformed merupakan fungsi yang akan dijalankan / handle radioButton50DM ketika di klik, statementnya adalah untuk mengubah nilai dari variabel OpsiPilihDiamond menjadi “50” dan idr menjadi 7.
7. Fungsi radioButton80DMActionPerformed merupakan fungsi yang akan dijalankan / handle radioButton80DM ketika di klik, statementnya adalah untuk mengubah nilai dari variabel OpsiPilihDiamond menjadi “80” dan idr menjadi 11.
8. Fungsi radioButton100DMActionPerformed merupakan fungsi yang akan dijalankan / handle radioButton100DM ketika di klik, statementnya adalah untuk mengubah nilai dari variabel OpsiPilihDiamond menjadi “100” dan idr menjadi 14.
9. Fungsi radioButton210DMActionPerformed merupakan fungsi yang akan dijalankan / handle radioButton210DM ketika di klik, statementnya adalah untuk mengubah nilai dari variabel OpsiPilihDiamond menjadi “210” dan idr menjadi 27.
10. Fungsi radioButton280DMActionPerformed merupakan fungsi yang akan dijalankan / handle radioButton280DM ketika di klik, statementnya adalah untuk mengubah nilai dari variabel OpsiPilihDiamond menjadi “280” dan idr menjadi 36.
11. Fungsi radioButton1075DMActionPerformed merupakan fungsi yang akan dijalankan / handle radioButton1075DM ketika di klik, statementnya adalah untuk mengubah nilai dari variabel OpsiPilihDiamond menjadi “1075” dan idr menjadi 135.
12. Fungsi buttonBeliMouseClicked merupakan fungsi yang akan handle ketika buttonBeli di klik. Statement fungsinya adalah :

- i. Inisialisasi variabel confirm dengan nilai return dari confirmDialog JOptionPane
 - ii. Melakukan pengecekan jika confirm bernilai 0 (User pilih yes pada pop up confirm), maka :
 - a) Inisialisasi variabel sqlGetUserId dengan string SQL untuk get (SELECT) data accounts yang (WHERE) emailnya sama dengan inputan user pada textFieldEmail.
 - b) Inisialisasi variabel account_id dengan nilai 0.
 - c) Membuat sebuah fungsi try catch, dimana jika tidak ada error maka jalankan statement pada try tpi jika ada error jalankan statement yang berada didalam catch
Statement try-nya adalah :
 - Mengeksekusi query dari sqlGetUserId.
 - Mengubah value dari variabel account_id dengan hasil dari get data index ke 1 (index id pada kolom accounts)
 - Inisialisasi variabel sql dengan nilai query SQL untuk INSERT data ke tabel transactions
 - Menampilkan popup message bahwa “Topup Berhasil”
 Statement catch nya adalah : Untuk menampilkan message error dengan memanfaatkan SQLException (class untuk error handling)
13. Fungsi buttonKonfirmasiActionPerformed merupakan fungsi untuk menghandle ketika button konfirmasi melakukan aksi atau di klik, statementnya adalah :
- i. Inisialisasi variabel sqlGetUserByEmail dengan string SQL untuk get (SELECT) data accounts yang (WHERE) emailnya sama dengan inputan user pada textField konfirmasiEmail.
 - ii. Membuat sebuah fungsi try catch, dimana jika tidak ada error maka jalankan statement pada try tpi jika ada error jalankan statement yang berada didalam catch
Statement try nya adalah untuk menjalankan perintah query untuk getData transaksi dengan (WHERE) account_id nya sama dengan id accounts user yang konfirmasi email lalu menampilkannya pada tabel dengan fungsi addRow.
Statement catch nya adalah : Untuk menampilkan message error dengan memanfaatkan SQLException (class untuk error handling)
14. Fungsi buttonLogoutActionPerformed merupakan fungsi untuk menghandle button logout ketika melakukan aksi / di klik, statement fungsinya adalah : Menampilkan form Auth dengan setVisible(true).

File CMSAdmin.java

```
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.view.JasperViewer;

public class CMSAdmin extends javax.swing.JFrame {

    Statement st;
    ResultSet rs;
    Connect connect;

    public CMSAdmin() {
        connect = new Connect();

        initComponents();

        getDataTransaksi();
        getDataUser();
    }

    private void getDataTransaksi(){
        Object header[]= { "ID", "User ID", "Server", "Total Diamond", "Total Bayar", "Nama Pembeli",
        "Email", "Status"};

        DefaultTableModel data= new DefaultTableModel(null, header);
        tabelDataTransaksi.setModel(data);

        String sql= "SELECT * FROM transactions INNER JOIN accounts ON transactions.account_id =
accounts.id";

        try{
            st= connect.con.createStatement();
            rs= st.executeQuery(sql);

            while(rs.next()){
```

```

        String col1= rs.getString(2);
        String col2= rs.getString(3);
        String col3= rs.getString(4);
        String col4= rs.getString(5);
        String col5= rs.getString("nama");
        String col6= rs.getString("email");
        String col7= rs.getString("status");
        String col8= rs.getString(1);

        String k[]= {col8, col1, col2, col3, col4, col5, col6, col7};
        data.addRow(k);
    }
} catch(SQLException e) {
    JOptionPane.showMessageDialog(null, e);
}
}

private void getDataUser(){
    Object header[]= {"ID", "Nama", "Username", "Email", "Password"};
    DefaultTableModel data= new DefaultTableModel(null, header);
    tabelDataUser.setModel(data);

    String sql= "SELECT * FROM accounts";

    try{
        st= connect.con.createStatement();
        rs= st.executeQuery(sql);

        while(rs.next()){
            String col1= rs.getString(1);
            String col2= rs.getString(2);
            String col3= rs.getString(3);
            String col4= rs.getString(4);
            String col5= rs.getString(5);

            String k[]= {col1, col2, col3, col4, col5};
            data.addRow(k);
        }
    } catch(SQLException e) {

```

```

        JOptionPane.showMessageDialog(null, e);
    }
}

//initComponents() JFrameForm

private void buttonEditActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String sql=
            "UPDATE transactions SET "
            + "status=" + comboBoxStatus.getSelectedItem()+ " "
            + "WHERE id=" + textFieldTransaksi.getText() + "";

        st.executeUpdate(sql);

        JOptionPane.showMessageDialog(null, "Data Transaksi berhasil di update");

        getDataTransaksi();
    } catch (SQLException e){
        JOptionPane.showMessageDialog(null, e);
    }
}

private void buttonLogoutActionPerformed(java.awt.event.ActionEvent evt) {
    new CMSAdmin().setVisible(false);
    new Auth().setVisible(true);
}

private void buttonCetakUserActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String reportPath = "src/ireport/ReportDataUser.jasper";

        JasperPrint jasperPrint = JasperFillManager.fillReport(reportPath, null, connect.con);

        JasperViewer.viewReport(jasperPrint, false);
    } catch (Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}
}

```

```

private void btnCetakTransaksiActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String reportPath = "src/ireport/ReportDataTransaksi.jasper";

        JasperPrint jasperPrint = JasperFillManager.fillReport(reportPath, null, connect.con);

        JasperViewer.viewReport(jasperPrint, false);
    } catch (Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
}

private void buttonDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    int confirmDelete =
        JOptionPane.showConfirmDialog(
            null,
            "Apakah Yakin Ingin Menghapusnya?",
            "Konfirmasi Hapus",
            JOptionPane.YES_NO_OPTION
        );
    if(confirmDelete == 0){
        try{
            st= connect.con.createStatement();
            String sql = "DELETE FROM transactions WHERE id= " + textFieldTransaksi.getText() + "";

            st.executeUpdate(sql);

            JOptionPane.showMessageDialog(null, "Data transaksi berhasil dihapus");
        } catch (SQLException e){
            JOptionPane.showMessageDialog(null, e);
        }
    } else{
        JOptionPane.showMessageDialog(
            null,
            "Data transaksi batal dihapus",
            "Batal Hapus",
            JOptionPane.WARNING_MESSAGE
        );
    }
}

```



```

    }

}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new CMSAdmin().setVisible(true);
        }
    });
}

// Variables declaration JFrameForm

}

```

Analisis :

1. File ini merupakan file form untuk admin.
2. Membuat fungsi getDataTransaksi sebagai fungsi untuk getData dari tabel transaksi, Statement fungsinya adalah :
 - i. Inisialisasi variabel sql dengan nilai syntax SQL untuk SELECT data transactions dengan gabungan (INNER JOIN) data dari tabel accounts yang id-nya sama dengan account_id pada tabel transaction
 - ii. sebuah fungsi try catch, dimana jika tidak ada error maka jalankan statement pada try tpi jika ada error jalankan statement yang berada didalam catch
 Stament try-nya adalah : untuk mengeksekusi query dari variabel sql dan memasukkan datanya ke dalam tabel tabelDataTransaksi dengan fungsi addRow()
 Statement catch nya adalah : Untuk menampilkan message error dengan memanfaatkan SQLException (class untuk error handling)
3. Membuat fungsi getDataUser sebagai fungsi untuk getData dari tabel accounts, Statement fungsinya adalah :
 - i. Inisialisasi variabel sql dengan nilai syntax SQL untuk SELECT data tabel accounts.
 - ii. sebuah fungsi try catch, dimana jika tidak ada error maka jalankan statement pada try tpi jika ada error jalankan statement yang berada

didalam catch

Statement try-nya adalah : untuk mengeksekusi query dari variabel sql dan memasukkan datanya ke dalam tabel tabelDataUser dengan fungsi addRow()

Statement catch nya adalah : Untuk menampilkan message error dengan memanfaatkan SQLException (class untuk error handling)

4. Fungsi buttonEditActionPerformed merupakan fungsi yang akan dijalankan ketika tombol edit melakukan aksi / di klik. Statementnya adalah Untuk melakukan edit pada data transaksi yang idnya sesuai dengan inputan user pada textField textFieldTransaksi
5. Fungsi buttonEditActionPerformed Merupakan fungsi yang akan dijalankan ketika tombol edit melakukan aksi / di klik. Statementnya adalah untuk menghapus transaksi yang id nya sama dengan nilai dari form input textFieldTransaksi.
6. Fungsi buttonCetakUserActionPerformed merupakan fungsi yang akan dijalankan ketika tombol cetak melakukan aksi / di klik. Statementnya adalah Untuk melakukan print dengan fungsi JasperPrint dengan print path nya "src/ireport/ReportDataUser.jasper".
7. Fungsi buttnCetakTransaksiActionPerformed merupakan fungsi yang akan dijalankan ketika tombol cetak melakukan aksi / di klik. Statementnya adalah Untuk melakukan print dengan fungsi JasperPrint dengan print path nya "src/ireport/ReportDataTransaksi.jasper".

Output:

- Auth.java

Merupakan file untuk menampilkan form Login dan Registrasi



A screenshot of a Java Swing window titled "LOGIN ALOPE TOPUP FF". The window has a title bar with standard OS controls. Below the title bar, there are two tabs: "LOGIN" (selected) and "REGISTER". The main content area features a vertical image of a Free Fire character on the left and a login form on the right. The form includes labels for "Username" and "Password", each followed by a text input field. At the bottom of the form are two buttons: "BATAL" (Cancel) and "LOGIN".



A screenshot of a Java Swing window titled "REGISTER ALOPE TOPUP FF". The window has a title bar with standard OS controls. Below the title bar, there are two tabs: "LOGIN" and "REGISTER" (selected). The main content area features a horizontal banner image at the top with the text "SAATNYA PETUALANGAN DINGINMU DIMULAH!". Below the banner is a registration form with four fields: "Nama Lengkap" (Full Name), "Email", "Username", and "Password". Each field has a corresponding text input box. At the bottom of the form are two buttons: "BATAL" (Cancel) and "DAFTAR" (Register).

- Beranda.java
Merupakan panel form yang akan muncul ketika login sebagai user.

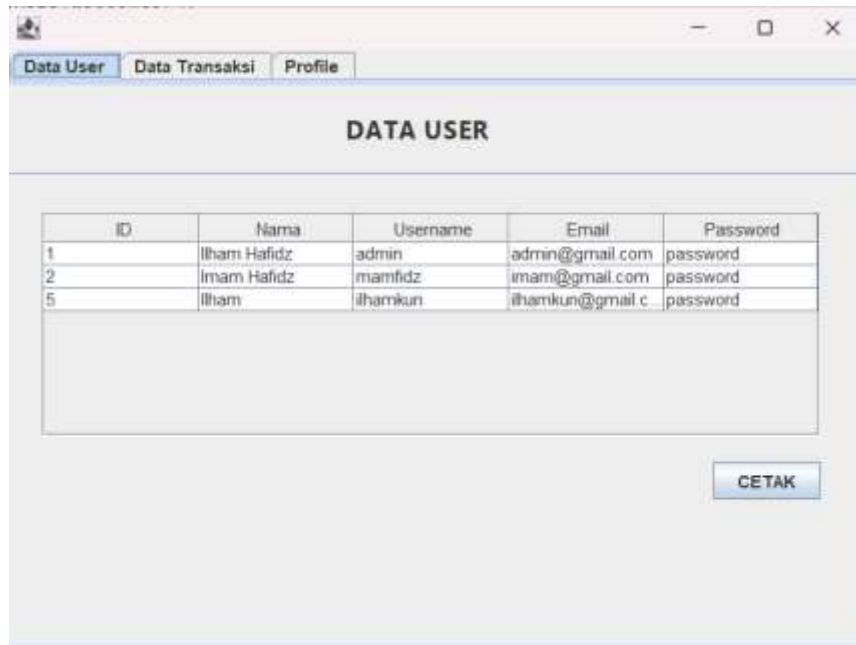
The screenshot shows a web form titled "TOP UP DIAMOND FREE FIRE". At the top is a banner image with the "FREE FIRE" logo. Below the banner are three tabs: "TOP UP" (selected), "DATA TRANSAKSI", and "PROFILE". The form contains the following fields and options:

- An "Email" input field.
- A "USER ID FREE FIRE" input field.
- A "SERVER" dropdown menu with "Asia" selected.
- A grid of radio button options for different diamond amounts and prices:
 - ☐ 5 (Rp 1.000)
 - ☐ 50 (Rp 7.000)
 - ☐ 90 (Rp 11.000)
 - ☐ 100 (Rp 14.000)
 - ☐ 160 (22.000)
 - ☐ 210 (Rp 27.000)
 - ☐ 280 (Rp 30.000)
 - ☐ 1075 (Rp 135.000)
- Two buttons at the bottom: "BATAL" and "BELI".

This screenshot shows the next step in the top-up process, titled "Konfirmasi Email". It features a "Konfirmasi Email" input field and a "Konfirmasi" button. Below these is a large text area with the instruction "Silahkan Konfirmasi Terlebih Dahulu" (Please confirm first).

- CMSAdmin.java

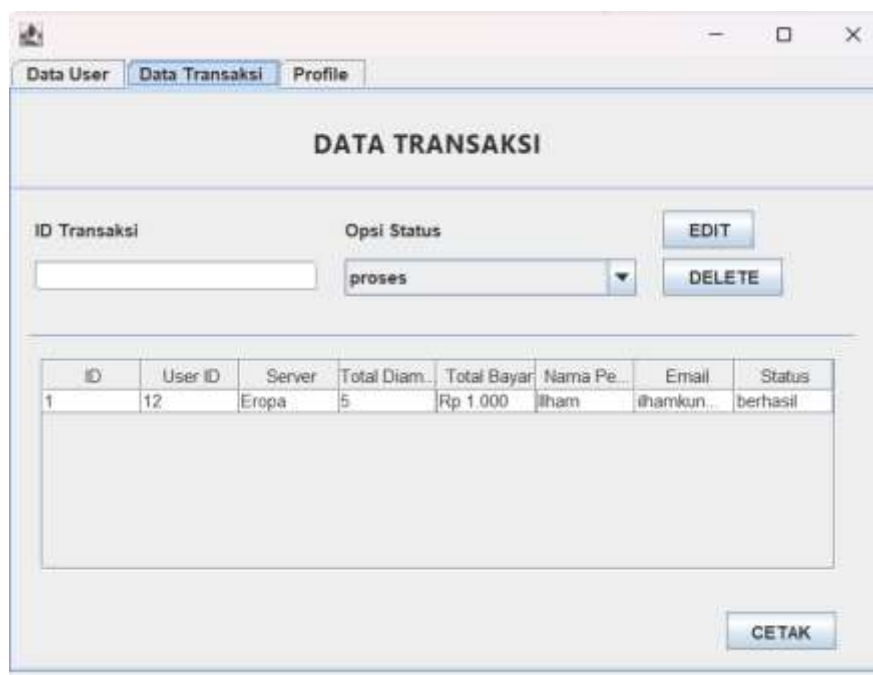
Merupakan panel form yang akan muncul ketika login sebagai admin



DATA USER

ID	Nama	Username	Email	Password
1	Ilham Hafidz	admin	admin@gmail.com	password
2	Imam Hafidz	mamfidz	imam@gmail.com	password
5	Ilham	ilhamkun	ilhamkun@gmail.c	password

CETAK



DATA TRANSAKSI

ID Transaksi:

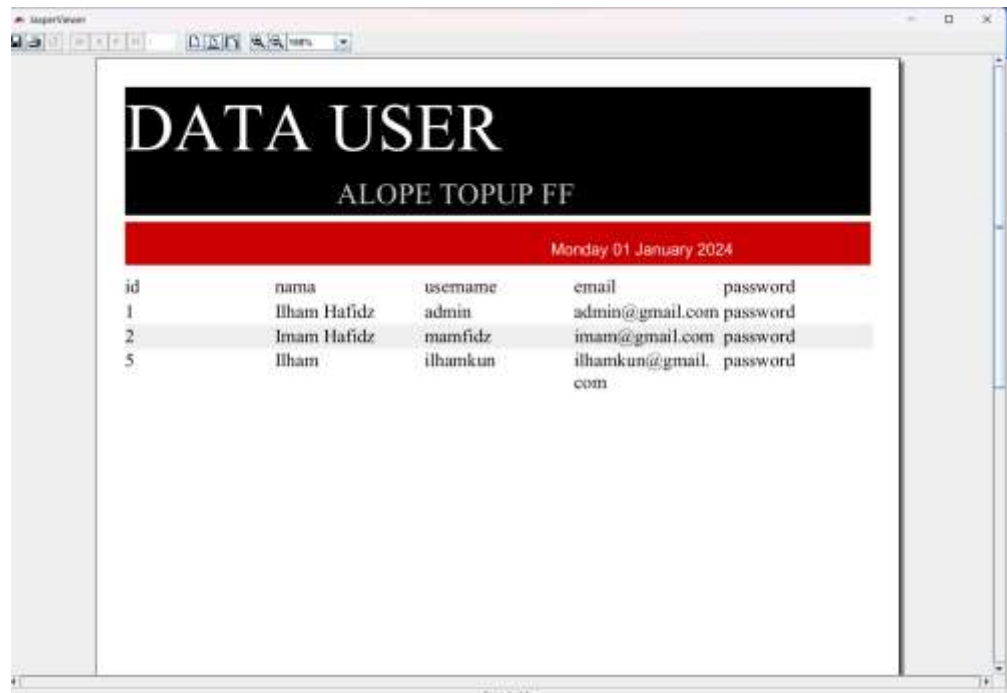
Opsi Status:

EDIT **DELETE**

ID	User ID	Server	Total Diam..	Total Bayar	Nama Pe...	Email	Status
1	12	Eropa	5	Rp 1.000	Ilham	ilhamkun...	berhasil

CETAK

Ketika tombol cetak Data User di klik



id	nama	username	email	password
1	Ilham Hafidz	admin	admin@gmail.com	password
2	Imam Hafidz	mamfidz	imam@gmail.com	password
5	Ilham	ilhamkun	ilhamkun@gmail.com	password

Ketika tombol cetak data transaksi di klik



id	user_id	server	Diamond	Harga	account_id	status
1	12	Eropa	5	1	5	berhasil