

# **LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK**

Disiapkan Untuk Memenuhi Tugas: Membuat *Button Download* dan Fitur  
*Security Question* di Java GUI

Mata Kuliah: Pemrograman Berorientasi Objek (PBO)

Dosen Pengampu: Bayu Adhi Nugroho, Ph. D.



Oleh:  
Nadiya Putri Intan Nur Rahmadhani  
(09020624056)

**KELAS H7B.3**  
**SISTEM INFORMASI**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
**UIN SUNAN AMPEL SURABAYA**  
**2025**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>1</b>
<b>DAFTAR GAMBAR.....</b>	<b>2</b>
<b>BAB I.....</b>	<b>3</b>
1.1. Latar Belakang.....	3
1.2. Tujuan Praktikum.....	3
1.3. Manfaat Praktikum .....	3
<b>BAB II .....</b>	<b>5</b>
2.1. Membuat Project Bernama “PertemuanKeempatbelas”.....	5
2.2. Membuat Package.....	5
2.3. Membuat Entity Unit.....	6
2.4. Membuat JFrame Form “Login” .....	14
2.5. Membuat JFrame Form Bernama “DaftarAkun” .....	18
2.6. Membuat JFrame Form Bernama “UbahPassword” .....	22
2.7. Membuat <i>Code</i> Untuk JFrame Form “UbahPassword” .....	23
2.8. Membuat JFrame Form Bernama “TokoBunga” .....	26
2.9. Membuat JDialog Bernama “InsertDialog” .....	45
2.10. Membuat JDialog Bernama “InsertDialog2” .....	49
2.11. Membuat JDialog Bernama “UpdateDialog” .....	55
2.12. Membuat JDialog Bernama “UpdateDialog2” .....	60
2.13. Membuat JDialog Bernama “DeleteDialog” .....	65
2.14. Membuat JDialog Bernama “DeleteDialog2” .....	70
2.15. Membuat jasper Reports.....	75
2.16. Hasil <i>Running</i> .....	76
<b>BAB III.....</b>	<b>84</b>
3.1. Kesimpulan.....	84
<b>DAFTAR PUSTAKA .....</b>	<b>85</b>

## DAFTAR GAMBAR

Gambar 1 Membuat project .....	5
Gambar 2 Membuat package Login dan TokoBunga .....	6
Gambar 3 Entity Unit Login_1 .....	6
Gambar 4 Entity Unit Tokobunga_1 .....	8
Gambar 5 Entity Unit Transaksitokobunga .....	11
Gambar 6 Persistence.xml.....	14
Gambar 7 Login .....	15
Gambar 8 Code halaman Login .....	17
Gambar 9 Membuat form Daftar Akun .....	18
Gambar 10 Desain halaman Daftar Akun .....	18
Gambar 11 Code pada halaman Daftar Akun.....	21
Gambar 12 Desain halaman Ubah Password .....	22
Gambar 13 Membuat form Ubah Password .....	22
Gambar 14 Code untuk halaman Ubah Password .....	26
Gambar 15 Membuat form TokoBunga .....	26
Gambar 16 JFrame utama .....	27
Gambar 17 Insert 1 .....	45
Gambar 18 Insert 2 .....	49
Gambar 19 Update 1.....	55
Gambar 20 Update 2.....	60
Gambar 21 Delete 1 .....	65
Gambar 22 Delete 2 .....	70
Gambar 23 Jasper reports.....	75
Gambar 24 Hasil running .....	76

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Pengembangan aplikasi berbasis GUI (Graphical User Interface) pada Java merupakan salah satu kemampuan penting dalam pemrograman modern, terutama karena aplikasi dengan antarmuka visual lebih mudah digunakan dan lebih interaktif bagi pengguna. Java Swing sebagai library GUI populer memungkinkan pembuatan tombol, tabel, dialog, serta fitur interaktif seperti ekspor data dan keamanan berbasis input pengguna. Menurut penelitian yang membahas efisiensi antarmuka Java Swing, penggunaan komponen GUI dapat meningkatkan pemahaman pengguna serta mempermudah navigasi dalam aplikasi desktop edukatif (Efendi et al., 2024).

Selain antarmuka yang mudah digunakan, aspek keamanan aplikasi juga menjadi perhatian penting. Salah satu mekanisme keamanan sederhana tetapi efektif adalah penggunaan security question, yaitu pertanyaan verifikasi yang digunakan untuk melindungi akun atau mengonfirmasi identitas pengguna sebelum melakukan tindakan tertentu. Penelitian menunjukkan bahwa mekanisme autentikasi tambahan, termasuk pertanyaan keamanan, dapat meningkatkan ketahanan aplikasi terhadap akses yang tidak sah dan memberikan lapisan perlindungan tambahan bagi pengguna (Al Godzali et al., 2025).

Selain keamanan, fitur download/export data juga menjadi bagian penting dalam aplikasi berbasis Java GUI. Fitur ini memungkinkan pengguna menyimpan data dalam format seperti CSV agar dapat dianalisis atau disimpan sebagai arsip. Studi terkait implementasi ekspor data pada aplikasi desktop menunjukkan bahwa fitur ekspor membantu meningkatkan fungsi sistem dan memberikan fleksibilitas bagi pengguna dalam memanipulasi data di luar aplikasi.

### **1.2. Tujuan Praktikum**

1. Untuk menerapkan pembuatan fitur Download (Export CSV) pada Java GUI menggunakan komponen Swing untuk menyimpan data tabel ke dalam file eksternal.
2. Untuk membuat dan mengimplementasikan Security Question sebagai mekanisme verifikasi identitas untuk meningkatkan keamanan aplikasi Java.
3. Untuk mengintegrasikan fitur GUI interaktif sehingga aplikasi memiliki fungsi download dan keamanan yang berjalan dengan baik, stabil, serta mudah digunakan.

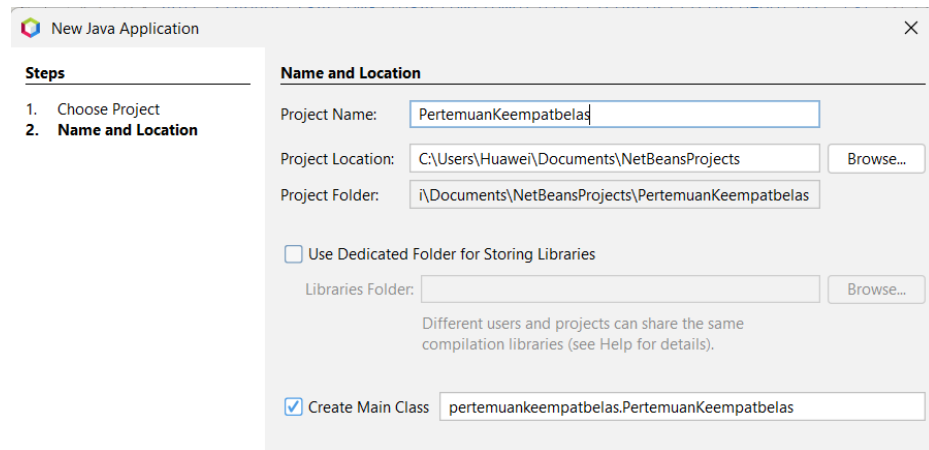
### **1.3. Manfaat Praktikum**

1. Agar mampu menerapkan pembuatan fitur Download (Export CSV) pada Java GUI menggunakan komponen Swing untuk menyimpan data tabel ke dalam file eksternal.
2. Agar mampu membuat dan mengimplementasikan Security Question sebagai mekanisme verifikasi identitas untuk meningkatkan keamanan aplikasi Java.
3. Agar mampu mengintegrasikan fitur GUI interaktif sehingga aplikasi memiliki fungsi download dan keamanan yang berjalan dengan baik, stabil, serta mudah digunakan.

## BAB II

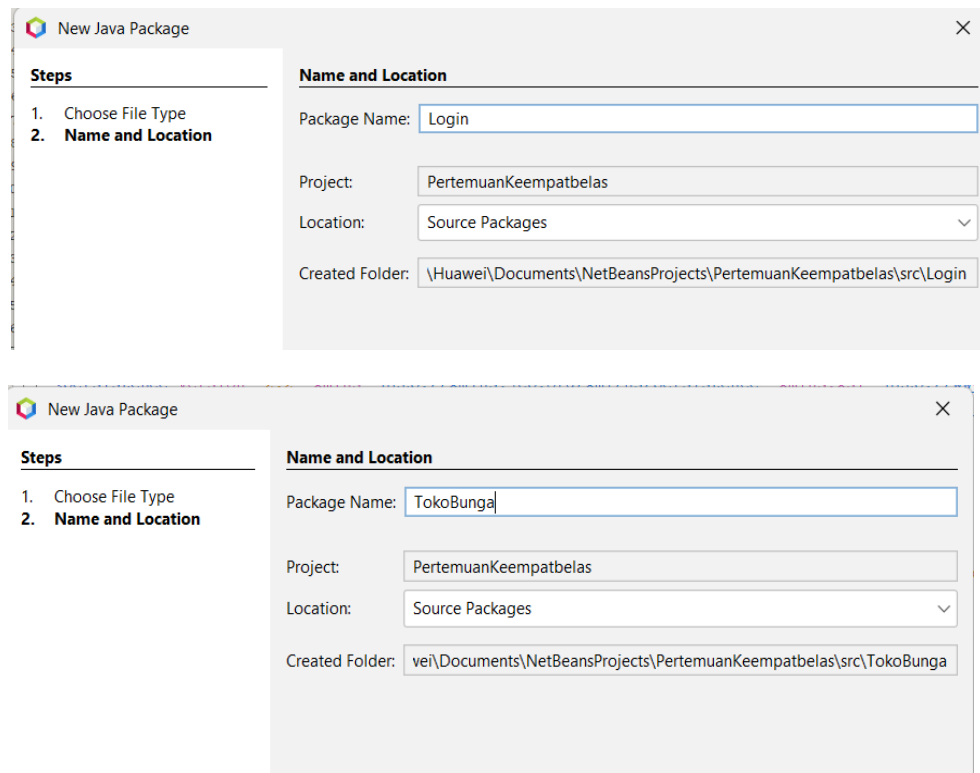
### PEMBAHASAN

#### 2.1. Membuat Project Bernama “PertemuanKeempatbelas”



Gambar 1 Membuat project

#### 2.2. Membuat Package



Gambar 2 Membuat package Login dan TokoBunga

## 2.3. Membuat Entity Unit

### 2.1. Login\_1

```
package Login;

import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
```

Gambar 3 Entity Unit Login\_1

```
@Entity
@Table(name = "login")
@NamedQueries({
    @NamedQuery(name = "Login_1.findAll", query = "SELECT l FROM Login_1 l"),
    @NamedQuery(name = "Login_1.findByUsername", query = "SELECT l FROM Login_1 l WHERE l.username = :username"),
    @NamedQuery(name = "Login_1.findByPassword", query = "SELECT l FROM Login_1 l WHERE l.password = :password"),
    @NamedQuery(name = "Login_1.findBySecurityQuestion", query = "SELECT l FROM Login_1 l WHERE l.securityQuestion = :securityQuestion"),
    @NamedQuery(name = "Login_1.findByAnswer", query = "SELECT l FROM Login_1 l WHERE l.answer = :answer")))

public class Login_1 implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @Basic(optional = false)
    @Column(name = "username")
    private String username;

    @Column(name = "password")
    private String password;

    @Column(name = "security_question")
    private String securityQuestion;

    @Column(name = "answer")
    private String answer;
```

---

```
public Login_1() {  
}  
  
public Login_1(String username) {  
    this.username = username;  
}  
  
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;  
}
```

---

```
public String getSecurityQuestion() {  
    return securityQuestion;  
}  
  
public void setSecurityQuestion(String securityQuestion) {  
    this.securityQuestion = securityQuestion;  
}  
  
public String getAnswer() {  
    return answer;  
}  
  
public void setAnswer(String answer) {  
    this.answer = answer;  
}
```



```

@Override
public int hashCode() {
    int hash = 0;
    hash += (username != null ? username.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Login_1)) {
        return false;
    }
    Login_1 other = (Login_1) object;
    if ((this.username == null && other.username != null) || (this.username != null && !this.username.equals(other.username))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Login.Login_1[ username=" + username + " ]";
}
}

```

## 2.2. Tokobunga\_1

```

package TokoBunga;

import java.io.Serializable;
import java.util.Collection;
import javax.persistence.Basic;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;
import javax.persistence.Table;

```

Gambar 4 Entity Unit Tokobunga\_1

```

@Entity
@Table(name = "tokobunga")
@NamedQueries({
    @NamedQuery(name = "Tokobunga_1.findAll", query = "SELECT t FROM Tokobunga_1 t"),
    @NamedQuery(name = "Tokobunga_1.findByIdbunga", query = "SELECT t FROM Tokobunga_1 t WHERE t.idbunga = :idbunga"),
    @NamedQuery(name = "Tokobunga_1.findByNamebunga", query = "SELECT t FROM Tokobunga_1 t WHERE t.namabunga = :namabunga"),
    @NamedQuery(name = "Tokobunga_1.findByKategori", query = "SELECT t FROM Tokobunga_1 t WHERE t.kategori = :kategori"),
    @NamedQuery(name = "Tokobunga_1.findByDeskripsi", query = "SELECT t FROM Tokobunga_1 t WHERE t.deskripsi = :deskripsi"),
    @NamedQuery(name = "Tokobunga_1.findByHarga", query = "SELECT t FROM Tokobunga_1 t WHERE t.harga = :harga"),
    @NamedQuery(name = "Tokobunga_1.findByStok", query = "SELECT t FROM Tokobunga_1 t WHERE t.stok = :stok"),
    @NamedQuery(name = "Tokobunga_1.findByUkuran", query = "SELECT t FROM Tokobunga_1 t WHERE t.ukuran = :ukuran")
})

```

```

public class Tokobunga_1 implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @Basic(optional = false)
    @Column(name = "idbunga")
    private String idbunga;
    @Column(name = "namabunga")
    private String namabunga;
    @Column(name = "kategoribunga")
    private String kategoribunga;
    @Column(name = "deskripsiproduk")
    private String deskripsiproduk;
    @Column(name = "harga")
    private Integer harga;
    @Column(name = "stok")
    private Integer stok;
    @Column(name = "ukuran")
    private String ukuran;
    @OneToMany(mappedBy = "idbunga", cascade = CascadeType.ALL)
    private Collection<Transaksitokobunga> transaksitokobungaCollection;

}

public Tokobunga_1() {
}

public Tokobunga_1(String idbunga) {
    this.idbunga = idbunga;
}

public String getIdbunga() {
    return idbunga;
}

public void setIdbunga(String idbunga) {
    this.idbunga = idbunga;
}

public String getNamabunga() {
    return namabunga;
}

public void setNamabunga(String namabunga) {
    this.namabunga = namabunga;
}

public String getKategoribunga() {
    return kategoribunga;
}

public void setKategoribunga(String kategoribunga) {
    this.kategoribunga = kategoribunga;
}
}

```

```
public String getDeskripsiproduk () {  
    return deskripsiproduk;  
}  
  
public void setDeskripsiproduk (String deskripsiproduk) {  
    this.deskripsiproduk = deskripsiproduk;  
}  
  
public Integer getHarga () {  
    return harga;  
}  
  
public void setHarga (Integer harga) {  
    this.harga = harga;  
}  
  
public Integer getStok () {  
    return stok;  
}  
  
public void setStok (Integer stok) {  
    this.stok = stok;  
}  
  
public String getUkuran () {  
    return ukuran;  
}  
  
public void setUkuran (String ukuran) {  
    this.ukuran = ukuran;  
}
```

```

    public Collection<Transaksitokobunga> getTransaksitokobungaCollection() {
        return transaksitokobungaCollection;
    }

    public void setTransaksitokobungaCollection(Collection<Transaksitokobunga> transaksitokobungaCollection) {
        this.transaksitokobungaCollection = transaksitokobungaCollection;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (idbunga != null ? idbunga.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof Tokobunga_1)) {
            return false;
        }
        Tokobunga_1 other = (Tokobunga_1) object;
        if ((this.idbunga == null && other.idbunga != null) || (this.idbunga != null && !this.idbunga.equals(other.idbunga))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return idbunga;
    }
}

```

### 2.3. Transaksitokobunga

```

package TokoBunga;

import java.io.Serializable;
import java.util.Date;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

```

Gambar 5 Entity Unit Transaksitokobunga

```

@Entity
@Table(name = "transaksitokobunga")
@NamedQueries({
    @NamedQuery(name = "Transaksitokobunga.findAll", query = "SELECT t FROM Transaksitokobunga t"),
    @NamedQuery(name = "Transaksitokobunga.findByIdtransaksi", query = "SELECT t FROM Transaksitokobunga t WHERE t.idtransaksi = :idtransaksi"),
    @NamedQuery(name = "Transaksitokobunga.findByNamapelanggan", query = "SELECT t FROM Transaksitokobunga t WHERE t.namapelanggan = :namapelanggan"),
    @NamedQuery(name = "Transaksitokobunga.findByTanggal", query = "SELECT t FROM Transaksitokobunga t WHERE t.tanggal = :tanggal"),
    @NamedQuery(name = "Transaksitokobunga.findByJumlah", query = "SELECT t FROM Transaksitokobunga t WHERE t.jumlah = :jumlah"),
    @NamedQuery(name = "Transaksitokobunga.findByTotalharga", query = "SELECT t FROM Transaksitokobunga t WHERE t.totalharga = :totalharga")
})

```

```

public class Transaksitokobunga implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @Basic(optional = false)
    @Column(name = "idtransaksi")
    private String idtransaksi;

    @Column(name = "namapelanggan")
    private String namapelanggan;

    @Column(name = "tanggal")
    @Temporal(TemporalType.DATE)
    private Date tanggal;

    @Column(name = "jumlah")
    private Integer jumlah;

    @Column(name = "totalharga")
    private Integer totalharga;

    @JoinColumn(name = "idbunga", referencedColumnName = "idbunga")
    @ManyToOne

```

```

private Tokobunga_1 idbunga;

public Transaksitokobunga() {
}

public Transaksitokobunga(String idtransaksi) {
    this.idtransaksi = idtransaksi;
}

public String getIdtransaksi() {
    return idtransaksi;
}

public void setIdtransaksi(String idtransaksi) {
    this.idtransaksi = idtransaksi;
}

public String getNamapelanggan() {
    return namapelanggan;
}

public void setNamapelanggan(String namapelanggan) {
    this.namapelanggan = namapelanggan;
}

public Date getTanggal() {
    return tanggal;
}

public void setTanggal(Date tanggal) {
    this.tanggal = tanggal;
}

public Integer getJumlah() {
    return jumlah;
}

public void setJumlah(Integer jumlah) {
    this.jumlah = jumlah;
}

public Integer getTotalharga() {
    return totalharga;
}

public void setTotalharga(Integer totalharga) {
    this.totalharga = totalharga;
}

public Tokobunga_1 getIdbunga() {
    return idbunga;
}

public void setIdbunga(Tokobunga_1 idbunga) {
    this.idbunga = idbunga;
}

```

```

@Override
public int hashCode() {
    int hash = 0;
    hash += (idtransaksi != null ? idtransaksi.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    if (!(object instanceof Transaksitokobunga)) {
        return false;
    }
    Transaksitokobunga other = (Transaksitokobunga) object;
    if ((this.idtransaksi == null && other.idtransaksi != null)
        || (this.idtransaksi != null && !this.idtransaksi.equals(other.idtransaksi))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "TokoBunga.Transaksitokobunga[ idtransaksi=" + idtransaksi + " ]";
}
}

```

Berikut adalah code dalam file.xml nya:

```

<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.2" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_2.xsd">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <class>TokoBunga.Tokobunga_1</class>
    <class>TokoBunga.Transaksitokobunga</class>
    <class>Login.Login_1</class>
    <properties>
        <property name="javax.persistence.jdbc.url" value="jdbc:postgresql://localhost:5432/DB_FBO_P4"/>
        <property name="javax.persistence.jdbc.user" value="postgres"/>
        <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver"/>
        <property name="javax.persistence.jdbc.password" value="1234"/>
    </properties>
</persistence>

```

Gambar 6 Persistence.xml

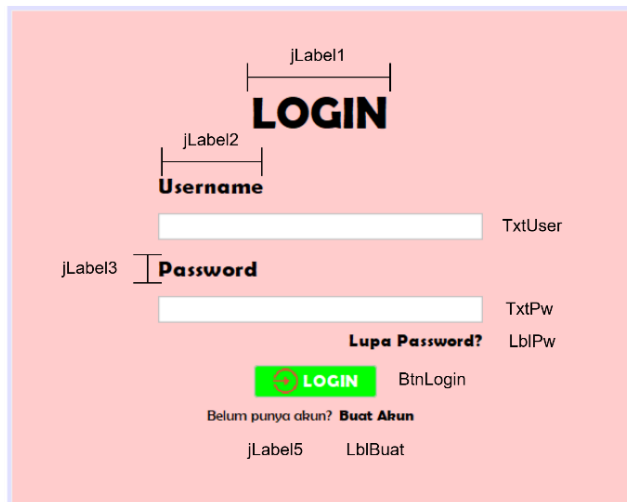
## 2.4. Membuat JFrame Form “Login”

Setelah membuat JFrame Form bernama “Login”, maka desain GUI nya agar sesuai kebutuhan.

The screenshot shows the 'New JFrame Form' dialog box. On the left, under 'Steps', step 2 'Name and Location' is selected. The main area contains the following fields:

- Class Name:** Login
- Project:** PertemuanKeempatbelas
- Location:** Source Packages (dropdown menu)
- Package:** Login (dropdown menu)
- Created File:** \\Documents\\NetBeansProjects\\PertemuanKeempatbelas\\src\\Login\\Login.java
- Superclass:** (empty field) with a 'Browse...' button
- Interfaces:** (empty field) with a 'Browse...' button

Berikut adalah desain GUI dari “Login”. Di mana, setiap nama variabel sudah tertera di gambar.



Gambar 7 Login

```
import javax.swing.JOptionPane;
import javax.persistence.*;

public class Login extends javax.swing.JFrame {

    public Login() {
        initComponents();
    }

    private void TxtUserActionPerformed(java.awt.event.ActionEvent evt)
    {
        if (TxtUser.getText().equals("")) {
            JOptionPane.showMessageDialog(null, "Isi Username
Dahulu!");
        } else {
            TxtPw.requestFocus();
        }
    }
}
```

Dalam *code* ini yang dilakukan pertama adalah mendeklarasikan *library persistence* dan *swing*, setelah itu mendeklarasikan *class* utama dan membuat *method login()* berisi *constructor initComponents()* yang digunakan untuk membangun semua elemen GUI.

Kemudian pada field Username dibuat kondisi ketika user menekan **enter** di *field Username*, maka akan ada 2 kondisi. Yang pertama jika kosong maka akan menampilkan pesan “Isi Username Dahulu!”, dan jika sudah di isi kursor otomatis pindah ke kolom *password*.



```

private void BtnLoginActionPerformed(java.awt.event.ActionEvent evt) {
    if (TxtUser.getText().equals("")
        | TxtPw.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Isi Kolom Terlebih Dahulu!");
    } else {
        EntityManagerFactory emf =
        Persistence.createEntityManagerFactory("PertemuanKetigabelasPU");

        EntityManager em = emf.createEntityManager();

        em.getTransaction().begin();

        String user = TxtUser.getText();

        String pw = TxtPw.getText();

        Login_1 y = em.find(Login_1.class, user);

        if (y == null) {
            JOptionPane.showMessageDialog(null, "Data tidak ditemukan");

            TxtUser.requestFocus();
        } else if (y.getPassword().equals(pw)) {
            JOptionPane.showMessageDialog(null, "Selamat datang");

            this.dispose();
        } else {
            JOptionPane.showMessageDialog(null, "Password salah!");

            TxtPw.setText("");

            TxtPw.requestFocus();
        }

        em.getTransaction().commit();

        em.close();

        emf.close();
    }
}

```

Setelah itu pada **BtnLogin** diberi *code* untuk masuk ke dalam program. Dimana ketika *user* menginput username/ password yang kosong maka akan menampilkan pesan peringatan.

Jika username dan password salah, maka akan menampilkan pesan, “Data tidak ditemukan”. Jika username dan password benar, maka akan menampilkan pesan, “Selamat Datang” dan JFrame Form akan ditutup karena ada function **this.dispose()**. Dan jika password tidak cocok maka akan menampilkan pesan peringatan, “Password salah!”. Setelah itu, transaksi akan diselesaikan dan koneksi akan di tutup.

```

private void LblBuatMouseClicked(java.awt.event.MouseEvent evt) {

    BuatAkun y = new BuatAkun();

    y.setVisible(true);

    this.dispose();

}

private void TxtPwActionPerformed(java.awt.event.ActionEvent evt) {

    BtnLogin.doClick();

}

private void LblPwMouseClicked(java.awt.event.MouseEvent evt) {

    UbahPassword y = new UbahPassword();

    y.setVisible(true);

    this.dispose();

}

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Login().setVisible(true);
        }
    });
}

```

Gambar 8 Code halaman Login

Ketika tulisan “**Buat Akun**” (**LblBuat**) di klik, maka akan membuka JFrame Form “**BuatAkun**” dan menutup JFrame Form “**Login**”.

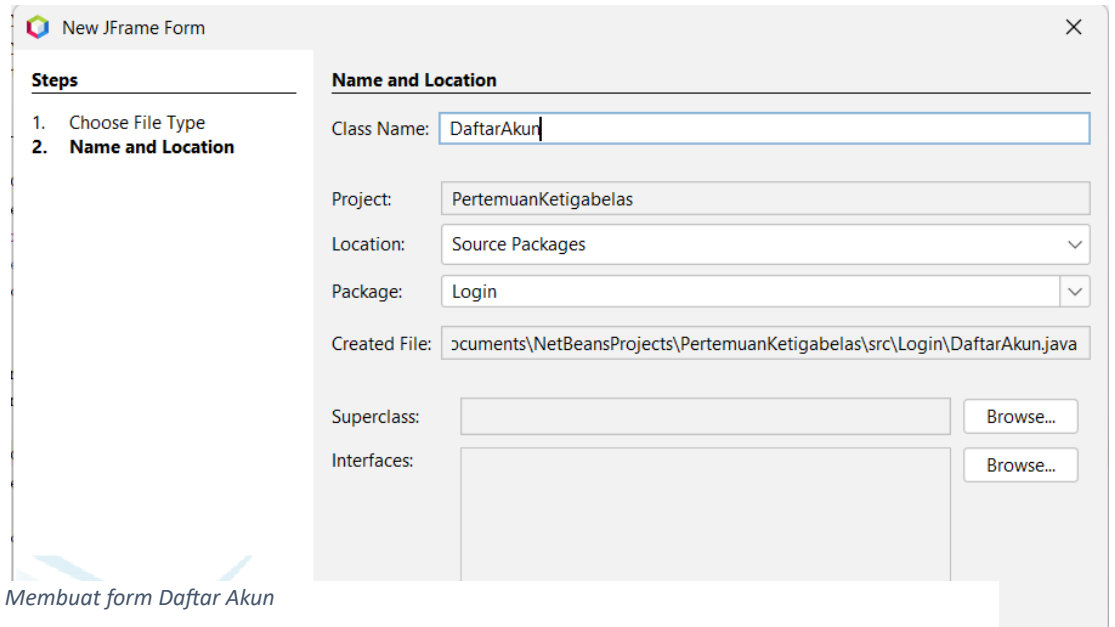
*Code* ini di buat untuk memudahkan *user* menjalankan fungsi **BtnLogin()** dengan menekan “Enter” setelah mengisi password.

Ketika tulisan “**Lupa Password**” (**LblPw**) di klik, maka akan membuka JFrame Form “**UbahPassword**” dan akan menutup JFrame Form “**Login**”.


*Main Method* digunakan untuk menjalankan operasi GUI.

## 2.5. Membuat JFrame Form Bernama “DaftarAkun”

Setelah membuat JFrame Form bernama “**DaftarAkun**”, maka desain GUI nya agar sesuai kebutuhan.



Gambar 9 Membuat form Daftar Akun



Gambar 10 Desain halaman Daftar Akun

Berikut adalah desain GUI dari “DaftarAkun”.

Berikut adalah *code* nya:

```

package Login;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import javax.swing.JOptionPane;

public class DaftarAkun extends javax.swing.JFrame {
    public DaftarAkun () {
        initComponents();
    }

    public void bersih(){
        TxtUser.setText("");
        TxtPw.setText("");
    }

    private void BtnDaftarActionPerformed(java.awt.event.ActionEvent
    evt) {
        if (TxtUser.getText().equals("")
            | TxtPw.getText().equals("")) {
            JOptionPane.showMessageDialog(null, "Isi Semua Kolom
            Terlebih Dahulu!");
        } else {
            String user = TxtUser.getText();
            String pw = TxtPw.getText();

```

Penjelasan lokasi dari JFrame Form, dan *import library* untuk *class* ini (Persistence & JOptionPane).

Mendeklarasikan *class* utama, dan otomatis terdapat *constructor* dimana Netbeans akan membuat komponen GUI.

Lalu membuat *method* **bersih()**. Untuk membersihkan *input form* setelah daftar selesai atau ada *error*.

*Code* untuk *button* **BtnDaftar**.

Jika kolom username atau password kosong, maka akan menampilkan pesan, "Isi Semua Kolom Terlebih Dahulu!".

Kondisi lain jika kolom username dan password terdapat isinya, maka program akan mengambil data yang di *input*.

```

EntityManagerFactory emf =
Persistence.createEntityManagerFactory("PertemuanKetigabelasPU");

EntityManager em = emf.createEntityManager();

em.getTransaction().begin();

Login_1 y = em.find(Login_1.class, user);

if (y != null) {

    JOptionPane.showMessageDialog(null, "Username sudah ada,
gunakan username lain!");

    bersih();

    TxtUser.requestFocus();

} else {

    Login_1 x = new Login_1();

    x.setUsername(user);

    x.setPassword(pw);

    em.persist(x);

    em.getTransaction().commit();

    JOptionPane.showMessageDialog(null, "Sukses dibuat!");

    bersih();

    Login z = new Login();

    z.setVisible(true);

    this.dispose();

}

em.close();

emf.close();

}

}

```

Setelah itu dilakukan koneksi ke *database* via JPA. Di mana, “PertemuanKetigabelasPU” adalah nama *persistence unit* dari file **persistence.xml**. Kemudian, EntityManager merupakan objek untuk operasi *database*. Dan *method* **begin()** adalah tanda untuk memulai transaksi.

Setelah itu dilakukan pengecekan apakah username sudah ada, dimana username adalah **primary key**. Jika username sudah dipakai maka akan menampilkan pesan seperti berikut, “Username sudah ada, gunakan username lain!”. Namun apabila kondisinya username belum digunakan, maka akun baru akan di buat.

Dengan membuat objek entitas baru lalu menyatat username & password, setelah itu data akan disimpan dalam *database* menggunakan *method*, **persist()**. Dan transaksi akan di *commit* (menjalankan perintah INSERT).

Setelah sukses menambahkan akun, maka tampilan akan langsung berpindah ke form **Login**.

Dan koneksi ke *database* akan di tutup

```

private void TxtUserActionPerformed(java.awt.event.ActionEvent evt)
{
    if (TxtUser.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Isi Username
Dahulu!");
    }else{
        TxtPw.requestFocus();
    }
}

private void LblBackMouseClicked(java.awt.event.MouseEvent evt)
{
    Login y = new Login();
    y.setVisible(true);
    this.dispose();
}

private void TxtPwActionPerformed(java.awt.event.ActionEvent
evt) {
    BtnDaftar.doClick();
}

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new DaftarAkun().setVisible(true);
        }
    });
}

```

Kemudian pada bagian *field* username (TxtUser), jika kosong maka akan menampilkan pesan, "Isi Username Dahulu!". Jika user menekan **Enter** di *field* username untuk di validasi dan jika benar maka akan langsung berpindah ke *field* password untuk mengisi data password

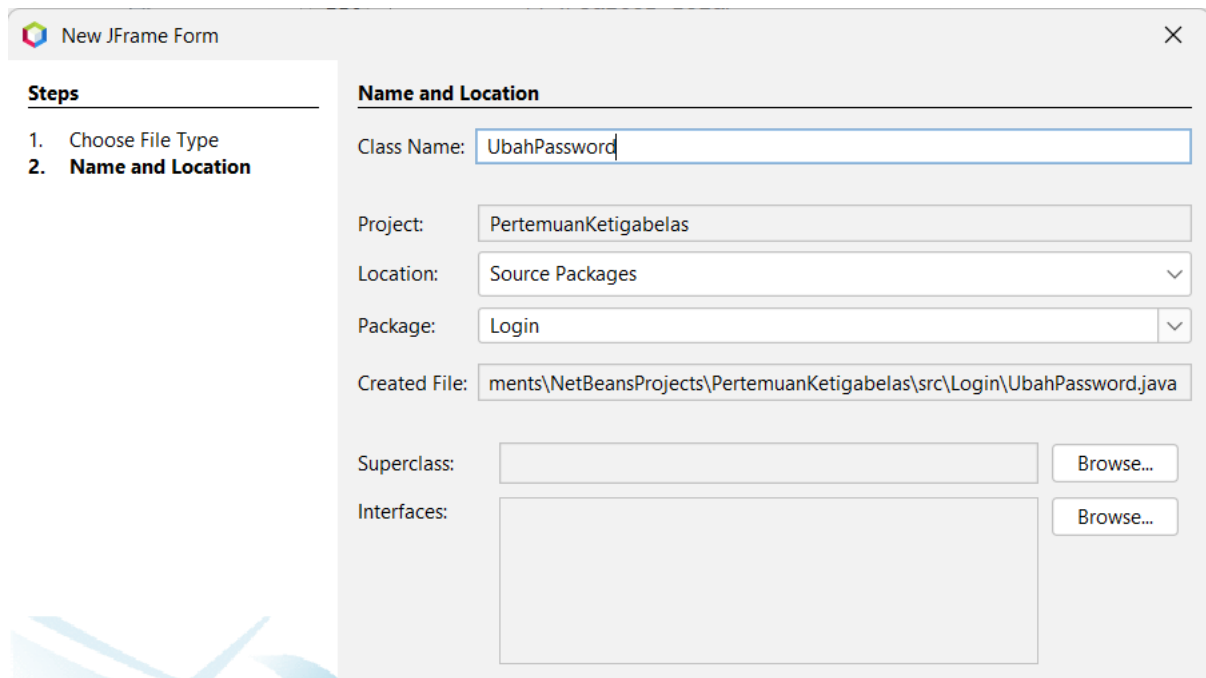
Ketika mengklik ikon back maka akan kembali ke halaman **Login**.

Jika menekan **Enter** di *field* password, maka otomatis akan mengklik *button* daftar.

*Main Method* digunakan untuk menjalankan program GUI.

Gambar 11 Code pada halaman Daftar Akun

## 2.6. Membuat JFrame Form Bernama “UbahPassword”



Gambar 13 Membuat form Ubah Password

The image displays a GUI design for a password change form. It has a light pink background. At the top left is a small icon of a person. The title 'UBAH PASSWORD' is centered in bold black text. Below the title are four labeled input fields: 'Username', 'Security Question' (which includes a dropdown menu with the text 'Pilih Security Question'), 'Answer', and 'Password'. To the right of the 'Security Question' dropdown is a button with a magnifying glass icon and the text 'SEARCH'. At the bottom center is a yellow button with a key icon and the text 'UBAH'.

Gambar 12 Desain halaman Ubah Password

Berikut adalah desain GUI untuk halaman “UbahPassword”

## 2.7. Membuat Code Untuk JFrame Form “UbahPassword”

```
package Login;

import javax.persistence.*;
import javax.swing.*;

public class UbahPassword extends javax.swing.JFrame {
    public UbahPassword() {
        initComponents();
        LblPsw.setVisible(false);
        BtnUbah.setVisible(false);
        TxtPw.setVisible(false);
    }

    private void
    TxtUserActionPerformed(java.awt.event.ActionEvent evt) {
        if (TxtUser.getText().equals("")) {
            JOptionPane.showMessageDialog(null, "Isi
            username dulu!");
        } else {
            BtnCari.doClick();
        }
    }

    private void
    LblBackMouseClicked(java.awt.event.MouseEvent evt) {
        Login y = new Login();
        y.setVisible(true);
        this.dispose();
    }
}
```

Lokasi dari JFrame Form

Mengimport dari *library* Java (*persistence* dan *swing*).

Mendeklarasikan *Main Class*, dan di dalamnya terdapat *constructor* untuk membuat form dan langsung menyembunyikan **LblPsw**, **TxtPw**, **BtnUbah**. Karena sebelum username dicari, maka password tidak boleh muncul.

Kemudian pada *field* username, jika kosong maka akan menampilkan pesan berbunyi, “Isi username dulu!”. Namun jika kondisi lain, kalau *field* terisi maka *button* cari akan dijalankan.

*Code* untuk kembali ke halaman **Login** apabila ikon back di klik.



Pada *button* cari (**BtnCari**), hal yang pertama dilakukan adalah memvalidasi apakah *field*

```
private void BtnCariActionPerformed(java.awt.event.ActionEvent
evt) {
    if (TxtUser.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Isi Terlebih Dahulu");
    } else {
        EntityManagerFactory emf =
        Persistence.createEntityManagerFactory("PertemuanKetigabelasPU"
);
        EntityManager em = emf.createEntityManager();
        em.getTransaction().begin();
        String user = TxtUser.getText();
        Login_1 y = em.find(Login_1.class, user);

        if (y == null) {
            JOptionPane.showMessageDialog(null, "Data tidak
ditemukan");
            TxtUser.requestFocus();
        } else {
            LblPsw.setVisible(true);
            BtnUbah.setVisible(true);
            TxtPw.setVisible(true);
            TxtUser.setEditable(false);
            TxtPw.requestFocus();
        }
        em.getTransaction().commit();
        em.close();
        emf.close();
    }
}
```

username kosong, jika iya maka akan menampilkan pesan untuk mengisi kolom terlebih dahulu. Kondisi lainnya apabila sudah terisi datanya maka akan melakukan koneksi ke *database* via JPA. Dan memulai transaksi, di mana akan mencari data yang dicari berdasarkan *primary key* username.

Hasilnya, jika username tidak ditemukan maka akan menampilkan pesan berbunyi, "Data tidak ditemukan". Dan jika kondisi lainnya data ditemukan maka akan menampilkan *field* password dan *button* ubah. Di mana *field* username tidak akan bisa di edit dan akan berfokus pada *field* password untuk membuat password baru.

Setelah itu data akan disimpan ke *database* dan koneksi ke *database* di tutup.

```

private void TxtPwActionPerformed(java.awt.event.ActionEvent evt)
{
    BtnUbah.doClick();
}

private void BtnUbahActionPerformed(java.awt.event.ActionEvent
evt) {
    if (TxtPw.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Cari Terlebih Dahulu");
    } else {
        EntityManagerFactory emf =
Persistence.createEntityManagerFactory("PertemuanKetigabelasPU");
        EntityManager em = emf.createEntityManager();
        em.getTransaction().begin();

        String user, pw;
        user = TxtUser.getText();
        pw = TxtPw.getText();
        Login_1 x = em.find(Login_1.class, user);
        x.setUsername(user);
        x.setPassword(pw);
        em.persist(x);
        em.getTransaction().commit();
        JOptionPane.showMessageDialog(null, "Berhasil diubah");
        em.close();
        emf.close();
        Login y = new Login();
        y.setVisible(true);
        this.dispose();
    }
}

```

Pada *Field* password, jika menekan tombol **Enter** maka akan langsung menjalankan *button* ubah.

Dan pada *button* ubah, hal yang pertama dilakukan adalah memvalidasi password kosong, jika kosong maka akan menampilkan pesan “Cari Terlebih Dahulu”. Barus setelah tidak kosong maka akan menjalankan koneksi ke *database* via JPA. Dan akan mengambil objek **Login\_1** berdasarkan username. Lalu password akan di update.

Setelah password berhasil di ubah, maka koneksi ke *database* akan di tutup

```

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)
    //</editor-fold>
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new UbahPassword().setVisible(true);
        }
    });
}

```

*Main Method* digunakan untuk menjalankan program GUI.

Gambar 14 Code untuk halaman Ubah Password

## 2.8. Membuat JFrame Form Bernama “TokoBunga”

Gambar 15 Membuat form TokoBunga

Berikut adalah desain dari JFrame Form utama:

**DATA PESANAN TOKO BUNGA**

Data Buket | Data Pesanan

ID Bunga	Nama Bunga	Kategori Bunga	Deskripsi Produk	Harga	Stok Bunga	Ukuran Buket
----------	------------	----------------	------------------	-------	------------	--------------

Insert Delete Upload  
Update Print Download

**DATA PESANAN TOKO BUNGA**

Data Buket | Data Pesanan

ID Transaksi	ID Bunga	Nama Pelanggan	Tanggal Transaksi	Jumlah Pesanan	Total Harga
--------------	----------	----------------	-------------------	----------------	-------------

Insert Delete Upload  
Update Print Download

Gambar 16 JFrame utama

Berikut adalah *code* yang ada di dalamnya:

```
package TokoBunga;
```

```
import java.io.*;
```

```
import java.sql.*;
```

```
import java.text.ParseException;
```

```
import java.text.SimpleDateFormat;
```

```
import java.util.Date;
```

```
import java.util.List;
```

```
import java.util.Locale;
```

```
import java.util.logging.*;
```

```
import javax.swing.*;
```

```
import javax.swing.filechooser.FileSystemView;
```

```
import javax.swing.table.*;
```

```
import net.sf.jasperreports.engine.*;
```

```
import net.sf.jasperreports.view.*;
```

```
import net.sf.jasperreports.engine.util.JRLoader;
```

```
import javax.persistence.*;
```

Import dari library java

```

public class TokoBunga extends javax.swing.JFrame {

    private EntityManagerFactory emf;

    private EntityManager em;

    String idBungaLama, namaBungaLama, kategoriBungaLama,
    deskripsiLama, hargaLama, stokLama, ukuranLama;

    public TokoBunga() {
        initComponents();
        connect();
        showTable1();
        showTable2();
    }

    public void connect() {
        try {
            emf =
Persistence.createEntityManagerFactory("PertemuanKesebelasPU");

            em = emf.createEntityManager();
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "Koneksi gagal: " +
e.getMessage());
        }
    }

    public void showTable1() {
        em.clear();

        EntityManager em =
Persistence.createEntityManagerFactory("PertemuanKesebelasPU").creat
eEntityManager();

        Query q = em.createNamedQuery("Tokobunga_1.findAll");

        List<Tokobunga_1> hasil = q.getResultList();

        DefaultTableModel model = new DefaultTableModel(
            new Object[]{"ID Bunga", "Nama Bunga", "Kategori", "Deskripsi",
"Harga", "Stok", "Ukuran"}, 0
        );
    }

```

Ini adalah **class GUI utama** yang mewarisi **JFrame**. Untuk mengelola koneksi database menggunakan **JPA**

Digunakan saat **edit/update data**, agar data sebelumnya bisa dibandingkan atau disimpan sementara.

Saat form dibuka:

1. `initComponents()` → generate UI dari NetBeans
2. `connect()` → buka koneksi database JPA
3. `showTable1()` → tampilkan data bunga
4. `showTable2()` → tampilkan data transaksi

Mengambil konfigurasi dari `persistence.xml` dan membuat koneksi.

Dipakai untuk *load data tabel bunga*.

#### Alur Kerja:

1. Clear entity manager
2. Query data bunga dari NamedQuery  
`Tokobunga_1.findAll`
3. Masukkan data ke tabel 1 (Data Buket)
4. Update tampilan tabel

```

DefaultTableCellRenderer centerRenderer = new DefaultTableCellRenderer();

centerRenderer.setHorizontalAlignment(JLabel.CENTER);

for (Tokobunga_1 t : hasil) {

    model.addRow(new Object[]{

        t.getIdbunga(),

        t.getNamabunga(),

        t.getKategoribunga(),

        t.getDeskripsiproduk(),

        t.getHarga(),

        t.getStok(),

        t.getUkuran()

    });

}

jTable1.getColumnModel().getColumn(1).setCellRenderer(centerRenderer);

jTable1.setModel(model);

model.fireTableDataChanged();

jTable1.revalidate();

jTable1.repaint();

}

public void showTable2() {

    try {

        em.clear();

        EntityManager em = Persistence.createEntityManagerFactory("PertemuanKesebelasPU").createEntityManager();

        Query q = em.createNamedQuery("Transaksitokobunga.findAll");

        List<Transaksitokobunga> hasil = q.getResultList();

        tbmk = new DefaultTableModel(

            new String[]{"ID Transaksi", "ID Bunga", "Nama Pelanggan", "Tanggal Transaksi", "Jumlah Pesanan", "Total Harga"}, 0

        );

        DefaultTableCellRenderer centerRenderer = new DefaultTableCellRenderer();

        centerRenderer.setHorizontalAlignment(JLabel.CENTER);

        SimpleDateFormat sdf = new SimpleDateFormat("dd-MMM-yyyy");

```

Begitupun yang method showTable2(), dipakai untuk *load data tabel transaksi*.

#### Alur Kerja:

1. Clear entity manager
2. Query data bunga dari NamedQuery Tokobunga\_1.findAll
3. Masukkan data ke tabel 1 (Data Buket)
4. Update tampilan tabel

2. Baca line per line
3. Pisahkan per ;
4. Validasi jumlah kolom
5. Parsing ID, nama, tanggal, jumlah
6. Cari objek bunga berdasarkan ID
7. Simpan transaksi baru ke databa

```
private void BtnUpload2ActionPerformed(java.awt.event.ActionEvent evt) {  
    JFileChooser jfc = new  
JFileChooser(FileSystemView.getFileSystemView().getHomeDirectory());  
    int returnValue = jfc.showOpenDialog(null);  
    if (returnValue == JFileChooser.APPROVE_OPTION) {  
        File filePilihan = jfc.getSelectedFile();  
        System.out.println("File yang dipilih: " + filePilihan.getAbsolutePath());  
    }  
}
```



```

EntityManagerFactory emf = Persistence.createEntityManagerFactory("PertemuanKesebelasPU");

EntityManager em = emf.createEntityManager();

try (BufferedReader br = new BufferedReader(new FileReader(filePilihan))) {

    String baris;

    String pemisah = ",";

    em.getTransaction().begin();

    while ((baris = br.readLine()) != null) {

        if (baris.trim().isEmpty()) {

            continue;

        }

        String[] data = baris.split(pemisah);

        if (data.length < 6) {

            System.out.println("Baris tidak lengkap, dilewati: " + baris);

            continue;

        }

        String idtransaksi = data[0].trim();

        String idbunga = data[1].trim();

        String namapelanggan = data[2].trim();

        String tanggalStr = data[3].trim();

        int jumlah = Integer.parseInt(data[4].trim());

        int totalharga = Integer.parseInt(data[5].trim());

        java.sql.Date tanggal;

        try {

            tanggal = java.sql.Date.valueOf(tanggalStr);

        } catch (Exception e) {

            System.out.println("Format tanggal salah pada baris: " + baris);

            continue;

        }
    }
}

```

```

Tokobunga_1 bunga = em.find(Tokobunga_1.class, idbunga);

    if (bunga == null) {

        System.out.println("Bunga ID " + idbunga + " tidak ditemukan, dilewati.");

        continue;

    }

    Transaksitokobunga t = new Transaksitokobunga();

    t.setIdtransaksi(idtransaksi);

    t.setIdbunga(bunga);

    t.setNamapelanggan(namapelanggan);

    t.setTanggal(tanggal);

    t.setJumlah(jumlah);

    t.setTotalharga(totalharga);

    em.persist(t);

    System.out.println("Input: " + idtransaksi + " - " + namapelanggan);

}

em.getTransaction().commit();

JOptionPane.showMessageDialog(null, "Semua data transaksi berhasil diinput!");

showTable2();

} catch (Exception ex) {

    if (em.getTransaction().isActive()) {

        em.getTransaction().rollback();

    }

    Logger.getLogger(TokoBunga.class.getName()).log(Level.SEVERE, null, ex);

    JOptionPane.showMessageDialog(null, "Terjadi kesalahan: " + ex.getMessage());

} finally {

    em.close();

    emf.close();

}

}

}

```

### Membuat method untuk fitur JasperReport pada data transaksi

```
private void BtnCetak2ActionPerformed(java.awt.event.ActionEvent evt) {  
    Connection conn = null;  
    Statement stmt;  
    PreparedStatement pstmt = null;  
  
    String driver = "org.postgresql.Driver";  
    String koneksi = "jdbc:postgresql://localhost:5432/DB_PBO_P4";  
    String user = "postgres";  
    String password = "1234";  
  
    JasperReport reports;  
    String path = ".\\src\\TokoBunga\\ReportDataTransaksi.jasper";  
    try {  
        try {  
            Class.forName(driver);  
            conn = DriverManager.getConnection(koneksi, user, password);  
        } catch (SQLException ex) {  
            Logger.getLogger(TokoBunga.class.getName()).log(Level.SEVERE, null, ex);  
        } catch (ClassNotFoundException ex) {  
            System.getLogger(TokoBunga.class.getName()).log(System.Logger.Level.ERROR,  
(String) null, ex);  
        }  
        reports = (JasperReport) JRLoader.loadObjectFromFile(path);  
        JasperPrint jprint = JasperFillManager.fillReport(path, null, conn);  
        JasperViewer jviewer = new JasperViewer(jprint, false);  
        jviewer.setDefaultCloseOperation(DISPOSE_ON_CLOSE);  
        jviewer.setVisible(true);  
    } catch (JRException ex) {  
        Logger.getLogger(TokoBunga.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

```

private void BtnDelete2ActionPerformed(java.awt.event.ActionEvent evt) {

    int row = jTable2.getSelectedRow();

    if (row == -1) {

        JOptionPane.showMessageDialog(this, "Pilih data dulu dari tabel!");

        return;

    }

    String idTransaksi = jTable2.getValueAt(row, 0).toString();

    String idBunga = jTable2.getValueAt(row, 1).toString();

    String namaPelanggan = jTable2.getValueAt(row, 2).toString();

    String tanggalStr = jTable2.getValueAt(row, 3).toString();

    String jumlahPesananStr = jTable2.getValueAt(row, 4).toString();

    String totalHargaStr = jTable2.getValueAt(row, 5).toString();

    java.util.Date tanggal = null;

    try {

        SimpleDateFormat sdf = new SimpleDateFormat("dd-MMM-yyyy",
Locale.ENGLISH);

        tanggal = sdf.parse(tanggalStr);

    } catch (Exception e) {

        e.printStackTrace();

    }

    int jumlahPesanan = 0;

    int totalHarga = 0;

    try {

        jumlahPesanan = Integer.parseInt(jumlahPesananStr);

        totalHarga = Integer.parseInt(totalHargaStr);

    } catch (NumberFormatException e) {

        JOptionPane.showMessageDialog(this, "Format angka tidak valid!");

        return;

    }
}

```

```
DeleteDialog2 dialog = new DeleteDialog2(new java.awt.Frame(), true,
idTransaksi, idBunga, namaPelanggan, tanggal, jumlahPesanan, totalHarga);
```

```
    dialog.setLocationRelativeTo(this);
```

```
    dialog.setVisible(true);
```

```
    showTable1();
```

```
    showTable2();
```

```
}
```

```
private void BtnInsert2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    InsertDialog2 dialog = new InsertDialog2(new java.awt.Frame(), true);
```

```
    dialog.setLocationRelativeTo(this);
```

```
    dialog.setVisible(true);
```

```
    showTable2();
```

```
}
```

```
private void BtnUpdate2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    int row = jTable2.getSelectedRow();
```

```
    if (row == -1) {
```

```
        JOptionPane.showMessageDialog(this, "Pilih data dulu dari tabel!");
```

```
        return;
```

```
    }
```

```
    SimpleDateFormat sdf = new SimpleDateFormat("dd-MMM-yyyy",
Locale.ENGLISH);
```

```
    Date tanggal = null;
```

```
    try {
```

```
        String tanggalStr = jTable2.getValueAt(row, 3) != null ?
jTable2.getValueAt(row, 3).toString() : "";
```

```
        if (!tanggalStr.isEmpty()) {
```

```
            tanggal = sdf.parse(tanggalStr);
```

```
        }
```

```
    } catch (ParseException e) {
```

```
        e.printStackTrace();
```

cari transaksi ->  
remove -> commit  
  
refresh table

```

        JOptionPane.showMessageDialog(this, "Format tanggal tidak valid: " +
e.getMessage());

        return;
    }

String idTransaksiLama = jTable2.getValueAt(row, 0).toString();

String idBunga = jTable2.getValueAt(row, 1).toString();

String namaPelanggan = jTable2.getValueAt(row, 2).toString();


int jumlahPesanan;

int totalHarga;

try {

    jumlahPesanan = Integer.parseInt(jTable2.getValueAt(row, 4).toString());

    totalHarga = Integer.parseInt(jTable2.getValueAt(row, 5).toString());
} catch (NumberFormatException nfe) {

    JOptionPane.showMessageDialog(this, "Nilai jumlah atau total harga tidak valid.");

    return;
}

java.awt.Frame parentFrame = null;

java.awt.Window w = SwingUtilities.getWindowAncestor(this);

if (w instanceof java.awt.Frame) {

    parentFrame = (java.awt.Frame) w;
}

UpdateDialog2 dialog = new UpdateDialog2(parentFrame != null ? parentFrame : new java.awt.Frame(),

    true, idTransaksiLama, idBunga, namaPelanggan, tanggal, jumlahPesanan, totalHarga);

dialog.setLocationRelativeTo(this);

dialog.setVisible(true);

showTable2();
}

private void BtnCetakActionPerformed(java.awt.event.ActionEvent evt) {

    Connection conn = null;

    Statement stmt;

    PreparedStatement pstmt = null;

```

1. Cari transaksi berdasarkan ID.
2. Ubah field sesuai input.
3. Commit.

```

String driver = "org.postgresql.Driver";

String koneksi = "jdbc:postgresql://localhost:5432/DB_PBO_P4";

String user = "postgres";

String password = "1234";

JasperReport reports;

String path = ".\\src\\TokoBunga\\ReportDataBunga.jasper";

try {
    try {
        Class.forName(driver);

        conn = DriverManager.getConnection(koneksi, user, password);

    } catch (SQLException ex) {

        Logger.getLogger(TokoBunga.class.getName()).log(Level.SEVERE, null, ex);

    } catch (ClassNotFoundException ex) {

    }

    System.getLogger(TokoBunga.class.getName()).log(System.Logger.Level.ERROR,
    (String) null, ex);

    }

    reports = (JasperReport) JRLoader.loadObjectFromFile(path);

    JasperPrint jprint = JasperFillManager.fillReport(path, null, conn);

    JasperViewer jviewer = new JasperViewer(jprint, false);

    jviewer.setDefaultCloseOperation(DISPOSE_ON_CLOSE);

    jviewer.setVisible(true);

    } catch (JRException ex) {

        Logger.getLogger(TokoBunga.class.getName()).log(Level.SEVERE, null, ex);

    }

}

private void BtnUpdateActionPerformed(java.awt.event.ActionEvent evt) {

    int row = jTable1.getSelectedRow();

    if (row == -1) {

        JOptionPane.showMessageDialog(this, "Pilih data dulu dari tabel!");

        return;

    }

}

```

Membuat method untuk fitur JasperReport pada data tokobunga

Mencari data lama berdasarkan ID bunga. Mengambil data baru dari text field. Memperbarui data di database. Commit perubahan

```

idBungaLama = jTable1.getValueAt(row, 0).toString();

    namaBungaLama = jTable1.getValueAt(row, 1).toString();

    kategoriBungaLama = jTable1.getValueAt(row, 2).toString();

    deskripsiLama = jTable1.getValueAt(row, 3).toString();

    hargaLama = jTable1.getValueAt(row, 4).toString();

    stokLama = jTable1.getValueAt(row, 5).toString();

    ukuranLama = jTable1.getValueAt(row, 6).toString();

    UpdateDialog dialog = new UpdateDialog(this, true, idBungaLama, namaBungaLama,
kategoriBungaLama,

        deskripsiLama, hargaLama, stokLama, ukuranLama);

    dialog.setLocationRelativeTo(this);

    dialog.setVisible(true);

    showTable1();
}

private void BtnInsertActionPerformed(java.awt.event.ActionEvent evt) {

    InsertDialog dialog = new InsertDialog(new java.awt.Frame(), true);

    dialog.setLocationRelativeTo(this);

    dialog.setVisible(true);

    showTable1();
}

private void BtnDeleteActionPerformed(java.awt.event.ActionEvent evt) {

    int row = jTable1.getSelectedRow();

    if (row == -1) {

        JOptionPane.showMessageDialog(this, "Pilih data dulu dari tabel!");

        return;

    }

String idBungaLama = jTable1.getValueAt(row, 0).toString();

    String namaBungaLama = jTable1.getValueAt(row, 1).toString();

    String kategoriBungaLama = jTable1.getValueAt(row, 2).toString();

    String deskripsiLama = jTable1.getValueAt(row, 3).toString();

    String hargaLama = jTable1.getValueAt(row, 4).toString();

```

ambil isi text field, buat entity Tokobunga\_1, set semua field (id, nama, deskripsi, harga, stok, ukuran), em.persist(entity), em.getTransaction().commit(), refresh table

Ambil ID bunga dari text field. Cari bunga dengan: em.find(Tokobunga\_1.class, id), Jika ketemu → dihapus, Jika tidak → pesan error.



```

String stokLama = jTable1.getValueAt(row, 5).toString();

String ukuranLama = jTable1.getValueAt(row, 6).toString();

DeleteDialog dialog = new DeleteDialog(new java.awt.Frame(), true, idBungaLama,
namaBungaLama, kategoriBungaLama, deskripsiLama, hargaLama, stokLama, ukuranLama);

dialog.setLocationRelativeTo(this);

dialog.setVisible(true);

showTable1();

showTable2();

}

private void BtnUploadActionPerformed(java.awt.event.ActionEvent evt) {

    JFileChooser jfc = new
JFileChooser(FileSystemView.getFileSystemView().getHomeDirectory());

    int returnValue = jfc.showOpenDialog(null);

    if (returnValue == JFileChooser.APPROVE_OPTION) {

        File filePilihan = jfc.getSelectedFile();

        System.out.println("yang dipilih : " + filePilihan.getAbsolutePath());

EntityManagerFactory emf =
Persistence.createEntityManagerFactory("PertemuanKesebelasPU");

        EntityManager em = emf.createEntityManager();

        try (BufferedReader br = new BufferedReader(new FileReader(filePilihan))) {

            String baris;

            String pemisah = ",";

            while ((baris = br.readLine()) != null) {

                if (baris.trim().isEmpty()) {

                    continue;

                }

String[] data = baris.split(pemisah);

                if (data.length < 7) {

                    System.out.println("Baris tidak lengkap, dilewati: " + baris);

                    continue;

                }

```

Pilih file CSV. Baca isi file baris per baris. Pisahkan pakai split(","). Validasi jumlah kolom. Buat entity Tokobunga\_1. Insert ke database. Refresh tabel

```

String idbunga = data[0].trim();

String namabunga = data[1].trim();

String kategoribunga = data[2].trim();

String deskripsiproduk = data[3].trim();

int harga = Integer.parseInt(data[4].trim());

int stok = Integer.parseInt(data[5].trim());

String ukuran = data[6].trim();


if (!idbunga.isEmpty() && !namabunga.isEmpty() && !kategoribunga.isEmpty()
    && !deskripsiproduk.isEmpty() && !ukuran.isEmpty()) {

    em.getTransaction().begin();


    Tokobunga_1 tb = new Tokobunga_1();

    tb.setIdbunga(idbunga);

    tb.setNamabunga(namabunga);

    tb.setKategoribunga(kategoribunga);

    tb.setDeskripsiproduk(deskripsiproduk);

    tb.setHarga(harga);

    tb.setStok(stok);

    tb.setUkuran(ukuran);


    em.persist(tb);

    em.getTransaction().commit();


    System.out.println("Berhasil input: " + idbunga + " - " + namabunga);

} else {

    System.out.println("Data tidak valid (ada kolom kosong): " + baris);

}

}

JOptionPane.showMessageDialog(null, "Semua data dari file berhasil diinput!");

showTable1();

```

```

    } catch (Exception ex) {

        if (em.getTransaction().isActive()) {

            em.getTransaction().rollback();

        }

        Logger.getLogger(TokoBunga.class.getName()).log(Level.SEVERE, null, ex);

        JOptionPane.showMessageDialog(null, "Terjadi kesalahan: " + ex.getMessage());

    } finally {

        if (em.isOpen()) {

            em.close();

        }

        if (emf.isOpen()) {

            emf.close();

        }

    }

}

}

}

}

public void exportTableToCSV(JTable table, String filename) {

    JFileChooser fileChooser = new JFileChooser();

    fileChooser.setDialogTitle("Simpan sebagai CSV");

    fileChooser.setSelectedFile(new File(filename));

    int userSelection = fileChooser.showSaveDialog(this);

    if (userSelection == JFileChooser.APPROVE_OPTION) {

        File fileToSave = fileChooser.getSelectedFile();

        if (!fileToSave.getAbsolutePath().toLowerCase().endsWith(".csv")) {

            fileToSave = new File(fileToSave.getAbsolutePath() + ".csv");

        }

        try (FileWriter writer = new FileWriter(fileToSave)) {

            DefaultTableModel model = (DefaultTableModel) table.getModel();

            int columnCount = model.getColumnCount();

```

Query semua data bunga. Buat file baru. Tulis isi data ke file. Beri pesan " Data berhasil diexport ke: " + fileToSave.getAbsolutePath()

```

        for (int i = 0; i < columnCount; i++) {
            writer.write(model.getColumnName(i));
            if (i < columnCount - 1) {
                writer.write(",");
            }
        }
        writer.write("\n");

        for (int i = 0; i < model.getRowCount(); i++) {
            for (int j = 0; j < columnCount; j++) {
                Object value = model.getValueAt(i, j);
                writer.write(value != null ? value.toString() : "");
                if (j < columnCount - 1) {
                    writer.write(",");
                }
            }
            writer.write("\n");
        }

        JOptionPane.showMessageDialog(this, "Data berhasil diexport ke: " + fileToSave.getAbsolutePath());
    } catch (IOException e) {
        JOptionPane.showMessageDialog(this, "Error saat mengexport data: " + e.getMessage(),
            "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void BtnDownload2ActionPerformed(java.awt.event.ActionEvent evt) {
    exportTableToCSV(jTable2, "Data Transaksi.csv");
}

private void BtnDownloadActionPerformed(java.awt.event.ActionEvent evt) {
    exportTableToCSV(jTable1, "Data Bunga.csv");
}

```

```

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)
    //</editor-fold>
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(TokoBunga.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(TokoBunga.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(TokoBunga.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(TokoBunga.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new TokoBunga().setVisible(true);
        }
    });
}
DefaultTableModel tbmk;

```

Main method

```

private javax.swing.JButton BtnCetak;

private javax.swing.JButton BtnCetak2;

private javax.swing.JButton BtnDelete;

private javax.swing.JButton BtnDelete2;

private javax.swing.JButton BtnDownload;

private javax.swing.JButton BtnDownload2;

private javax.swing.JButton BtnExit;

private javax.swing.JButton BtnInsert;

private javax.swing.JButton BtnInsert2;

private javax.swing.JButton BtnUpdate;

private javax.swing.JButton BtnUpdate2;

private javax.swing.JButton BtnUpload;

private javax.swing.JButton BtnUpload2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3; private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2; private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3; private javax.swing.JTabbedPane jTabbedPane2;

private javax.swing.JTable jTable1;

private javax.swing.JTable jTable2;

private javax.swing.JTable jTable3;
}

```

## 2.9. Membuat JDialog Bernama “InsertDialog”



The image shows a Java Swing dialog box titled "InsertDialog" with a pink background. It contains seven text input fields, each with a label to its left: "ID Bunga", "Nama Bunga", "Kategori Bunga", "Deskripsi Produk", "Harga", "Stok Bunga", and "Ukuran Buket". An "Insert" button is positioned at the bottom right of the dialog.

Gambar 17 Insert 1

Berikut adalah *code* nya:

```
package TokoBunga;

import TokoBunga.Tokobunga_1;

import java.io.*;

import java.util.logging.*;

import javax.swing.*;

import javax.swing.table.*;

import javax.persistence.*;

public class InsertDialog extends java.awt.Dialog {

    private EntityManagerFactory emf;

    private EntityManager em;

    public void connect() {

        try {

            emf = Persistence.createEntityManagerFactory("PertemuanKesebelasPU");

            em = emf.createEntityManager();

        } catch (Exception e) {

            JOptionPane.showMessageDialog(this, "Koneksi gagal: " + e.getMessage());

        }

    }

}
```

```

public InsertDialog(java.awt.Frame parent, boolean modal) {

    super(parent, modal);

    initComponents();

    connect();

}

```

[Generate Code]

```

private void closeDialog(java.awt.event.WindowEvent evt) {

    setVisible(false);

    dispose();

}

private void BtnInsertActionPerformed(java.awt.event.ActionEvent evt) {

    if (TxtIdBunga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtNamaBunga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtKategoriBunga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtDeskripsiProduk.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtHarga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtStokBunga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtUkuranBuket.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else {

try {

        EntityManager em = emf.createEntityManager();

        em.getTransaction().begin();

        Tokobunga_1 tb = new Tokobunga_1();

```

```

tb.setIdbunga(TxtIdBunga.getText());

    tb.setNamabunga(TxtNamaBunga.getText());

    tb.setKategoribunga(TxtKategoriBunga.getText());

    tb.setDeskripsiproduk(TxtDeskripsiProduk.getText());

    tb.setHarga(Integer.parseInt(TxtHarga.getText()));

    tb.setStok(Integer.parseInt(TxtStokBunga.getText()));

    tb.setUkuran(TxtUkuranBuket.getText());

    em.persist(tb);

    em.getTransaction().commit();

    JOptionPane.showMessageDialog(this, "Data berhasil diperbarui!");

    dispose();

    em.close();

    emf.close();

} catch (Exception e) {

    JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());

}

}

}

// public static void main(String args[]) {

//     java.awt.EventQueue.invokeLater(new Runnable() {

//         public void run() {

//             InsertDialog dialog = new InsertDialog(new java.awt.Frame(), true);

//             dialog.addWindowListener(new java.awt.event.WindowAdapter() {

//                 public void windowClosing(java.awt.event.WindowEvent e) {

//                     System.exit(0);

//                 }

//             });

//             dialog.setVisible(true);

//         }

//     });

// }

```



```
// Variables declaration - do not modify

private javax.swing.JButton BtnInsert;
private javax.swing.JTextField TxtDeskripsiProduk;
private javax.swing.JTextField TxtHarga;
private javax.swing.JTextField TxtIdBunga;
private javax.swing.JTextField TxtKategoriBunga;
private javax.swing.JTextField TxtNamaBunga;
private javax.swing.JTextField TxtStokBunga;
private javax.swing.JTextField TxtUkuranBuket;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel1;

// End of variables declaration
}
```

## 2.10. Membuat JDialog Bernama “InsertDialog2”



Gambar 18 Insert 2

Berikut *code* nya:

```
package TokoBunga;

import TokoBunga.Transaksitokobunga;

import TokoBunga.InsertDialog;

import java.text.*;
import java.util.*;
import javax.persistence.*;
import javax.swing.*;
import javax.swing.table.*;

public class InsertDialog2 extends java.awt.Dialog {

    private EntityManagerFactory emf;

    private EntityManager em;

    public void connect() {

        try {

            emf = Persistence.createEntityManagerFactory("PertemuanKesebelasPU");

            em = emf.createEntityManager();

        } catch (Exception e) {
```

```
JOptionPane.showMessageDialog(this, "Koneksi gagal: " + e.getMessage());
```

```
}
```

```
}
```

```
public InsertDialog2(java.awt.Frame parent, boolean modal) {
```

```
    super(parent, modal);
```

```
    initComponents();
```

```
    connect();
```

```
    TxtIdBunga.addKeyListener(new java.awt.event.KeyAdapter() {
```

```
        public void keyReleased(java.awt.event.KeyEvent evt) {
```

```
            hitungTotalHarga();
```

```
        }
```

```
    });
```

```
    SpJumlahPesanan.addPropertyChangeListener(evt -> {
```

```
        if ("value".equals(evt.getPropertyName())) {
```

```
            hitungTotalHarga();
```

```
        }
```

```
    });
```

```
}
```

```
[Generate Code]
```

```
private void closeDialog(java.awt.event.WindowEvent evt) {
```

```
    setVisible(false);
```

```
    dispose();
```

```
}
```

```
private void hitungTotalHarga() {
```

```
    try {
```

```
        String idBunga = TxtIdBunga.getText().trim();
```

```
        if (idBunga.isEmpty()) {
```

```
            return;
```

```
        }
```

```

Tokobunga_1 bunga = em.find(Tokobunga_1.class, idBunga);

    if (bunga == null) {

        TxtTotalHarga.setText("");

        return;

    }

    int hargaAsli = bunga.getHarga();

    int jumlah = (int) SpJumlahPesanan.getValue();

    int total = hargaAsli * jumlah;

    TxtTotalHarga.setText(String.valueOf(total));

} catch (Exception e) {

    System.out.println("Error hitung total: " + e.getMessage());

}

}

private void TxtTotalHargaActionPerformed(java.awt.event.ActionEvent evt) {

    hitungTotalHarga();

}

private void BtnInsert2ActionPerformed(java.awt.event.ActionEvent evt) {

    int jumlah = (Integer) SpJumlahPesanan.getValue();

    if (TxtIdTransaksi.getText().equals(""))

        || TxtIdBunga.getText().equals("")

        || TxtNamaPelanggan.getText().equals("")

        || DtcTanggalTransaksi.getDate() == null

        || jumlah == 0

        || TxtTotalHarga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

        return;

    }

try {

    String IdTransaksi = TxtIdTransaksi.getText();

    String IdBunga = TxtIdBunga.getText();

    String NamaPelanggan = TxtNamaPelanggan.getText();

    Date Tanggal = DtcTanggalTransaksi.getDate();

```

```

int Jumlah = jumlah;

int TotalHarga = Integer.parseInt(TxtTotalHarga.getText());

emf = Persistence.createEntityManagerFactory("PertemuanKesebelasPU");

em = emf.createEntityManager();

em.getTransaction().begin();

Tokobunga_1 bunga = em.find(Tokobunga_1.class, TxtIdBunga.getText());

if (bunga == null) {

    JOptionPane.showMessageDialog(null, "ID Bunga tidak ditemukan!", "Error",
JOptionPane.ERROR_MESSAGE);

    return;

}

Transaksitokobunga trx = em.find(Transaksitokobunga.class, TxtIdTransaksi.getText());

if (trx != null) {

    JOptionPane.showMessageDialog(null, "ID Transaksi sudah ada!", "Error",
JOptionPane.ERROR_MESSAGE);

    return;

}

Transaksitokobunga transaksiBaru = new Transaksitokobunga();

transaksiBaru.setIdtransaksi(IdTransaksi);

transaksiBaru.setIdbunga(bunga);

transaksiBaru.setNamapelanggan>NamaPelanggan);

transaksiBaru.setTanggal(Tanggal);

transaksiBaru.setJumlah(Jumlah);

transaksiBaru.setTotalharga(TotalHarga);

```

```

        em.persist(transaksiBaru);

        em.getTransaction().commit();
OptionPane.showMessageDialog(null, "Data transaksi berhasil disimpan!");

        TxtIdTransaksi.setText("");
        TxtIdBunga.setText("");
        TxtNamaPelanggan.setText("");
        DtcTanggalTransaksi.setDate(null);
        SpJumlahPesanan.setValue(0);
        TxtTotalHarga.setText("");
        dispose();
    } catch (Exception e) {
        if (em.getTransaction().isActive()) {
            em.getTransaction().rollback();
        }

        JOptionPane.showMessageDialog(this, "Terjadi kesalahan: " +
e.getMessage());
    } finally {
        if (em != null) {
            em.close();
        }

        if (emf != null) {
            emf.close();
        }
    }
}

```

```

// public static void main(String args[]) {
//     java.awt.EventQueue.invokeLater(new Runnable() {
//         public void run() {
//             InsertDialog2 dialog = new InsertDialog2(new java.awt.Frame(), true);
//             dialog.addWindowListener(new java.awt.event.WindowAdapter() {
//                 public void windowClosing(java.awt.event.WindowEvent e) {
//                     System.exit(0);
//                 }
//             });
//             dialog.setVisible(true);
//         }
//     });
// }

// Variables declaration - do not modify

private javax.swing.JButton BtnInsert2;

private com.toedter.calendar.JDateChooser DtcTanggalTransaksi;

private com.toedter.components.JSpinField SpJumlahPesanan;

private javax.swing.JTextField TxtIdBunga;

private javax.swing.JTextField TxtIdTransaksi;

private javax.swing.JTextField TxtNamaPelanggan;

private javax.swing.JTextField TxtTotalHarga;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JPanel jPanel1;

// End of variables declaration
}

```

## 2.11. Membuat JDialog Bernama “UpdateDialog”



The image shows a Java Swing dialog box titled "UpdateDialog" with a pink background. It contains seven text input fields, each with a label to its left: "ID Bunga", "Nama Bunga", "Kategori Bunga", "Deskripsi Produk", "Harga", "Stok Bunga", and "Ukuran Buket". An "Update" button is positioned at the bottom right of the dialog.

Gambar 19 Update 1

Berikut *code* nya:

```
package TokoBunga;

import java.io.*;
import java.util.logging.*;
import javax.swing.*;
import javax.swing.table.*;
import javax.persistence.*;
import java.util.List;

public class UpdateDialog extends java.awt.Dialog {

    private EntityManagerFactory emf;

    private EntityManager em;

    String idBungaLama, namaBungaLama, kategoriBungaLama,
    deskripsiLama, hargaLama, stokLama, ukuranLama;
```



```

public void connect() {
    try {
        emf = Persistence.createEntityManagerFactory("PertemuanKesebelasPU");
        em = emf.createEntityManager();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Koneksi gagal: " + e.getMessage());
    }
}

public UpdateDialog(java.awt.Frame parent, boolean modal, String idBungaLama, String namaBungaLama, String
kategoriBungaLama,
    String deskripsiLama, String hargaLama, String stokLama, String ukuranLama) {
    super(parent, modal);
    initComponents();
    connect();
    this.idBungaLama = idBungaLama;
    this.namaBungaLama = namaBungaLama;
    this.kategoriBungaLama = kategoriBungaLama;
    this.deskripsiLama = deskripsiLama;
    this.hargaLama = hargaLama;
    this.stokLama = stokLama;
    this.ukuranLama = ukuranLama;
    TxtIdBunga.setText(idBungaLama);
    TxtNamaBunga.setText(namaBungaLama);
    TxtKategoriBunga.setText(kategoriBungaLama);
    TxtDeskripsiProduk.setText(deskripsiLama);
    TxtHarga.setText(hargaLama);
    TxtStokBunga.setText(stokLama);
    TxtUkuranBuket.setText(ukuranLama);
}

```

```

private void closeDialog(java.awt.event.WindowEvent evt) {
    setVisible(false);
    dispose();
}

private void BtnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    if (TxtIdBunga.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Isi semua data");
    } else if (TxtNamaBunga.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Isi semua data");
    } else if (TxtKategoriBunga.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Isi semua data");
    } else if (TxtDeskripsiProduk.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Isi semua data");
    } else if (TxtHarga.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Isi semua data");
    } else if (TxtStokBunga.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Isi semua data");
    } else if (TxtUkuranBuket.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Isi semua data");
    } else {
try {
        EntityManager em = emf.createEntityManager();
        em.getTransaction().begin();

        TypedQuery<Tokobunga_1> query = em.createQuery(
            "SELECT t FROM Tokobunga_1 t WHERE t.idbunga = :idbunga AND t.namabunga = :namabunga "
            + "AND t.kategoribunga = :kategoribunga AND t.deskripsiproduk = :deskripsiproduk AND t.harga = :harga "
            + "AND t.stok = :stok AND t.ukuran = :ukuran", Tokobunga_1.class);
        query.setParameter("idbunga", idBungaLama);
        query.setParameter("namabunga", namaBungaLama);
        query.setParameter("kategoribunga", kategoriBungaLama);

```

```

query.setParameter("deskripsiproduk", deskripsiLama);

    query.setParameter("harga", Integer.parseInt(hargaLama));

    query.setParameter("stok", Integer.parseInt(stokLama));

    query.setParameter("ukuran", ukuranLama);

    List<Tokobunga_1> hasil = query.getResultList();

    if (!hasil.isEmpty()) {

        Tokobunga_1 tb = hasil.get(0);

        tb.setIdbunga(TxtIdBunga.getText());

        tb.setNamabunga(TxtNamaBunga.getText());

        tb.setKategoribunga(TxtKategoriBunga.getText());

        tb.setDeskripsiproduk(TxtDeskripsiProduk.getText());

        tb.setHarga(Integer.parseInt(TxtHarga.getText()));

        tb.setStok(Integer.parseInt(TxtStokBunga.getText()));

        tb.setUkuran(TxtUkuranBuket.getText());

em.persist(tb);

        em.getTransaction().commit();

        JOptionPane.showMessageDialog(this, "Data berhasil diperbarui!");

        dispose();

    }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(this, "Error: " + e.getMessage());

    }

    }

}

// public static void main(String args[]) {
//     java.awt.EventQueue.invokeLater(new Runnable() {
//         public void run() {
//             UpdateDialog dialog = new UpdateDialog(new java.awt.Frame(), true);

```

```

//      dialog.addWindowListener(new java.awt.event.WindowAdapter() {
//          public void windowClosing(java.awt.event.WindowEvent e) {
//              System.exit(0);
//          }
//      });
//      dialog.setVisible(true);
//  }
//  });
//  }

// Variables declaration - do not modify
private javax.swing.JButton BtnUpdate;
private javax.swing.JTextField TxtDeskripsiProduk;
private javax.swing.JTextField TxtHarga;
private javax.swing.JTextField TxtIdBunga;
private javax.swing.JTextField TxtKategoriBunga;
private javax.swing.JTextField TxtNamaBunga;
private javax.swing.JTextField TxtStokBunga;
private javax.swing.JTextField TxtUkuranBuket;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel1;
// End of variables declaration
}

```

## 2.12. Membuat JDialog Bernama “UpdateDialog2”



The screenshot shows a Java Swing dialog box titled "UpdateDialog2". It has a pink background and a white border. Inside, there are six labels on the left and corresponding input fields on the right. The labels are: "ID Transaksi", "ID Bunga", "Nama Pelanggan", "Tanggal Transaksi", "Jumlah Pesanan", and "Total Harga". The input fields are: a text box for "ID Transaksi", a text box for "ID Bunga", a text box for "Nama Pelanggan", a date picker for "Tanggal Transaksi", a spinner for "Jumlah Pesanan" (showing 0), and a text box for "Total Harga". A pink "Update" button is located at the bottom right of the dialog.

Gambar 20 Update 2

Berikut *code* nya:

```
package TokoBunga;

import TokoBunga.Transaksitokobunga;

import java.text.*;
import java.util.*;
import javax.persistence.*;
import javax.swing.*;
import javax.swing.table.*;

public class UpdateDialog2 extends java.awt.Dialog {

    private EntityManagerFactory emf;

    private EntityManager em;

    String idTransaksiLama, idBungaLama, namaPelangganLama;

    Date tanggalLama;

    int jumlahLama, totalHargaLama;

    public void connect() {

        try {
```

```

        emf = Persistence.createEntityManagerFactory("PertemuanKesebelasPU");
        em = emf.createEntityManager();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Koneksi gagal: " + e.getMessage());
    }
}

```

```

public UpdateDialog2(java.awt.Frame parent, boolean modal,
    String idTransaksi, String idBunga, String namaPelanggan,
    Date tanggal, int jumlah, int totalHarga) {
    super(parent, modal);
    initComponents();
    connect();
}

```

```

this.idTransaksiLama = idTransaksi;
this.idBungaLama = idBunga;
this.namaPelangganLama = namaPelanggan;
this.tanggalLama = tanggal;
this.jumlahLama = jumlah;
this.totalHargaLama = totalHarga;

TxtIdTransaksi.setText(idTransaksi);
TxtIdBunga.setText(idBunga);
TxtNamaPelanggan.setText(namaPelanggan);
DtcTanggalTransaksi.setDate(tanggal);
SpJumlahPesanan.setValue(jumlah);
TxtTotalHarga.setText(String.valueOf(totalHarga));
}

```

[Generate Code]

```

private void closeDialog(java.awt.event.WindowEvent evt) {
    setVisible(false);
    dispose();
}

```

```

private void BtnUpdate2ActionPerformed(java.awt.event.ActionEvent evt) {

    int jumlah = (Integer) SpJumlahPesanan.getValue();

    if (TxtIdTransaksi.getText().trim().equals("")
        || TxtIdBunga.getText().trim().equals("")
        || TxtNamaPelanggan.getText().trim().equals("")
        || DtcTanggalTransaksi.getDate() == null
        || jumlah == 0
        || TxtTotalHarga.getText().trim().equals("")) {

        JOptionPane.showMessageDialog(this, "Isi semua data!");
        return;
    }
    try {
        String idTransaksiBaru = TxtIdTransaksi.getText().trim();
        String idBungaBaru = TxtIdBunga.getText().trim();
        String namaPelangganBaru = TxtNamaPelanggan.getText().trim();
        Date tanggalBaru = DtcTanggalTransaksi.getDate();
        int jumlahBaru = jumlah;
        int totalHargaBaru;
        try {
            totalHargaBaru = Integer.parseInt(TxtTotalHarga.getText().trim());
        } catch (NumberFormatException nfe) {
            JOptionPane.showMessageDialog(this, "Total harga harus berupa angka!");
            return;
        }
    }
    em.getTransaction().begin();

    if (!idTransaksiBaru.equals(idTransaksiLama)) {
        JOptionPane.showMessageDialog(this, "ID Transaksi tidak boleh diubah!", "Peringatan",
JOptionPane.WARNING_MESSAGE);
        TxtIdTransaksi.setText(idTransaksiLama);
    }
}

```

```

em.getTransaction().rollback();

    return;
}

Transaksitokobunga trx = em.find(Transaksitokobunga.class, idTransaksiLama);
if (trx == null) {
    JOptionPane.showMessageDialog(this, "Transaksi tidak ditemukan!");
    em.getTransaction().rollback();
    return;
}

Tokobunga_1 bunga = em.find(Tokobunga_1.class, idBungaBaru);

if (bunga == null) {
    try {
        TypedQuery<Tokobunga_1> q = em.createQuery(
            "SELECT t FROM Tokobunga_1 t WHERE LOWER(t.namabunga) = :nm", Tokobunga_1.class);
        q.setParameter("nm", idBungaBaru.toLowerCase());
        bunga = q.getSingleResult();
    } catch (NoResultException nre) {
        JOptionPane.showMessageDialog(this, "ID / Nama Bunga tidak ditemukan!", "Error",
JOptionPane.ERROR_MESSAGE);
        em.getTransaction().rollback();
        return;
    }
}

trx.setIdbunga(bunga);

trx.setNamapelanggan(namaPelangganBaru);
trx.setTanggal(tanggalBaru);
trx.setJumlah(jumlahBaru);
trx.setTotalharga(totalHargaBaru);
em.merge(trx);
em.getTransaction().commit();

```



```

JOptionPane.showMessageDialog(this, "Data transaksi berhasil diperbarui!");

        dispose();
    } catch (Exception e) {
        if (em.getTransaction().isActive()) {
            em.getTransaction().rollback();
        }
        JOptionPane.showMessageDialog(this, "Terjadi kesalahan: " + e.getMessage());
    }
}

// public static void main(String args[]) {
//     java.awt.EventQueue.invokeLater(new Runnable() {
//         public void run() {
//             UpdateDialog2 dialog = new UpdateDialog2(new java.awt.Frame(), true);
//             dialog.addWindowListener(new java.awt.event.WindowAdapter() {
//                 public void windowClosing(java.awt.event.WindowEvent e) {
//                     System.exit(0);
//                 }
//             });
//             dialog.setVisible(true);
//         }
//     });
// }

// Variables declaration - do not modify
private javax.swing.JButton BtnUpdate2;

private com.toedter.calendar.JDateChooser DtcTanggalTransaksi;

private com.toedter.components.JSpinField SpJumlahPesanan;

private javax.swing.JTextField TxtIdBunga;

private javax.swing.JTextField TxtIdTransaksi;

private javax.swing.JTextField TxtNamaPelanggan;

private javax.swing.JTextField TxtTotalHarga;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

```

```

private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JPanel jPanel1;
// End of variables declaration
}

```

### 2.13. Membuat JDialog Bernama “DeleteDialog”

Gambar 21 Delete 1

Berikut *code* nya:

```

package TokoBunga;

import java.io.*;
import java.util.logging.*;
import javax.swing.*;
import javax.swing.table.*;
import javax.persistence.*;
import java.util.List;

```

```

public class DeleteDialog extends java.awt.Dialog {

    String idBungaLama, namaBungaLama, kategoriBungaLama, deskripsiLama,
    hargaLama, stokLama, ukuranLama;

    private EntityManagerFactory emf;

    private EntityManager em;

    public void connect() {

        try {

            emf = Persistence.createEntityManagerFactory("PertemuanKesebelasPU");

            em = emf.createEntityManager();

        } catch (Exception e) {

            JOptionPane.showMessageDialog(this, "Koneksi gagal: " + e.getMessage());

        }

    }

    public DeleteDialog(java.awt.Frame parent, boolean modal, String idBungaLama, String
    namaBungaLama, String kategoriBungaLama,

        String deskripsiLama, String hargaLama, String stokLama, String ukuranLama) {

        super(parent, modal);

        initComponents();

        connect();

        this.idBungaLama = idBungaLama;

        this.namaBungaLama = namaBungaLama;

        this.kategoriBungaLama = kategoriBungaLama;

        this.deskripsiLama = deskripsiLama;

        this.hargaLama = hargaLama;

        this.stokLama = stokLama;

        this.ukuranLama = ukuranLama;

        TxtIdBunga.setText(idBungaLama);

        TxtNamaBunga.setText(namaBungaLama);

        TxtKategoriBunga.setText(kategoriBungaLama);

        TxtDeskripsiProduk.setText(deskripsiLama);

```

```

TxtHarga.setText(hargaLama);

TxtStokBunga.setText(stokLama);

TxtUkuranBuket.setText(ukuranLama);


TxtIdBunga.setEditable(false);

TxtNamaBunga.setEditable(false);

TxtKategoriBunga.setEditable(false);

TxtDeskripsiProduk.setEditable(false);

TxtHarga.setEditable(false);

TxtUkuranBuket.setEditable(false);

}

[Generate Code]

private void closeDialog(java.awt.event.WindowEvent evt) {

    setVisible(false);

    dispose();

}

private void BtnYaActionPerformed(java.awt.event.ActionEvent evt) {

    if (TxtIdBunga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtNamaBunga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtKategoriBunga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtDeskripsiProduk.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtHarga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtStokBunga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else if (TxtUkuranBuket.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");

    } else {

```

```

try {

    String idbunga;

    idbunga = TxtIdBunga.getText();

    EntityManagerFactory emf =
Persistence.createEntityManagerFactory("PertemuanKesebelasPU");

    EntityManager em = emf.createEntityManager();


    em.getTransaction().begin();
    Tokobunga_1 tb = em.find(Tokobunga_1.class, idbunga);
    if (tb == null) {
        JOptionPane.showMessageDialog(null, "Data tidak ditemukan");
    } else {
        em.remove(tb);
        em.getTransaction().commit();
        JOptionPane.showMessageDialog(null, "Data Berhasil Dihapus");
        TxtIdBunga.setEditable(false);
        dispose();
    }
} catch (Exception e) {
    em.close();
    emf.close();
}
}
}

```

```

// public static void main(String args[]) {
//     java.awt.EventQueue.invokeLater(new Runnable() {
//         public void run() {
//             DeleteDialog dialog = new DeleteDialog(new java.awt.Frame(), true);
//             dialog.addWindowListener(new java.awt.event.WindowAdapter() {
//                 public void windowClosing(java.awt.event.WindowEvent e) {
//                     System.exit(0);
//                 }
//             });
//             dialog.setVisible(true);
//         }
//     });
// }

// Variables declaration - do not modify
private javax.swing.JButton BtnTidak;
private javax.swing.JButton BtnYa;
private javax.swing.JTextField TxtDeskripsiProduk;
private javax.swing.JTextField TxtHarga;
private javax.swing.JTextField TxtIdBunga;
private javax.swing.JTextField TxtKategoriBunga;
private javax.swing.JTextField TxtNamaBunga;
private javax.swing.JTextField TxtStokBunga;
private javax.swing.JTextField TxtUkuranBuket;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel1;

// End of variables declaration

```

## 2.14. Membuat JDialog Bernama “DeleteDialog2”

Gambar 22 Delete 2

Berikut *code* nya:

```
package TokoBunga;

import TokoBunga.Transaksitokobunga;
import java.text.*;
import java.util.*;
import javax.persistence.*;
import javax.swing.*;
import javax.swing.table.*;

public class DeleteDialog2 extends java.awt.Dialog {

    private EntityManagerFactory emf;
    private EntityManager em;

    private String idTransaksiLama, idBungaLama, namaPelangganLama;
    private Date tanggalLama;
    private int jumlahLama, totalHargaLama;
```

```

public void connect() {

    try {

        emf = Persistence.createEntityManagerFactory("PertemuanKesebelasPU");

        em = emf.createEntityManager();

    } catch (Exception e) {

        JOptionPane.showMessageDialog(this, "Koneksi gagal: " + e.getMessage());

    }

}

public DeleteDialog2(java.awt.Frame parent, boolean modal,

    String idTransaksi, String idBunga, String namaPelanggan,

    Date tanggal, int jumlah, int totalHarga) {

    super(parent, modal);

    initComponents();

    connect();

    this.idTransaksiLama = idTransaksi;

    this.idBungaLama = idBunga;

    this.namaPelangganLama = namaPelanggan;

    this.tanggalLama = tanggal;

    this.jumlahLama = jumlah;

    this.totalHargaLama = totalHarga;

    TxtIdTransaksi.setText(idTransaksi);

    TxtIdBunga.setText(idBunga);

    TxtNamaPelanggan.setText(namaPelanggan);

    DtcTanggalTransaksi.setDate(tanggal);

    SpJumlahPesanan.setValue(jumlah);

    TxtTotalHarga.setText(String.valueOf(totalHarga));

    TxtIdTransaksi.setEditable(false);

    TxtIdBunga.setEditable(false);

    TxtNamaPelanggan.setEditable(false);

    DtcTanggalTransaksi.setEnabled(false);

    SpJumlahPesanan.setEnabled(false);

    TxtTotalHarga.setEditable(false);

}

```



```

private void closeDialog(java.awt.event.WindowEvent evt) {
    setVisible(false);
    dispose();
}

private void BtnTidak2ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

private void BtnYa2ActionPerformed(java.awt.event.ActionEvent evt) {
    int jumlah = (Integer) SpJumlahPesanan.getValue();
    if (TxtIdTransaksi.getText().equals(""))
        || TxtIdBunga.getText().equals("")
        || TxtNamaPelanggan.getText().equals("")
        || DtcTanggalTransaksi.getDate() == null
        || jumlah == 0
        || TxtTotalHarga.getText().equals("")) {

        JOptionPane.showMessageDialog(null, "Isi semua data");
    } else {
        EntityManagerFactory emf = null;
        EntityManager em = null;
    try {
        emf = Persistence.createEntityManagerFactory("PertemuanKesebelasPU");
        em = emf.createEntityManager();

        String idTransaksi = TxtIdTransaksi.getText();

        em.getTransaction().begin();
        Transaksitokobunga trx = em.find(Transaksitokobunga.class, idTransaksiLama);

        if (trx == null) {
            JOptionPane.showMessageDialog(null, "Data tidak ditemukan");
        } else {

```

```

em.remove(trx);

        em.getTransaction().commit();

        JOptionPane.showMessageDialog(null, "Data Berhasil Dihapus");

        dispose();
    }
} catch (Exception e) {

    if (em != null && em.getTransaction().isActive()) {

        em.getTransaction().rollback();

    }

    JOptionPane.showMessageDialog(null, "Terjadi kesalahan: " + e.getMessage());
} finally {

    if (em != null) {

        em.close();

    }

    if (emf != null) {

        emf.close();

    }

}

}

}

// public static void main(String args[]) {
//     java.awt.EventQueue.invokeLater(new Runnable() {
//         public void run() {
//             DeleteDialog2 dialog = new DeleteDialog2(new java.awt.Frame(), true);
//             dialog.addWindowListener(new java.awt.event.WindowAdapter() {
//                 public void windowClosing(java.awt.event.WindowEvent e) {
//                     System.exit(0);
//                 }
//             });
//             dialog.setVisible(true);
//         }
//     });
// }

```

```
// Variables declaration - do not modify

private javax.swing.JButton BtnTidak2;

private javax.swing.JButton BtnYa2;

private com.toedter.calendar.JDateChooser DtcTanggalTransaksi;

private com.toedter.components.JSpinField SpJumlahPesanan;

private javax.swing.JTextField TxtIdBunga;

private javax.swing.JTextField TxtIdTransaksi;

private javax.swing.JTextField TxtNamaPelanggan;

private javax.swing.JTextField TxtTotalHarga;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JPanel jPanel1;


// End of variables declaration

}
```

## 2.15. Membuat jasper Reports

Data Buket Bunga					
			Toko Bunga		
ID Bunga	Nama Bunga	Kategori Buket	Deskripsi Produk	Harga	Stok Ukuran
\$F	\$F	\$F	\$F{deskripsi_produk}	\$F	\$F
new java.util.Date()			"Page "+\$V+" "		

		<p>Flowers</p> <p>There are two flowers, the first one is printed only starting from the second page, the other one only in the first page.</p>
		Background
<p>Keterangan Ukuran Buket:</p> <p>1. K= Kecil</p> <p>2. M= Medium</p> <p>3. B= Besar</p>		

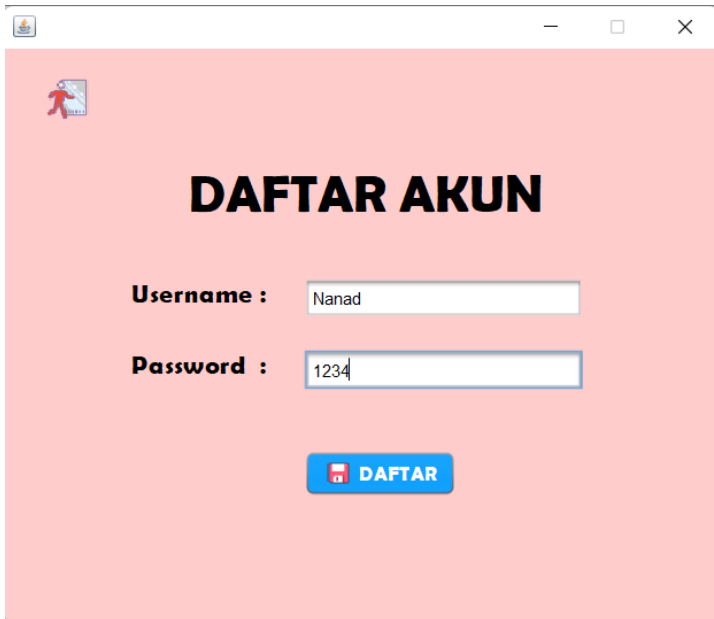
Data bunga

Gambar 23 Jasper reports

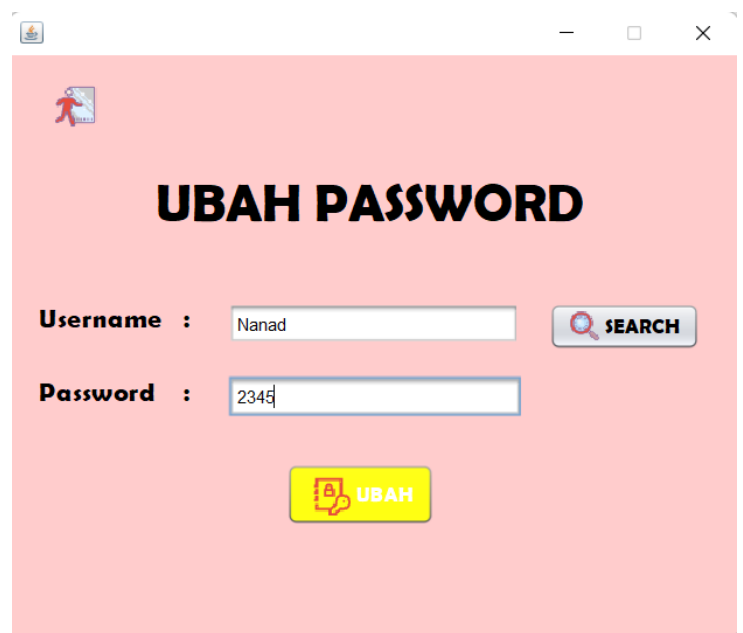
data transaksi

Data Pesanan Buket					
			Toko Bunga		
ID Transaksi	ID Bunga	Nama Pelanggan	Tanggal Transaksi	Jumlah Pesanan	Total Harga
\$F	\$F	\$F	\$F{tanggal} == null ?	\$F{jumlah}	\$F
new java.util.Date()					
				"Page "+\$V+" "	

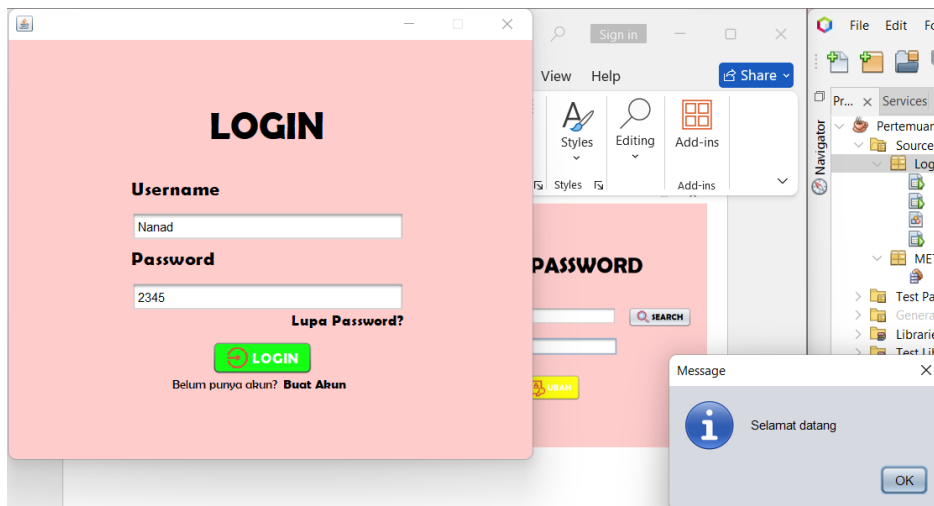
## 2.16. Hasil Running



A screenshot of a web application window titled "DAFTAR AKUN" (Register Account). The background is light pink. At the top left is a small icon of a person running. The title "DAFTAR AKUN" is in large, bold, black letters. Below the title, there are two input fields: "Username :" with the value "Nanad" and "Password :" with the value "1234". Below the password field is a blue button with a white icon of a person and the text "DAFTAR".



A screenshot of a web application window titled "UBAH PASSWORD" (Change Password). The background is light pink. At the top left is a small icon of a person running. The title "UBAH PASSWORD" is in large, bold, black letters. Below the title, there are two input fields: "Username :" with the value "Nanad" and "Password :" with the value "2345". To the right of the username field is a "SEARCH" button with a magnifying glass icon. Below the password field is a yellow button with a white icon of a key and the text "UBAH".



A screenshot of a web application window titled "LOGIN". The background is light pink. At the top left is a small icon of a person running. The title "LOGIN" is in large, bold, black letters. Below the title, there are two input fields: "Username" with the value "Nanad" and "Password" with the value "2345". Below the password field is a link "Lupa Password?". Below the login fields is a green button with a white icon of a person and the text "LOGIN". Below the button is the text "Belum punya akun? [Buat Akun](#)". In the background, there is a "Sign in" window with a "Share" button and a "Message" box with the text "Selamat datang" and an "OK" button.

Gambar 24 Hasil running

DATA PESANAN TOKO BUNGA

Data Buket

Data Pesanan

ID Bunga	Nama Bunga	Kategori	Deskripsi	Harga	Stok	Ukuran
A02	Anggrek	Buket Valen...	Simbol keang...	60000	50	M (25)
K03	Kamboja Jep...	Buket Special	Berbentuk u...	30000	20	K (10)
P01	Peony	Buket Valen...	Kelopaknya t...	50000	70	M (25)
M01	Mawar Merah	Buket Valen...	Cocok untuh ...	30000	30	K (10)
M03	Melati	Buket Special	Cocok untuh ...	30000	45	M (25)
M02	Matahari	Buket Friends	Cocok untuh ...	30000	35	M (25)
G01	Garberia Putih	Buket Friends	Cocok untuh ...	40000	60	h (25)

Insert

Delete

Upload

Update

Print

Download

DATA PESANAN TOKO BUNGA

Data Buket

Data Pesanan

ID Transaksi	ID Bunga	Nama Pelanggan	Tanggal Transaksi	Jumlah Pesanan	Total Harga
P14	Mawar Merah	Nina	15-Nov-2025	3	90000
P15	Mawar Merah	Nina	15-Nov-2025	3	90000
P12	Mawar Merah	Budi	22-Nov-2025	3	90000
P13	Mawar Merah	Budi	22-Nov-2025	3	90000

Insert

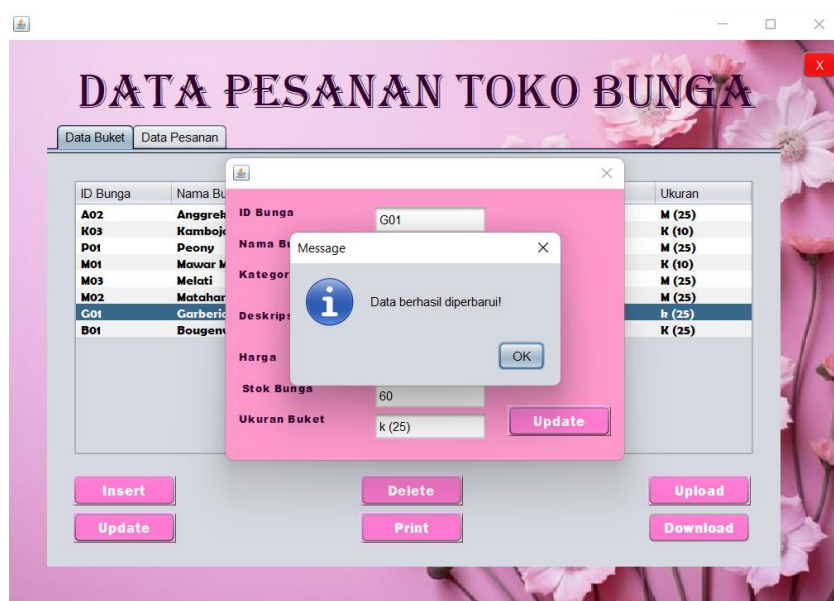
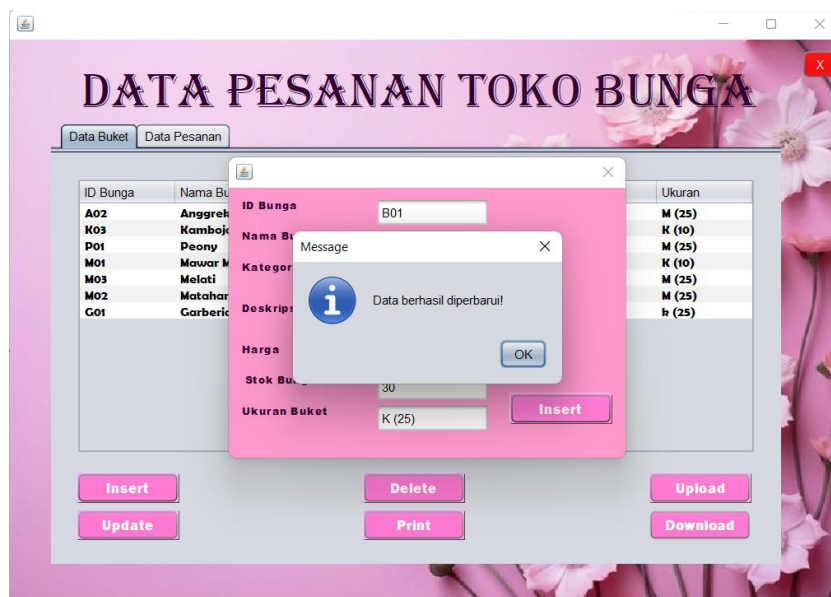
Delete

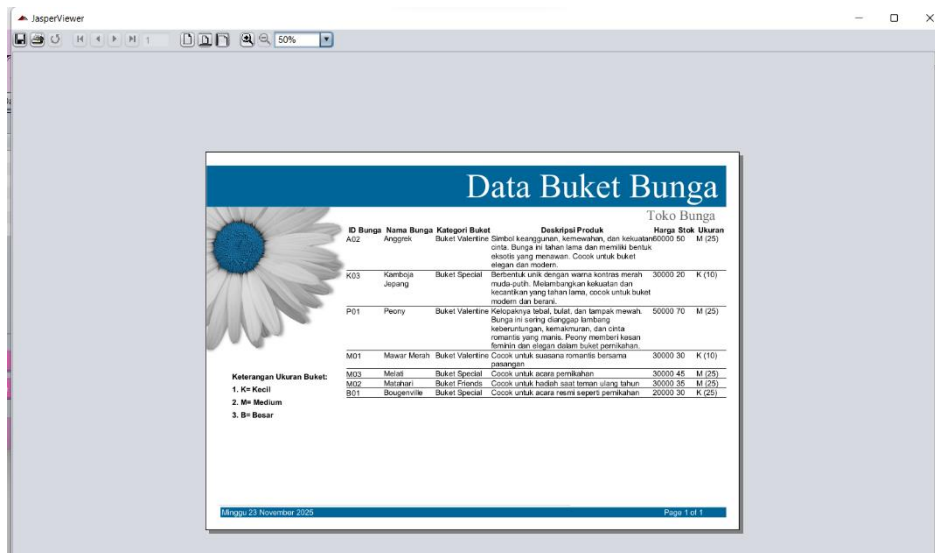
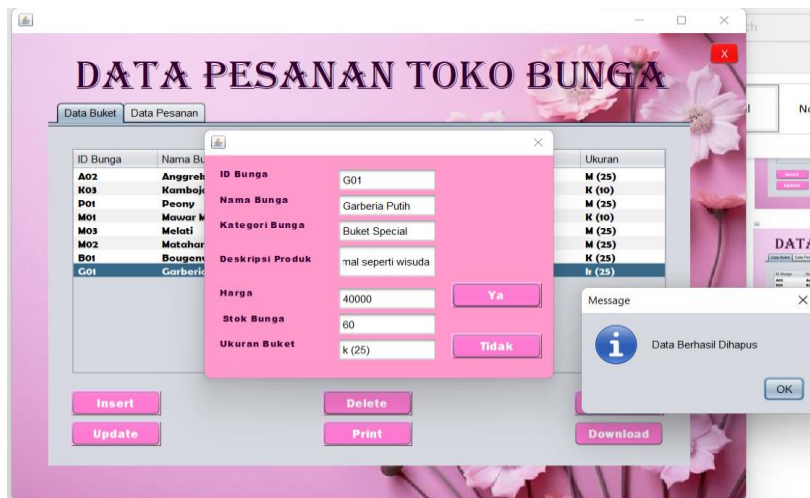
Upload

Update

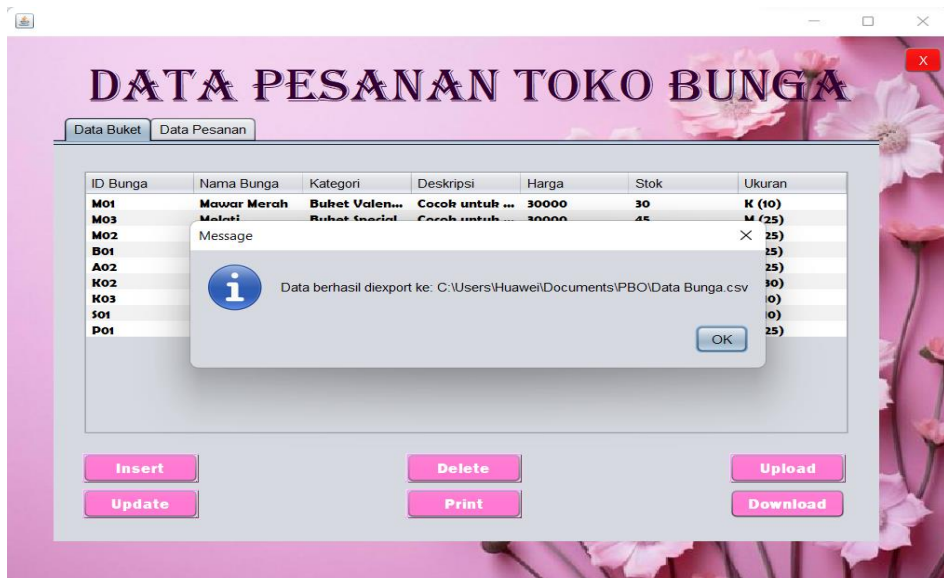
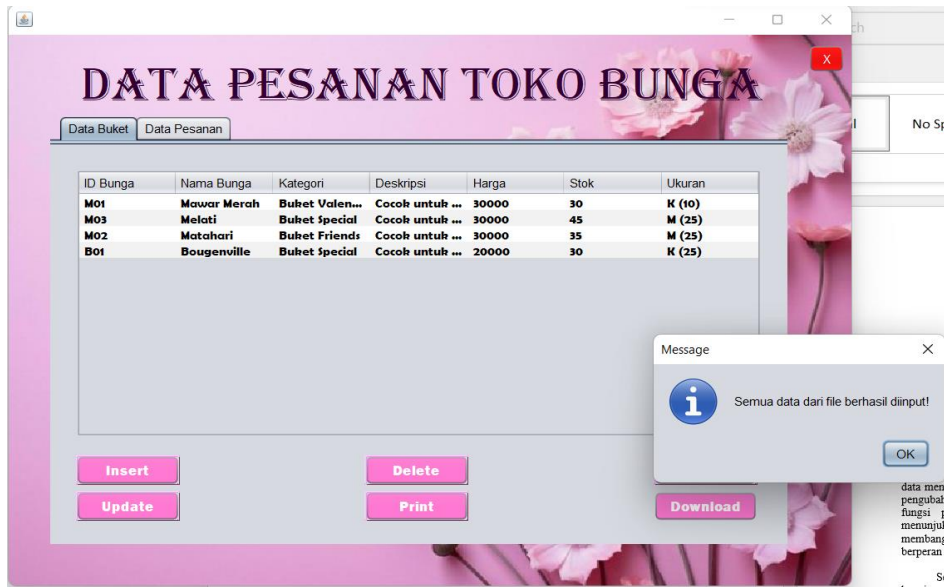
Print

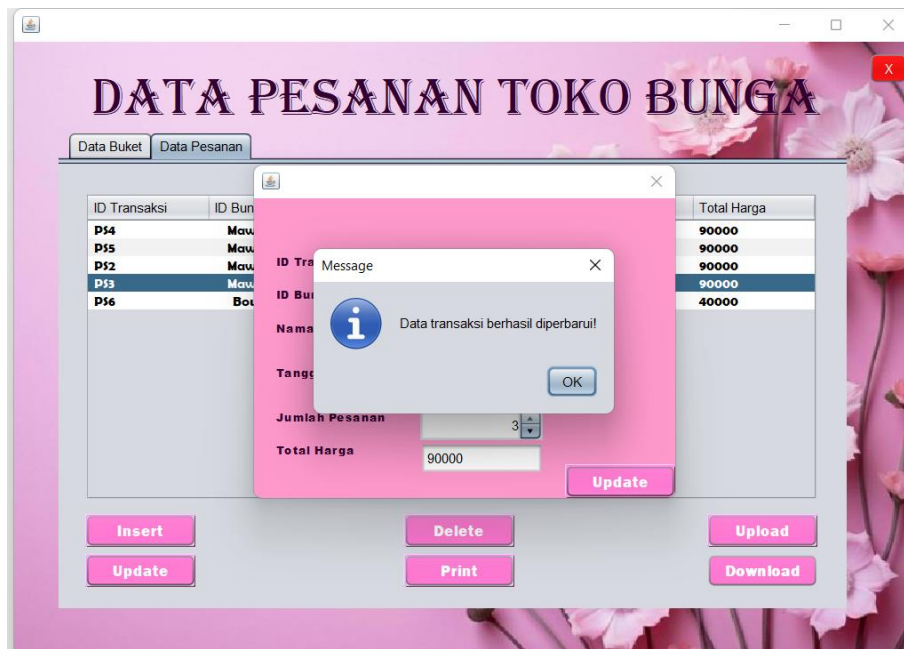
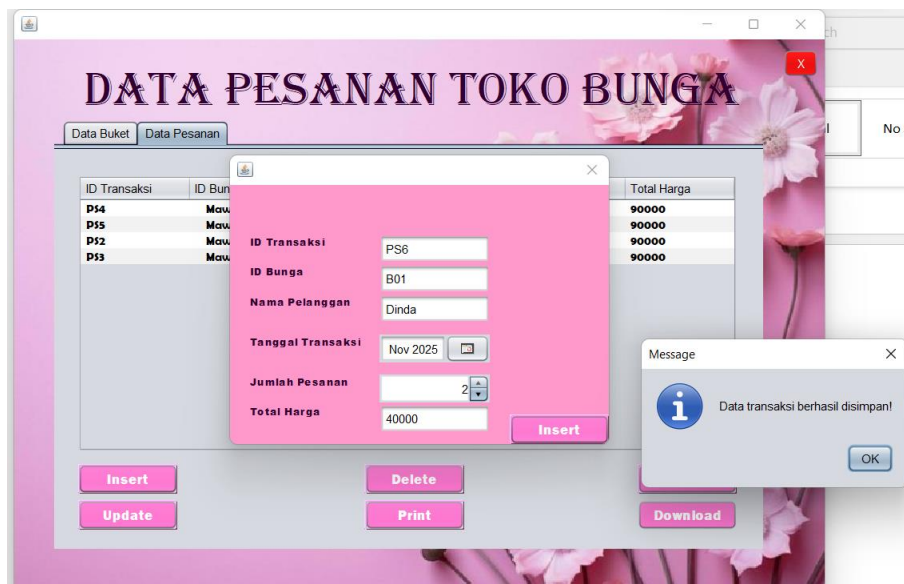
Download

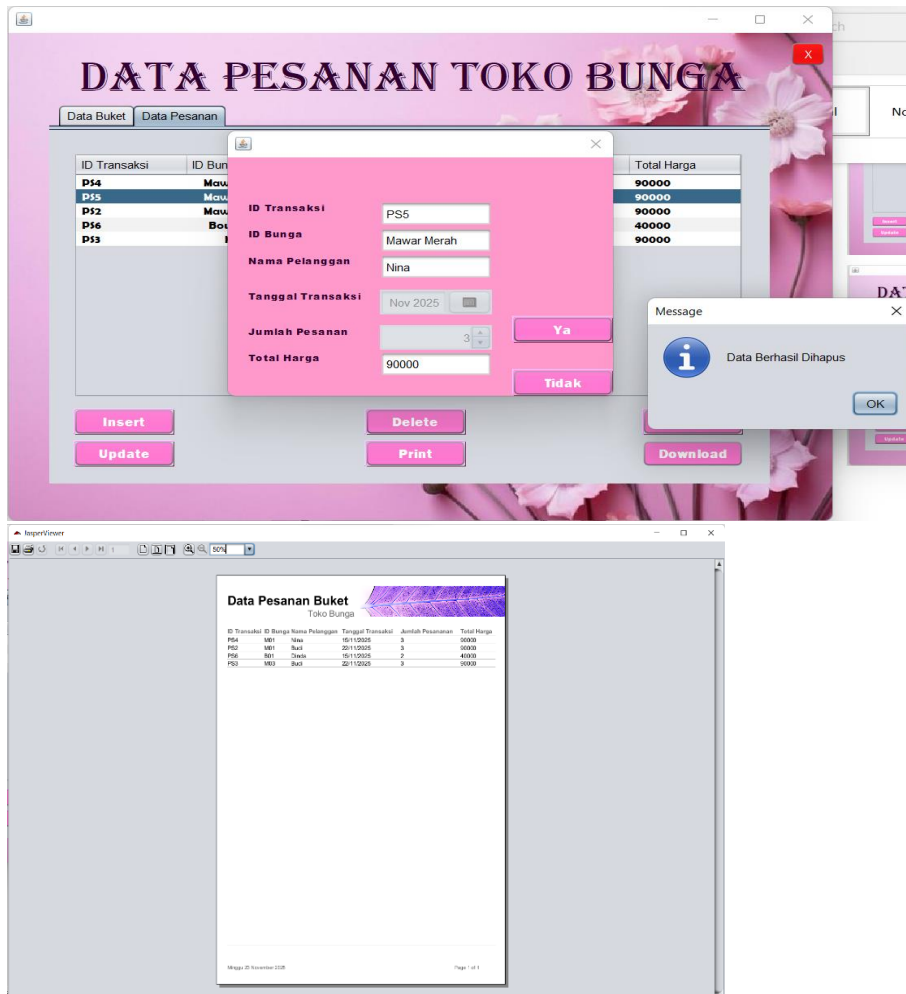


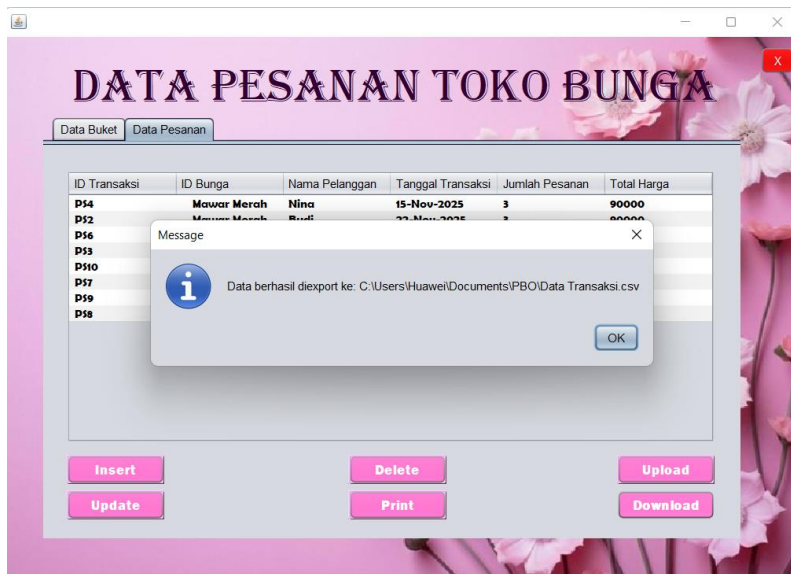












## BAB III

### PENUTUP

#### 3.1. Kesimpulan

Praktikum ini berhasil menerapkan pembuatan fitur Download (Export CSV) dan Security Question dalam aplikasi Java GUI dengan memanfaatkan berbagai *class* dan *method* yang telah dikembangkan. Fitur download direalisasikan melalui method `exportTableToCSV()`, yang dipanggil oleh `BtnDownloadActionPerformed()` dan `BtnDownload2ActionPerformed()` pada class `TokoBunga`, sehingga data pada `JTable` dapat disimpan menjadi file CSV yang mudah diakses pengguna.

Implementasi fitur keamanan juga berjalan baik melalui serangkaian class seperti `Login`, `DaftarAkun`, dan `UbahPassword`, yang masing-masing memanfaatkan method penting seperti `BtnLoginActionPerformed()`, `BtnDaftarActionPerformed()`, `BtnCariActionPerformed()`, dan `BtnUbahActionPerformed()`. Khusus fitur *security question*, mekanisme verifikasi pengguna dilakukan melalui proses pencarian akun menggunakan `EntityManager.find()` yang terintegrasi dalam class-class tersebut sehingga validasi identitas dapat dilakukan sebelum pengguna mengubah password atau melakukan tindakan penting lainnya.

Secara keseluruhan, praktikum ini memperkuat pemahaman tentang pembuatan aplikasi Java GUI yang interaktif serta aman. Penggunaan class `TokoBunga`, `InsertDialog`, `UpdateDialog`, `DeleteDialog`, serta entity `Login_1`, `Tokobunga_1`, dan `Transaksitokobunga` menunjukkan pemanfaatan OOP dan JPA secara optimal. Integrasi antara fitur download dan fitur keamanan membuktikan bahwa aplikasi dapat dibuat tidak hanya fungsional tetapi juga aman dan mudah digunakan oleh pengguna.

## DAFTAR PUSTAKA

- Al Godzali, G., Athallah, R. I., & Rivalni, E. (2025). Evaluasi Keamanan Autentikasi Pengguna pada Sistem Operasi Windows dan Linux. *Jurnal Ilmu Komputer Dan Teknologi Informasi*, 3(1), 31–40.
- Efendi, R., Wahyono, T., & Wideasari, I. R. (2024). Uji kerentanan keamanan pada web Sistem Informasi Akademik Satya Wacana menggunakan metode Vulnerability Assessment. *Aiti*, 21(1), 44–57. <https://doi.org/10.24246/aiti.v21i1.44-57>