

LAPORAN TUGAS BESAR
PEMROGRAMAN BERORIENTASI OBJEK
SISTEM TOKO ELEKTRONIK



Disusun Oleh :

NAMA : AVENIA ROHMATUN NISA

NIM : 32602200047

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2024

DAFTAR ISI

HALAMAN JUDUL.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	iii
BAB I Pendahuluan	1
1.1 Latar belakang	1
1.2 Tujuan.....	1
BAB II Konsep Dasar PBO	2
2.1 Inheritance	2
2.2 Polimorfisme	2
2.3 Encapsulation	2
2.4 Getter dan Setter	2
2.5 Interface	2
BAB III Struktur Program.....	3
3.1 File Utama (Toko Elektronik)	3
3.2 Kelas Produk	4
3.3 Kelas Elektronik	5
3.4 Interface Kategori	6
3.5 Antarmuka utama (MainForm)	6
3.6 Antarmuka tambah produk (ElektronikForm)	9
BAB IV Implementasi Program.....	12
4.1 Menjalankan Program	12
4.2 Menambahkan Produk.....	13
BAB V Kesimpulan	15

DAFTAR GAMBAR

Gambar 1. 1 File Utama Program Toko Elektronik	3
Gambar 1. 2 Kelas Produk 1	4
Gambar 1. 3 Kelas Produk 2	4
Gambar 1. 4 Kelas Elektronik 1	5
Gambar 1. 5 MainForm 1	7
Gambar 1. 6 MainForm 2	7
Gambar 1. 7 MainForm 3	7
Gambar 1. 8 ElektronikForm 1	9
Gambar 1. 9 ElektronikForm 2	9
Gambar 1. 10 ElektronikForm 3	10
Gambar 1. 11 ElektronikForm 4	10
Gambar 2. 1 Open Neatbeans	12
Gambar 2. 2 Open Projects	12
Gambar 2. 3 Arahkan ke project	12
Gambar 2. 4 Open struktur project	13
Gambar 2. 5 Run program	13
Gambar 2. 6 Tunggu program muncul	13
Gambar 2. 7 Tambah Elektronik	13
Gambar 2. 8 Isi Data, Klick Simpan	14
Gambar 2. 9 Keluar Dialog Pesanan	14

BAB I Pendahuluan

1.1 Latar belakang

Dalam era teknologi modern ini, bisnis toko elektronik semakin berkembang pesat. Dengan banyaknya jenis produk elektronik yang beragam, manajemen toko elektronik menjadi semakin kompleks. Oleh karena itu, diperlukan suatu sistem yang mampu mempermudah pengelolaan dan manajemen produk di dalam toko tersebut.

Latar belakang dibuatnya program Sistem Manajemen Toko Elektronik ini adalah sebagai respons terhadap kebutuhan akan solusi yang efisien dalam mengelola produk elektronik. Seiring dengan kemajuan teknologi, diperlukan suatu sistem yang tidak hanya dapat menyimpan informasi produk dengan baik tetapi juga memanfaatkan konsep-konsep pemrograman berorientasi objek agar dapat diintegrasikan dengan baik ke dalam struktur bisnis toko elektronik.

1.2 Tujuan

Tujuan utama dari pembuatan program ini adalah untuk memberikan solusi yang efisien dan mudah digunakan dalam manajemen produk pada sebuah toko elektronik. Dengan menerapkan konsep-konsep dasar pemrograman berorientasi objek, diharapkan program ini dapat memberikan fleksibilitas dan skalabilitas dalam mengelola produk, sekaligus memudahkan pengembangan sistem di masa depan.

Melalui penerapan konsep inheritance, polymorphism, encapsulation, getter dan setter, serta interface, diharapkan program ini dapat menjadi dasar yang kokoh untuk pengembangan lebih lanjut. Dengan kata lain, program ini dirancang not only for the present but also for the future, sehingga dapat terus berkembang sesuai dengan kebutuhan toko elektronik yang terus berubah

BAB II Konsep Dasar PBO

2.1 Inheritance

Konsep inheritance digunakan untuk menunjukkan hubungan hierarki antar kelas pada program. Pada program ini, terdapat kelas Produk sebagai kelas dasar yang diwarisi oleh kelas Elektronik.

2.2 Polimorfisme

Polymorphism digunakan untuk memungkinkan suatu objek dapat memiliki banyak bentuk (tipe data) dengan cara yang konsisten. Pada program ini, polymorphism diimplementasikan melalui metode getInfo() pada kelas Produk yang dioverride oleh kelas Elektronik.

2.3 Encapsulation

Encapsulation digunakan untuk menyembunyikan implementasi internal suatu objek dan hanya mengekspos fungsionalitas yang diperlukan. Pada program ini, atribut-atribut kelas seperti nama, harga, dan jenis dienkapsulasi dalam kelas Produk dan Elektronik.

2.4 Getter dan Setter

Getter dan setter digunakan untuk mengakses dan mengubah nilai atribut suatu objek. Dalam program ini, metode getter dan setter diterapkan pada kelas Produk untuk mengakses dan mengubah nilai atribut.

2.5 Interface

Interface digunakan untuk menyediakan kontrak untuk kelas-kelas yang mengimplementasikannya. Pada program ini, terdapat interface Kategori yang diimplementasikan oleh kelas Elektronik

BAB III Struktur Program

3.1 File Utama (Toko Elektronik)

File utama berisi kelas TokoElektronik yang merupakan entry point program dan menampilkan antarmuka utama.

```
package elektronik.avenia_rohmatun;

/*
author : avenia rohmatun
nim : 32602200047

package elektronik.avenia_rohmatun;; Mendefinisikan package atau paket dari kelas ini, yaitu elektronik.avenia_rohmatun.
SwingUtilities.invokeLater(...): Memastikan bahwa pembuatan dan tampilan GUI (Graphical User Interface)
dilakukan di EDT (Event Dispatch Thread) untuk menghindari potensi masalah concurrency.
new MainForm().setVisible(true);: Membuat objek dari kelas MainForm (antarmuka utama program) dan menampilkannya ke layar.
public static void main(String[] args) { ... }: Metode utama yang akan dijalankan saat program dimulai.
*/
import javax.swing.*;

public class TokoElektronik {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new MainForm().setVisible(true);
            }
        });
    }
}
```

Gambar 1. 1 File Utama Program Toko Elektronik

Penjelasan:

1. /* ... */: Ini adalah komentar blok yang digunakan untuk memberikan penjelasan atau dokumentasi terhadap sejumlah baris kode.
2. package elektronik.avenia_rohmatun;; Mendefinisikan package atau paket dari kelas ini, yaitu elektronik.avenia_rohmatun.
3. Author : Avenia Rohmatun: Memberikan informasi tentang penulis atau pembuat program. Berguna untuk dokumentasi dan identifikasi.
4. NIM : 32602200047: Menyertakan NIM (Nomor Induk Mahasiswa) atau identifikasi lainnya. Berguna untuk identifikasi dan dokumentasi pribadi.
5. SwingUtilities.invokeLater(...): Memastikan bahwa pembuatan dan tampilan GUI (Graphical User Interface) dilakukan di EDT (Event Dispatch Thread) untuk menghindari potensi masalah concurrency.
6. new MainForm().setVisible(true);: Membuat objek dari kelas MainForm (antarmuka utama program) dan menampilkannya ke layar.
7. public static void main(String[] args) { ... }: Metode utama yang akan

dijalankan saat program dimulai.

3.2 Kelas Produk

Kelas Produk sebagai kelas dasar yang merepresentasikan produk dengan atribut nama dan harga. Lebih mudahnya adalah file kelas yang menampung kode tentang data informasi yang ditampilkan di program apa saja, baik kita atur (set) atau kita dapatkan (get) dari masing – masing atribut.

```
package elektronik.avenia_rohmatus;

/*
 * Author   : Avenia Rohmatun
 * NIM      : 32602200047
 *
 * Berikut adalah penjelasan terkait getter dan setter pada kelas Produk.
 */

public class Produk {
    // Atribut privat untuk nama dan harga produk
    private String nama;
    private double harga;

    // Konstruktor untuk inisialisasi objek Produk dengan nama dan harga
    public Produk(String nama, double harga) {
        this.nama = nama;
        this.harga = harga;
    }

    // Getter untuk mendapatkan nilai atribut nama
    public String getNama() {
        return nama;
    }

    // Setter untuk mengubah nilai atribut nama
    public void setNama(String nama) {
        this.nama = nama;
    }

    // Getter untuk mendapatkan nilai atribut harga
    public double getHarga() {
```

Gambar 1. 2 Kelas Produk 1

```
    // Getter untuk mendapatkan nilai atribut harga
    public double getHarga() {
        return harga;
    }

    // Setter untuk mengubah nilai atribut harga
    public void setHarga(double harga) {
        this.harga = harga;
    }

    // Metode getInfo untuk mendapatkan informasi lengkap tentang produk
    public String getInfo() {
        return "Produk: " + nama + ", Harga: " + harga;
    }
}
```

Gambar 1. 3 Kelas Produk 2

Penjelasan:

1. `private String nama;` dan `private double harga;`: Mendeklarasikan dua atribut privat (nama dan harga) untuk kelas Produk.
2. `public Produk(String nama, double harga) { ... }`: Konstruktor untuk kelas Produk yang menerima input nama dan harga saat pembuatan objek.
3. `public String getNama() { return nama; }`: Getter untuk mendapatkan nilai atribut nama.
4. `public void setNama(String nama) { this.nama = nama; }`: Setter untuk

mengubah nilai atribut nama.

5. `public double getHarga() { return harga; }`: Getter untuk mendapatkan nilai atribut harga.
6. `public void setHarga(double harga) { this.harga = harga; }`: Setter untuk mengubah nilai atribut harga.
7. `public String getInfo() { return "Produk: " + nama + ", Harga: " + harga; }`: Metode `getInfo` yang memberikan informasi lengkap tentang produk, termasuk nama dan harga.

3.3 Kelas Elektronik

Kelas Elektronik merupakan turunan dari kelas Produk dan mewakili produk elektronik dengan tambahan atribut jenis.

```
package elektronik.avenia_rohmatun;

import java.util.*;

/**
 * Author : Avenia Rohmatun
 * NIM : 32602200047
 *
 * Di kode ini, terdapat implementasi konsep inheritance, polymorphism, dan encapsulation.
 * Berikut adalah penjelasan bagiannya:
 */

public class Elektronik extends Produk { // (1) Inheritance
    private String jenis; // Encapsulation: Atribut jenis dienkapsulasi (private) di dalam kelas Elektronik.

    public Elektronik(String nama, double harga, String jenis) {
        super(nama, harga); // (2) Memanggil konstruktor superclass (Produk) menggunakan super().
        this.jenis = jenis; // Encapsulation: Melakukan inisialisasi atribut jenis di dalam konstruktor.
    }

    // Getter dan setter untuk atribut jenis
    public String getJenis() { // (3) Polymorphism: Menggunakan polimorfisme dengan membuat metode getJenis().
        return jenis;
    }

    public void setJenis(String jenis) { // (3) Polymorphism: Menggunakan polimorfisme dengan membuat metode setJenis().
        this.jenis = jenis;
    }

    @Override
    public String getInfo() { // (3) Polymorphism: Menggunakan polimorfisme dengan melakukan overriding metode getInfo().
        return super.getInfo() + ", Jenis: " + jenis; // (3) Polymorphism: Memanggil metode getInfo() dari superclass (Produk).
    }
}
```

Gambar 1. 4 Kelas Elektronik 1

Penjelasan:

1. Inheritance: Kelas Elektronik meng-extend kelas Produk, sehingga mewarisi atribut dan metode dari kelas Produk.
2. Memanggil konstruktor superclass (Produk) menggunakan `super()`: Dalam konstruktor Elektronik, menggunakan `super(nama, harga)` untuk memanggil konstruktor kelas Produk dan menginisialisasi atribut nama dan harga.
3. Polymorphism: Implementasi polimorfisme dengan membuat metode `getJenis()`, `setJenis()`, dan melakukan overriding pada metode `getInfo()`.

Polimorfisme memungkinkan pemanggilan metode dari superclass (dalam hal ini, Produk) melalui objek turunannya (Elektronik).

3.4 Interface Kategori

Interface Kategori menyediakan kontrak untuk kelas-kelas yang ingin menentukan kategori produk.

```
package elektronik.avenia_rohmatun;

/*
 * Author : Avenia Rohmatun
 * NIM : 32602200047
 *
 * Dalam kode ini, terdapat implementasi konsep Interface.
 * Berikut adalah penjelasan dengan sebuah komentar:
 */

public interface Kategori {
    String getKategori(); // Interface: Mendeklarasikan metode getKategori() tanpa implementasi.
}
```

Penjelasan:

1. Interface: Merupakan sebuah kontrak yang mendeklarasikan metode tanpa memberikan implementasi. Interface digunakan untuk menentukan suatu kontrak yang harus dipenuhi oleh kelas-kelas yang mengimplementasikannya.
2. String getKategori();: Interface Kategori mendeklarasikan metode getKategori() yang harus diimplementasikan oleh kelas-kelas yang menggunakan interface ini. Kelas-kelas yang mengimplementasikan interface Kategori harus menyediakan implementasi konkret untuk metode getKategori().

3.5 Antarmuka utama (MainForm)

Antarmuka utama program yang memungkinkan pengguna untuk memilih operasi yang ingin dilakukan, seperti menambahkan produk elektronik.

```

package elektronik.avenia_rohmatus;

/*
 * Author : Avenia Rohmatun
 * NIM : 32602200047
 *
 * Berikut adalah penjelasan lengkap kode MainForm baris perbaris dengan komentar:
 */

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;

public class MainForm extends JFrame {
    private JButton btnTambahElektronik; // Deklarasi JButton untuk menambah produk elektronik
    private ArrayList<Produk> daftarProduk; // ArrayList untuk menyimpan daftar produk elektronik

    public MainForm() {
        daftarProduk = new ArrayList<>(); // Inisialisasi ArrayList saat objek MainForm dibuat
        initComponents(); // Panggil metode untuk menginisialisasi komponen GUI
    }

    private void initComponents() {
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE); // Menetapkan operasi default ketika jendela ditutup
        setTitle("Sistem Manajemen Toko Elektronik"); // Menetapkan judul jendela

        btnTambahElektronik = new JButton("Tambah Elektronik"); // Membuat tombol "Tambah Elektronik"
        btnTambahElektronik.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                btnTambahElektronikActionPerformed(evt);
            }
        });
    }
}

```

Gambar 1. 5 MainForm 1

```

    }

    private void btnTambahElektronikActionPerformed(ActionEvent evt) {
        // Menambahkan ActionListener untuk menangani aksi tombol "Tambah Elektronik"

        GroupLayout layout = new GroupLayout(this.getContentPane()); // Menggunakan GroupLayout untuk mengatur tata letak
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(GroupLayout.Alignment.LEADING)
                .addGroup(
                    layout.createSequentialGroup()
                        .addGap(30, 30, 30)
                        .addComponent(btnTambahElektronik)
                        .addGap(30, 30, 30)
                )
        ); // Mengatur tata letak horizontal menggunakan GroupLayout

        layout.setVerticalGroup(
            layout.createParallelGroup(GroupLayout.Alignment.LEADING)
                .addGroup(
                    layout.createSequentialGroup()
                        .addGap(30, 30, 30)
                        .addComponent(btnTambahElektronik)
                        .addGap(30, 30, 30)
                )
        ); // Mengatur tata letak vertikal menggunakan GroupLayout

        pack(); // Mengatur ukuran jendela berdasarkan komponen yang ada
    }

    private void btnTambahElektronikActionPerformed(ActionEvent evt) {
        ElektronikForm elektronikForm = new ElektronikForm(this, true, daftarProduk);
        elektronikForm.setVisible(true);
    }

    public void tambahProduk(Produk produk) {
        daftarProduk.add(produk);
    }
}

```

Gambar 1. 6 MainForm 2

```

// Metode yang akan dijalankan saat tombol "Tambah Elektronik" ditekan.
private void btnTambahElektronikActionPerformed(ActionEvent evt) {
    ElektronikForm elektronikForm = new ElektronikForm(this, true, daftarProduk);
    elektronikForm.setVisible(true);
}

//Metode untuk menambahkan produk ke dalam ArrayList
public void tambahProduk(Produk produk) {
    daftarProduk.add(produk);
}
}

```

Gambar 1. 7 MainForm 3

Penjelasan:

1. private JButton btnTambahElektronik;; Deklarasi tombol "Tambah Elektronik".
2. private ArrayList<Produk> daftarProduk;; Deklarasi ArrayList untuk menyimpan daftar produk elektronik.
3. public MainForm() { ... }; Konstruktor untuk kelas MainForm, di mana

ArrayList daftarProduk diinisialisasi.

4. `private void initComponents() { ... }`: Metode untuk menginisialisasi komponen GUI.
5. `setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);`: Menetapkan operasi default ketika jendela ditutup.
6. `setTitle("Sistem Manajemen Toko Elektronik");`: Menetapkan judul jendela.
7. `btnTambahElektronik = new JButton("Tambah Elektronik");`: Membuat tombol "Tambah Elektronik".
8. `btnTambahElektronik.addActionListener(new ActionListener() { ... });`: Menambahkan ActionListener untuk menanggapi aksi tombol "Tambah Elektronik".
9. `GroupLayout layout = new GroupLayout(getContentPane());`: Menggunakan GroupLayout untuk mengatur tata letak komponen.
10. `pack();`: Mengatur ukuran jendela berdasarkan komponen yang ada.
11. `private void btnTambahElektronikActionPerformed(ActionEvent evt) { ... }`: Metode yang akan dijalankan saat tombol "Tambah Elektronik" ditekan.
12. `ElektronikForm elektronikForm = new ElektronikForm(this, true, daftarProduk);`: Membuat objek ElektronikForm dengan meneruskan referensi ke MainForm dan daftarProduk.
13. `elektronikForm.setVisible(true);`: Menampilkan antarmuka form tambah produk elektronik.
14. `public void tambahProduk(Produk produk) { daftarProduk.add(produk); }`: Metode untuk menambahkan produk ke dalam ArrayList daftarProduk.

3.6 Antarmuka tambah produk (ElektronikForm)

Antarmuka untuk menambah produk elektronik dengan input nama, harga, dan jenis produk.

```
package elektronik.avenia_rohmatus;
/*
 * Author : Avenia Rohmatun
 * NIM : 32602200047
 *
 * Berikut adalah penjelasan lengkap kode ElektronikForm baris perbaris dengan komentar:
 */

import javax.swing.*;
import java.util.ArrayList;

public class ElektronikForm extends JDialog {
    private JLabel lblNama, lblHarga, lblJenis; // Deklarasi JLabel untuk nama, harga, dan jenis produk
    private JTextField txtNama, txtHarga, txtJenis; // Deklarasi JTextField untuk input nama, harga, dan jenis produk
    private JButton btnSimpan; // Deklarasi JButton untuk tombol simpan
    private ArrayList<Produk> daftarProduk; // ArrayList untuk menyimpan daftar produk

    // Konstruktor dengan parameter untuk menerima referensi ke MainForm dan daftarProduk
    public ElektronikForm(java.awt.Frame parent, boolean modal, ArrayList<Produk> daftarProduk) {
        super(parent, modal);
        this.daftarProduk = daftarProduk; // Inisialisasi daftarProduk dengan ArrayList yang diterima sebagai parameter
        initComponents(); // Panggil metode untuk menginisialisasi komponen GUI
    }

    private void initComponents() {
        setDefaultCloseOperation(operation: javax.swing.WindowConstants.DISPOSE_ON_CLOSE); // Menetapkan operasi default ketika jendela ditutup
        setTitle(title: "Tambah Produk Elektronik"); // Menetapkan judul jendela

        lblNama = new JLabel(text: "Nama:"); // Membuat label "Nama"
        lblHarga = new JLabel(text: "Harga:"); // Membuat label "Harga"
        lblJenis = new JLabel(text: "Jenis:"); // Membuat label "Jenis"
    }
}
```

Gambar 1. 8 ElektronikForm 1

```
lblJenis = new JLabel(text: "Jenis:"); // Membuat label "Jenis"

txtNama = new JTextField(); // Membuat JTextField untuk input nama
txtHarga = new JTextField(); // Membuat JTextField untuk input harga
txtJenis = new JTextField(); // Membuat JTextField untuk input jenis

btnSimpan = new JButton(text: "Simpan"); // Membuat tombol "Simpan"
btnSimpan.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSimpanActionPerformed(evt);
    }
}); // Menambahkan ActionListener untuk menanggapi aksi tombol "Simpan"

GroupLayout layout = new GroupLayout(this.getContentPane()); // Menggunakan GroupLayout untuk mengatur tata letak
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    group: layout.createParallelGroup(GroupLayout.Alignment.LEADING)
        .addGroup(group: layout.createSequentialGroup()
            .addGap(30, 30, 30)
            .addGroup(group: layout.createParallelGroup(GroupLayout.Alignment.LEADING, sizeable: false)
                .addComponent(component: lblNama)
                .addComponent(component: lblHarga)
                .addComponent(component: lblJenis))
            .addGroup(group: layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(component: txtNama)
                .addComponent(component: txtHarga)
                .addComponent(component: txtJenis))
            .addContainerGap(GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(group: layout.createSequentialGroup()
            .addComponent(component: btnSimpan)
            .addContainerGap())
);

```

Gambar 1. 9 ElektronikForm 2

```

61         .addComponent(component: txtHarga)
62         .addComponent(component: txtJenis)))
63         .addContainerGap(pref: 30, max: Short.MAX_VALUE))
64         .addGroup(alignment: GroupLayout.Alignment.TRAILING, group: layout.createSequentialGroup())
65         .addContainerGap(pref: GroupLayout.DEFAULT_SIZE, max: Short.MAX_VALUE)
66         .addComponent(component: btnSimpan)
67         .addGap(min: 30, pref: 30, max: 30))
68     }; // Mengatur tata letak horizontal menggunakan GroupLayout
69
70     layout.setVerticalGroup(
71         group: layout.createParallelGroup(alignment: GroupLayout.Alignment.LEADING)
72         .addGroup(group: layout.createSequentialGroup())
73         .addGap(min: 30, pref: 30, max: 30)
74         .addComponent(component: lblNama, alignment: GroupLayout.Alignment.BASELINE)
75         .addComponent(component: txtNama, min: GroupLayout.PREFERRED_SIZE, pref: GroupLayout.DEFAULT_SIZE, max: GroupLayout.PREFERRED_SIZE)
76         .addPreferredGap(type: LayoutStyle.ComponentPlacement.UNRELATED)
77         .addGroup(layout.createParallelGroup(alignment: GroupLayout.Alignment.BASELINE)
78         .addComponent(component: lblHarga)
79         .addComponent(component: txtHarga, min: GroupLayout.PREFERRED_SIZE, pref: GroupLayout.DEFAULT_SIZE, max: GroupLayout.PREFERRED_SIZE)
80         .addPreferredGap(type: LayoutStyle.ComponentPlacement.UNRELATED)
81         .addGroup(layout.createParallelGroup(alignment: GroupLayout.Alignment.BASELINE)
82         .addComponent(component: lblJenis)
83         .addComponent(component: txtJenis, min: GroupLayout.PREFERRED_SIZE, pref: GroupLayout.DEFAULT_SIZE, max: GroupLayout.PREFERRED_SIZE)
84         .addPreferredGap(type: LayoutStyle.ComponentPlacement.UNRELATED)
85         .addComponent(component: btnSimpan)
86         .addContainerGap(pref: 30, max: Short.MAX_VALUE))
87     ); // Mengatur tata letak vertikal menggunakan GroupLayout
88
89     pack(); // Mengatur ukuran jendela berdasarkan komponen yang ada
90
91
92

```

Gambar 1. 10 ElektronikForm 3

```

        .addPreferredGap(type: LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(component: btnSimpan)
        .addContainerGap(pref: 30, max: Short.MAX_VALUE))
    }; // Mengatur tata letak vertikal menggunakan GroupLayout
    pack(); // Mengatur ukuran jendela berdasarkan komponen yang ada
}

private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
    String nama = txtNama.getText(); // Mengambil nilai input nama
    double harga = Double.parseDouble(txtHarga.getText()); // Mengambil nilai input harga
    String jenis = txtJenis.getText(); // Mengambil nilai input jenis

    Elektronik elektronik = new Elektronik(nama, harga, jenis); // Membuat objek Elektronik dengan nilai yang diambil
    daftarProduk.add(elektronik); // Menambahkan objek Elektronik ke dalam daftarProduk

    // Membuat StringBuilder untuk menampilkan daftar produk yang telah dipekan
    StringBuilder daftar = new StringBuilder("Daftar Produk yang Telah Dipesan:\n");
    for (Produk produk : daftarProduk) {
        daftar.append("Produk: ").append(produk.getInfo()).append("\n");
    }

    JOptionPane.showMessageDialog(parentComponent, this, message: daftar.toString(), title: "Daftar Produk", messageType: JOptionPane.INFORMATION_MESSAGE);
    this.dispose(); // Menutup form setelah proses selesai
}
}

```

Gambar 1. 11 ElektronikForm 4

Penjelasan:

1. import javax.swing.*;: Mengimpor pustaka Swing untuk komponen GUI.
2. import java.util.ArrayList;: Mengimpor pustaka ArrayList untuk menyimpan daftar produk.
3. public class ElektronikForm extends JDialog {: Mendeklarasikan kelas ElektronikForm yang meng-extend JDialog.
4. private JLabel lblNama, lblHarga, lblJenis;: Deklarasi label (JLabel) untuk nama, harga, dan jenis produk.
5. private JTextField txtNama, txtHarga, txtJenis;: Deklarasi JTextField untuk input nama, harga, dan jenis produk.
6. private JButton btnSimpan;: Deklarasi tombol (JButton) "Simpan".
7. private ArrayList<Produk> daftarProduk;: Deklarasi ArrayList untuk menyimpan daftar produk.
8. public ElektronikForm(java.awt.Frame parent, boolean modal, ArrayList<Produk> daftarProduk) {: Konstruktor ElektronikForm

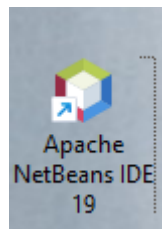
dengan menerima referensi ke MainForm dan daftarProduk sebagai parameter.

9. `super(parent, modal);` Memanggil konstruktor superclass (JDialog) dengan merujuk ke parent (MainForm) dan menyatakan apakah bersifat modal atau tidak.
10. `this.daftarProduk = daftarProduk;` Menginisialisasi daftarProduk dengan ArrayList yang diterima sebagai parameter.
11. `initComponents();` Memanggil metode untuk menginisialisasi komponen GUI.
12. `private void initComponents() {` Metode untuk menginisialisasi komponen GUI.
13. `setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);` Menetapkan operasi default ketika jendela ditutup.
14. `setTitle("Tambah Produk Elektronik");` Menetapkan judul jendela.
15. Inisialisasi komponen GUI:
 - a. `lblNama = new JLabel("Nama:");`
 - b. `lblHarga = new JLabel("Harga:");`
 - c. `lblJenis = new JLabel("Jenis:");`
 - d. `txtNama = new JTextField();`
 - e. `txtHarga = new JTextField();`
 - f. `txtJenis = new JTextField();`
 - g. `btnSimpan = new JButton("Simpan")`

BAB IV IMPLEMENTASI PROGRAM

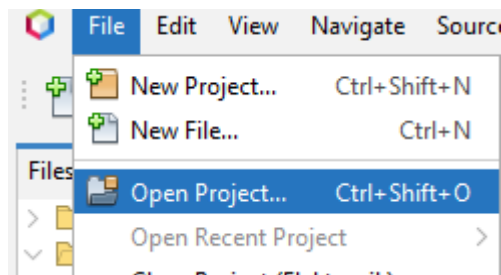
4.1 Menjalankan Program

1. Untuk menjalankan program klik neatbeans

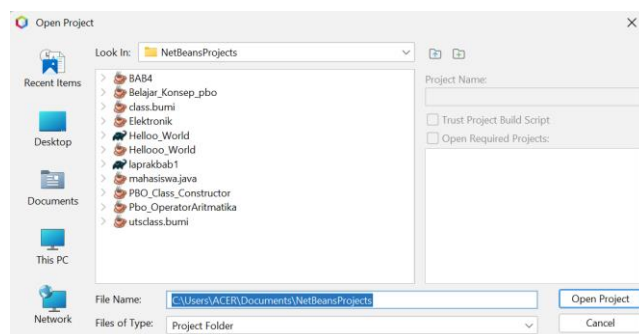


Gambar 2. 1 Open Neatbeans

2. Open new project, arahkan ke elektronik

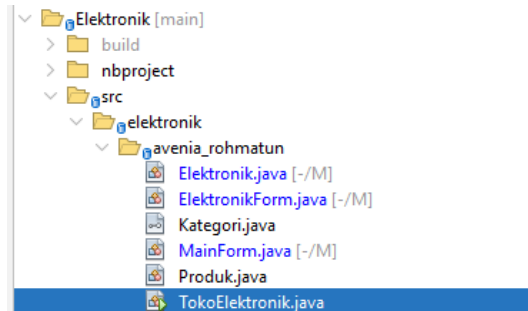


Gambar 2. 2 Open Projects



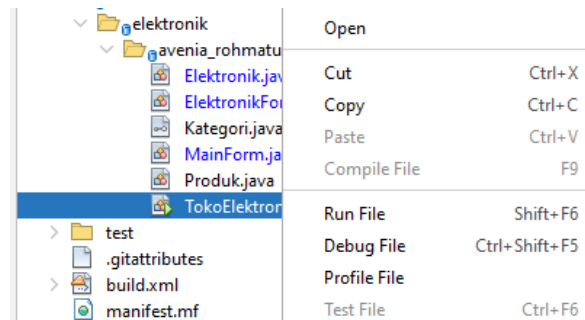
Gambar 2. 3 Arahkan ke project

3. Buka src/elektronik/avenia_rohmatun



Gambar 2. 4 Open struktur project

4. Klik file TokoElektronik.java, lalu klik kanan run file atau shift + f6



Gambar 2. 5 Run program

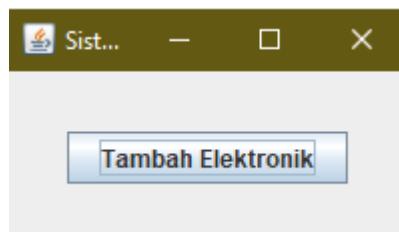
5. Tunggu program muncul



Gambar 2. 6 Tunggu program muncul

4.2 Menambahkan Produk

1. Klik Tombol Tambah Elektronik



Gambar 2. 7 Tambah Elektronik

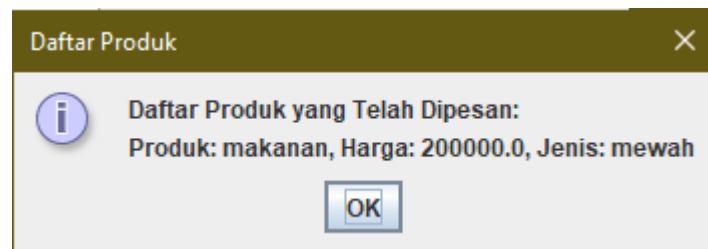
2. Isi Data Nama, Harga, Jenis. Lalu klik Simpan



The screenshot shows a dialog box titled "Tambah Produk Elektronik" with a close button (X) in the top right corner. Inside the dialog, there are three input fields labeled "Nama:", "Harga:", and "Jenis:". Below these fields is a blue button labeled "Simpan".

Gambar 2. 8 Isi Data, Klik Simpan

3. Maka akan muncul sebuah dialog konfirmasi daftar produk yang telah dipesan adalah (apa), harga(berapa), jenis(apa).



The screenshot shows a confirmation dialog box titled "Daftar Produk" with a close button (X) in the top right corner. On the left side, there is an information icon (i). The main text reads "Daftar Produk yang Telah Dipesan:" followed by "Produk: makanan, Harga: 200000.0, Jenis: mewah". At the bottom center, there is a blue button labeled "OK".

Gambar 2. 9 Keluar Dialog Pesanan

BAB V Kesimpulan

Program Java GUI NeatBeans Sistem Manajemen Toko Elektronik ini merupakan implementasi dari beberapa konsep dasar pemrograman berorientasi objek, seperti:

1. Inheritance: Terlihat pada kelas Elektronik yang merupakan turunan dari kelas Produk. Dengan menggunakan inheritance, kelas Elektronik dapat mewarisi properti dan metode dari kelas Produk.
2. Polymorphism: Polymorphism terjadi pada metode getInfo yang di-override di kelas Elektronik. Meskipun metode ini dideklarasikan di kelas induk Produk, namun perilaku (implementasinya) dapat berbeda di kelas anak Elektronik.
3. Encapsulation: Konsep encapsulation diterapkan pada penggunaan akses modifier private pada variabel-variabel di kelas Produk dan Elektronik. Akses ke variabel-variabel ini hanya dapat dilakukan melalui metode getter dan setter.
4. Getter dan Setter: Terdapat metode getter dan setter pada kelas Produk untuk memanipulasi nilai variabel nama dan harga. Penggunaan getter dan setter membantu dalam menjaga kontrol terhadap akses dan manipulasi data.
5. Interface: Konsep interface diimplementasikan pada Kategori yang memiliki satu metode getKategori(). Kelas Elektronik mengimplementasikan interface Kategori, sehingga wajib menyediakan implementasi untuk metode getKategori().

Dalam struktur program ini, terdapat dua frame utama, yaitu MainForm dan ElektronikForm. MainForm berfungsi sebagai tampilan utama yang memungkinkan pengguna untuk menambahkan produk elektronik. Produk yang ditambahkan akan disimpan dalam daftarProduk. ElektronikForm adalah formulir tambah produk elektronik yang memanfaatkan daftarProduk

yang diteruskan dari MainForm. Produk yang ditambahkan melalui formulir ini akan langsung ditampilkan dalam pesan dialog.

Keseluruhan program memberikan gambaran bagaimana konsep-konsep dasar OOP diterapkan dalam pembuatan program dengan antarmuka grafis untuk manajemen toko elektronik.

DAFTAR PUSTAKA

- (Rosenberg and Stephens, 2007; Nasser, Counsell and Shepperd, 2010; Perry, 2010)Nasser, E., Counsell, S. and Shepperd, M. (2010) ‘Class movement and relocation: An empirical study of Java inheritance evolution’, *Journal of Systems and Software*, 83(2), pp. 303–315. Available at: <https://doi.org/10.1016/j.jss.2009.08.011>.
- Perry, J.S. (2010) *Table of content, IEEE Aerospace and Electronic Systems Magazine*. Available at: <https://doi.org/10.1109/maes.2010.5638796>.
- Rosenberg, D. and Stephens, M. (2007) *Use case driven object modeling with UML: Theory and Practice, Use Case Driven Object Modeling with UML: Theory and Practice*. Available at: https://doi.org/10.1007/978-1-4302-0369-8_1.