TUGAS AKHIR

PENDAHULUAN

Deskripsi Umum

Studi kasus pemrograman dalam manajemen penyewaan alat outdoor adalah analisis mendalam tentang bagaimana teknologi dan pemrograman komputer digunakan untuk meningkatkan efisiensi, dan pengelolaan operasional dalam penyewaan alat outdoor. Dalam studi kasus ini, akan mempelajari berbagai aspek, seperti mengelola data produk, penyewaan secara online, dan integrasi teknologi lainnya.

Konsep Object Oriented Programming yang Digunakan

Object-Oriented Programming (OOP) atau diartikan dalam bahasa indonesia yaitu Pemrograman Berorientasi Objek adalah paradigma pemrograman tingkat lanjut setelah paradigma prosedural yang berfokus pada permodelan dunia nyata ke dalam bentuk objek-objek yang memiliki atribut (data) dan method (fungsi) yang berkaitan satu sama lain. OOP dapat membantu dalam mengorganisasi dan mengelola kode dengan lebih terstruktur dan mudah dimengerti.

TUGAS AKHIR

ANALISIS PERSIAPAN

Topik Bahasan

1. Modul 1: Pengenalan Java OOP
2. Modul 2: Class dan Bagiannya
3. Modul 3: Constructor dan Object

Tujuan Pertemuan

Tujuan dari Pertemuan 1 (satu) ini bertujuan untuk memberikan pemahaman dasar tentang pemrograman berorientasi objek dengan bahasa pemrograman Java, mencakup konsep dasar OOP, pembuatan Class dan bagiannya, serta penggunakan Constructor dan Cbject dalam pengembangan perangkat lunak.

Isi Pertemuan

```
public class Produk{
    private String nama;
    private String merek;
    private String jenis;
    private double harga;
    private String deskripsi;
    private int stok;

    public Produk (String nama, String merek, String jenis,
double harga, String deskripsi, int stok) {
        this.nama = nama;
        this.merek = merek;
        this.jenis = jenis;
        this.harga = harga;
        this.deskripsi = deskripsi;
        this.stok = stok;
    }
```

Penjelasan

Kode di atas adalah implementasi dari sebuah kelas Java yang bervariabel "Produk". Kelas ini memiliki beberapa atribut, yaitu "nama" (untuk menyimpan nama produk), "merek" (untuk menyimpan merek produk), "jenis" (untuk menyimpan jenis produk), "harga" (untuk menyimpan harga produk), "stok" (untuk menyimpan stok ketersediaan produk). Selain itu, kelas ini memiliki sebuah konstruktor yang digunakan untuk membuat objek Produk baru dengan menginisialisasi atribut-atributnya berdasarkan nilai-nilai yang diberikan saat objek dibuat. Dengan demikian, kelas ini digunakan untuk mempresentasikan informasi tentang Produk dalam suatu sistem program.

```
public class CustomerModel {
    private String nomorHp;
    private String alamat;

    public CustomerModel(String nomorHp, String alamat) {
        this.nomorHp = nomorHp;
        this.alamat = alamat;
    }
```

Penjelasan

Pada kode di atas merupakan implementasi dari sebuah kelas Java yang bervariabel "CustomerModel". Kelas ini memiliki beberapa atribut, yaitu "nomorHp" (untuk menyimpan nomor hp customer), "alamat" (untuk menyimpan alamat customer). Kelas ini juga memiliki sebuah konstruktor yang digunakan untuk membuat objek CustomerModel baru dengan menginisialisasi atribut-atributnya berdasarkan nilai-nilai yang diberikan saat objek dibuat.

TUGAS AKHIR

IMPLEMENTASI

Topik Bahasan

1. Modul 4: Input Process Output
2. Modul 5: Data Collection

Tujuan Pertemuan

Modul 4: “Input Process Output” pada bahasa pemrograman Java input merupakan suatu proses memasukkan data ke dalam sebuah program komputer baik dilakukan secara statis maupun dinamis (input dari user). Pada process berisi sebuah logika-logika dari program yang kita buat, yaitu seperti pengujian, dan perulangan. Dan untuk membuat sebuah output program dapat menggunakan perintah “System.out” yang biasanya oran-orang menyebutnya dengan “sout” atau “sysout”. Sementara itu, Modul 5: “Data Collection” merujuk pada struktur data yang digunakan untuk mengelompokkan dan menyimpan sejumlah elemen atau objek dalam satu kesatuan. Data Collection memungkinkan kita untuk mengelola, mengakses, dan memanipulasi sejumlah besar data dengan lebih terstruktur dan efisien.

Korelasi antara kedua modul ini terletak pada fakta bahwa “Input Process Input” memberikan dasar untuk memahami bagaimana data diproses dalam bahasa pemrograman Java, sementara “Data Collection” melengkapi pemahaman ini dengan mengajarkan cara mendapatkan data yang dibutuhkan dalam proses tersebut.

Hasil Implementasi

```
public class AdminController {
    private List<Produk> produks;
    private List<Penyewaan> rentals;
```

```
public AdminController() {
    produks = new ArrayList<>();
    rentals = new ArrayList<>();
}
```

Penjelasan

Pada kode program ini memiliki kelas berupa ArrayList yang menyimpan objek dari kelas terkait dengan operasional Admin, yaitu:

1. ‘List<Produk> produks’ : sebuah list yang akan menyimpan objek-objek dari kelas Produk. List ini berfungsi untuk menampung data produk yang ada.
2. ‘produks’ : sebuah objek ArrayList kosong dan menginisialisasi variabel produks untuk menampung data produk.

TUGAS AKHIR

Topik Bahasan

1. Modul 6: Encapsulation
2. Modul 7: Inheritance

Tujuan Pertemuan

Modul 6: “Encapsulation” pada bahasa pemrograman Java adalah untuk memahami konsep dasar pemaketan data dan metode kelas dalam kelas, sehingga dapat menjaga keamanan dan integritas data, meningkatkan modularitas, memudahkan keterbacaan kode serta mengendalikan akses terhadap data dan memastikan data hanya bisa dimodifikasi dengan cara yang valid. Sementara itu, tujuan dari Modul 7: “Inheritance” adalah untuk memungkinkan sebuah kelas turunan (subclass) mewarisi atribut dan metode dari kelas lain/induk (superclass).

Korelasi antar kedua modul ini terletak pada fakta bahwa encapsulation membantu melindungi data dalam kelas, sementara inheritance memungkinkan pembuatan hierarki kelas yang dapat memanfaatkan sifat dari metode yang sudah ada, menciptakan hubungan yang kuat antara objek dan mengurangi duplikasi kode dalam pemrograman Java. Dengan encapsulation, meskipun data diwariskan melalui inheritance, akses tetap terkontrol dan aman, keduanya memainkan peran penting dalam pembangunan aplikasi yang efesien dan modular.

Hasil Implementasi

```
public class CustomerModel extends User {
    private String nomorHp;
    private String alamat;

    public CustomerModel(String username, String password,
String nomorHp, String alamat) {
        super(username, password);
        this.nomorHp = nomorHp;
        this.alamat = alamat;
```

```
    }
    public String getNomorHp() {
        return nomorHp;
    }
    public void setNomorHp(String nomorHp) {
        this.nomorHp = nomorHp;
    }
    public String getAlamat() {
        return alamat;
    }
    public void setAlamat(String alamat) {
        this.alamat = alamat;
    }
```

Penjelasan

Pada kelas “CustomerModel” merupakan turunan dari kelas “User”. Dalam konteks pemarisan ini, kelaas “CustomerModel” adalah kelas turunan (subclass) dari kelas induk “User” (superclass), ini berarti bahwa kelas “CustomerModel” mewarisi atribut-atribut dan metode-metode yang ada dalam kelas “User”. Selain itu, kelas “CustomerModel” memiliki atribut tambahan berupa “nomorHp” dan “alamat” yang berkaitan dengan informasi tambahan tentang tamu yang tidak ada dalam kelas “User”.

Poin pentingnya adalah pewarisan ditunjukkan melalui kata kunci extends super() dalam konstruktor memastikan kelas induk diinisialisasi dengan benar Encapsulation diimplementasikan melalui access modifier private. Struktur ini memungkinkan untuk membuat objek khusus dengan atribut tambahan diluar “username” dan “password”.

TUGAS AKHIR

Topik Bahasan

1. Modul 8: Relasi Antar Class
2. Modul 9: Polymorphism

Tujuan Pertemuan

Tujuan dari Modul 8: “Relasi Antar Class” berfokus pada penyembunyian suatu kelas dan membatasi akses langsung ke data melalui modifier private. Sementara itu, Modul 9: “Polymorphism” memungkinkan objek untuk bereaksi secara berbeda terhadap method yang sama, memberikan fleksibilitas dalam desain sistem.

Korelasi antara kedua modul ini terletak pada fakta bahwa dengan menggabungkan kedua konsep ini, pengembang dapat membuat sistem yang tidak hanya aman dan terlindungi, tetapi juga sangat adaptif terhadap perubahan dan pengembangan lebih lanjut. Keduanya dapat memainkan peran penting dalam kemampuan untuk membuat kode yang lebih abstrak dan modular.

Hasil Implementasi

```
public class Penyewaan {
    private CustomerModel customer;
    private Keranjang keranjang;
    private LocalDate tanggalSewa;
    private LocalDate tanggalPengembalian;
    private String metodePengiriman;
    private String metodePembayaran;
    private String status;
    private double denda;

    public Penyewaan(Customer customer, Keranjang keranjang,
LocalDate tanggalSewa, LocalDate tanggalPengembalian,
                     String metodePengiriman, String
metodePembayaran) {
```

```
        this.customer = customer;
        this.keranjang = keranjang;
        this.tanggalSewa = tanggalSewa;
        this.tanggalPengembalian = tanggalPengembalian;
        this.metodePengiriman = metodePengiriman;
        this.metodePembayaran = metodePembayaran;
        this.status = "Diantar";
        this.denda = hitungDenda(tanggalPengembalian,
tanggalSewa);
    }

    public String getCustomer() {
        return customer;
    }
    public double hitungDenda(LocalDate tanggalPengembalian,
LocalDate tanggalSewa) {
        if (tanggalPengembalian.isAfter(tanggalSewa)) {
            long terlambat =
ChronoUnit.DAYS.between(tanggalSewa, tanggalPengembalian);
            return terlambat * 10000; // Rp 10.000 per hari
        }
        return 0;
    }
    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append("Penyewaan oleh:
").append(customer).append("\n");
     sb.append("Keranjang: ").append(keranjang).append("\n");
        sb.append("Tanggal Sewa:
").append(tanggalSewa).append("\n");
        sb.append("Tanggal Pengembalian:
").append(tanggalPengembalian).append("\n");
        sb.append("Status: ").append(status).append("\n");
        sb.append("Total Biaya:
").append(totalCost).append("\n");
        sb.append("Daftar Item:\n");
        for (Produk keranjang : keranjang) {
            sb.append("-
").append(keranjang.getCustomer()).append("\n");
```

```
        }
        return sb.toString();
    }
```

Penjelasan

Kelas “Penyewaan” memiliki dua atribut yaitu “customer” dan “keranjang” yang masing-masing merupakan objek dari kelas “CustomerModel” dan “Keranjang”, yang berarti kelas “Penyewaan” memiliki refrensi ke objek-objek dari kelas lain untuk mempresentasikan data customer dan keranjang yang terkait dengan penyewaan.

Selain itu. kelas ini juga memiliki metode Override “toString” mengoverride metode bawaan dari kelas “Object” untuk memberikan representasi string yang mudah dibaca, dan berisi informasi informasi lengkap tentang penyewaan. Kesuluruhan desain ini mendemonstrasikan fleksibilitas dan hubungan yang kuat antara kelas-kelas dalam suatu sistem.

TUGAS AKHIR

Topik Bahasan

1. Modul 10: Package
2. Modul 11: Abstraction dan Interfaces

Tujuan Pertemuan

Tujuan yang dapat dicapai dari Modul 10: “Package” adalah untuk mengelompokkan kelas-kelas yang memiliki fungsi atau tujuan serupa ke dalam paket, yang juga membantu juga dalam pemeliharaan dan manajemen yang lebih baik. Sedangkan dari Modul 11: “Abstraction dan Interfaces”, tujuannya untuk memungkinkan fleksibilitas dan modularitas dalam sistem, karena berbagai kelas yang berbeda dapat berinteraksi melalui antarmuka (interfaces) atau abstraksi (abstraction) tanpa bergantung pada implementasi spesifik, yang juga dapat diorganisasi secara lebih efisien melalui pengguna paket.

Korelasi antara kedua modul ini terletak pada integrasi konsep paket dengan penggunaan antarmuka dan abstraksi, membantu menciptakan struktur kode yang lebih terstruktur, mudah dipelihara, dan sesuai prinsip OOP.

Hasil Implementasi

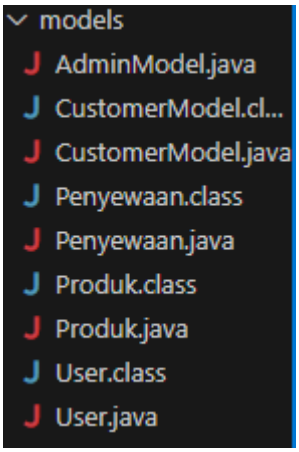
```
package models;

public abstract class User {
    private String username;
    private String password;

    public User(String username, String password) {
        this.username = username;
        this.password = password;
    }
```

TUGAS AKHIR

Penjelasan



Didalam terdapat banyak kelas-kelas, yaitu ada "AdminModel.java", "CustomerModel.java", "Penyewaan.java", "Produk.java" dan "User.java".

Kode program ini merupakan implementasi konsep abstraksi dalam Java melalui penggunaan kelas abstrak. Kelas "User" dideklarasikan sebagai abstrak dengan tujuan menjadi kerangka dasar untuk kelas-kelas turunan, seperti kelas "AdminModel" dan "CustomerModel", yang akan menambahkan atribut spesifik. Kelas ini memiliki dua atribut yaitu "username" dan "password", yang diatur melalui konstruktor untuk memastikan setiap objek kelas "User" memiliki data yang valid sejak awal. Atribut-atribut tersebut bersifat private, yang memastikan data hanya dapat diakses dan dimodifikasi secara tidak langsung melalui kelas itu sendiri. Dengan pendekatan ini, kode menjadi lebih modular dan teroganisir.

TUGAS AKHIR

Topik Bahasan

1. Modul 12: Exception Handling
2. Modul 13: Pemrograman Multimedia

Tujuan Pertemuan

Tujuan yang dapat dicapai dari Modul 12: “Exceptiong Handling” untuk mengelola dan menangani kondisi kesalahan (error) atau kejadian tak terduga yang dapat terjadi selama eksekusi program tanpa menyebabkan program berhenti secara mendadak, sehingga dapat menciptakan aplikasi yang lebih handal dan dapat diandalkan. Sementara itu, Modul 13: “Pemrograman Multimedia” bertujuan untuk mengintegrasikan berbagai elemen media, seperti gambar, suara, video, animasi, dan teks, dalam satu aplikasi yang interaktif dan menarik.

Korelasi kedua modul ini terletak dalam pengembangan perangkat lunak, dimana exception handling memastikan aplikasi berjalan dengan lancar tanpa ganguan, sementara multimedia memperkaya antarmuka dan interaksi melalui integrasi berbagai elemen media.

Hasil Implementasi

```
import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

public class SimpleMultimediaExceptionHandling {

    public static void main(String[] args) {
        // Path ke gambar yang ingin dimuat
        String imagePath = "path/to/your/image.jpg"; // Ganti
dengan path file gambar yang valid
```

```
        // Coba memuat gambar dan menangani kemungkinan
kesalahan
        try {
            // Memuat gambar
            BufferedImage image = loadImage(imagePath);
            System.out.println("Gambar berhasil dimuat!");
            System.out.println("Lebar gambar: " +
image.getWidth() + "px");
            System.out.println("Tinggi gambar: " +
image.getHeight() + "px");
        } catch (IOException e) {
            // Menangani kesalahan jika file tidak ditemukan
atau format gambar tidak didukung
            System.err.println("Terjadi kesalahan saat memuat
gambar: " + e.getMessage());
        }
    }

    // Fungsi untuk memuat gambar dari file
    public static BufferedImage loadImage(String path) throws
IOException {
        File file = new File(path);
        if (!file.exists()) {
            throw new IOException("File gambar tidak ditemukan
di path: " + path);
        }
        return ImageIO.read(file);
    }
}
```

Penjelasan

Pada program ini diatur dengan path menuju file gambar yang dimuat, kemudian mencoba memuat gambar tersebut dengan menggunakan fungsi “loadImage()”. Fungsi ini membaca file gambar dari path yang diberikan menggunakan “ImageIO.read()”. Jika file tidak ditemukan atau ada masalah saat membaca gambar, maka fungsi ini akan melemparkan “IOException”, yang akan ditangkap oleh blok “catch” untuk menangani kesalahan tersebut. Jika gambar berhasil dimuat, program akan menampilkan pesan “Gambar berhasil dimuat!”.

Jika terjadi kesalahan, pesan kesalahan yang relevan akan ditampilkan tentang masalah yang terjadi dengan pesan “Terjadi kesalahan saat memuat gambar: Tidak dapat membaca file gambar”.

Dengan demikian, program ini menggabungkan exception handling dengan pemrograman multimedia untuk memastikan aplikasi tetap berjalan dengan lancar dalam membaca dan mengelola file gambar.

TUGAS AKHIR

Topik Bahasan

1. Modul 14: Graphic User Interface (GUI)

Tujuan Pertemuan

Tujuan yang dapat dicapai dalam modul ini untuk membangun antarmuak pengguna dalam aplikasi Java. Swing dirancang untuk memberikan tampilan dan nuansa yang lebih konsisten dan lebih kaya. Dengan menguasai Java GUI, pengembang dapat menciptakan aplikasi yang lebih menarik dan mudah digunakan, yang merupakan kemampuan penting dalam pengembangan perangkat lunak modern.

Hasil Implementasi

```
package views;

import controllers.*;
import java.awt.*;
import javax.swing.*;

public class CustomerView {
    private CustomerController customerController;

    public CustomerView(CustomerController customerController) {
        this.customerController = customerController;
        createGUI();
    }

    private void createGUI() {
        JFrame frame = new JFrame("Customer Login");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 300);

        JPanel panel = new JPanel();
```

```
        panel.setLayout(new GridLayout(3, 2));

        JLabel usernameLabel = new JLabel("Username:");
        JTextField usernameField = new JTextField();

        JLabel passwordLabel = new JLabel("Password:");
        JPasswordField passwordField = new JPasswordField();

        JButton loginButton = new JButton("Login");
        JButton registerButton = new JButton("Register");

        loginButton.addActionListener(e -> {
            String username = usernameField.getText();
            String password = new
String(passwordField.getPassword());

            if (customerController.loginCustomer(username,
password)) {
                JOptionPane.showMessageDialog(frame, "Login
Successful! Welcome, " + username + "!");
            } else {
                JOptionPane.showMessageDialog(frame, "Login
Failed! Silahkan Coba Lagi.", "Error",
JOptionPane.ERROR_MESSAGE);
            }
        });

        registerButton.addActionListener(e -> {
            String username =
JOptionPane.showInputDialog("Masukkan Username:");
            String password =
JOptionPane.showInputDialog("Masukkan Password:");
            String nomorHp =
JOptionPane.showInputDialog("Masukkan Nomor Hp:");
            String alamat =
JOptionPane.showInputDialog("Masukkan Alamat:");

            customerController.registerCustomer(username,
password, nomorHp, alamat);
```

```
            JOptionPane.showMessageDialog(frame, "Registration
Successful! Silahkan Login.");
        });

        panel.add(usernameLabel);
        panel.add(usernameField);
        panel.add(passwordLabel);
        panel.add(passwordField);
        panel.add(loginButton);
        panel.add(registerButton);

        frame.add(panel);
        frame.setVisible(true);
    }
}
```

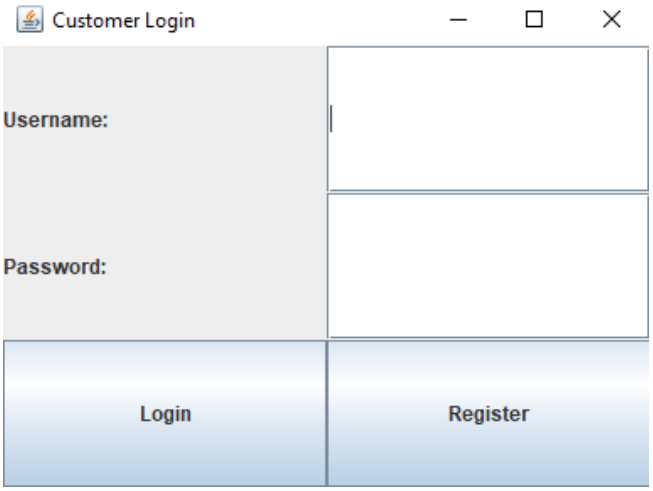
Penjelasan

Kode program di atas merupakan implementasi antarmuka pengguna berbasis Graphic User Interfaces (GUI) menggunakan pustaka Swing di bahasa pemrograman Java. Kelas "CustomerView" bertanggung jawab untuk membuat dan mengelola antarmuka pengguna, dalam metode "createGUI()", elemen-elemen antarmuka seperti label, input teks, dan tombol diatur menggunakan layout grid agar tampil teroganisir. Tombol "Login" memicu proses auntentikasi dengan memvalidasi username dan password melalui metode "loginCustomer()". Jika login berhasil, pesan sukses ditampilkan, sedangkan jika gagal login akan muncul pesan kesalahan. Tombol "Register" memungkinkan pengguna untuk memasukkan data akun baru, yang kemudian diteruskan ke metode "registerCustomer()" untuk menyimpan data tersebut. Program ini memanfaatkan dialog input dan pesan pop-up untuk memberikan notifikasi kepada pengguna, membuat interaksi lebih lebih intuitif dan interaktif.

TUGAS AKHIR

HASIL DAN DOKUMENTASI

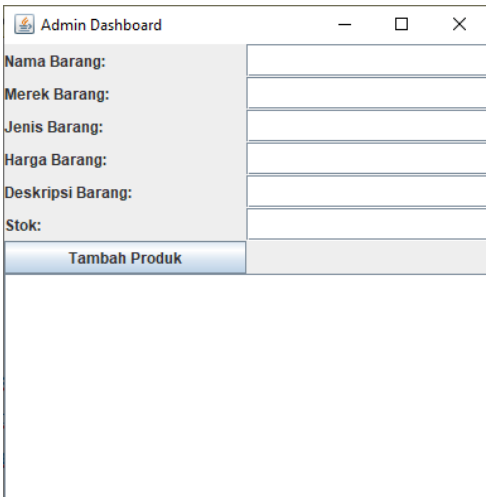
Tampilan Login



Pada halaman login terdapat judul bernama “Customer Login” pada bagian atas, kemudian untuk melalukan proses autentikasi maka pengguna harus memasukkan data akun berupa Username dan Password, sebelum melalukan proses Login, pengguna harus melakukan Register terlebih dahulu untuk membuat akun baru.

TUGAS AKHIR

Tampilan Dashboard Aplikasi



Pada halaman dashboard terdapat judul bernama “Admin Dashboard” pada bagian atas. Antarmuka memiliki label seperti “Nama Barang”, “Merek Barang”, “Jenis Barang”, “Deskripsi Barang”, dan “Stok”, yang masing-masing menunjukkan data spesifik produk yang perlu diinput oleh admin, kemudian untuk melakukan proses autentikasi maka pengguna harus memasukkan data berupa nama, merek, jenis, deskripsi, dan jumlah stok. Bagian bawah layar dilengkapi dengan tombol “Tambah Produk”, yang berfungsi untuk memproses dan menyimpan data produk ke dalam sistem setelah semua data diisi.