

LAPORAN PRAKTIKUM PEMOGRAMAN BERORIENTASI OBJEK CLASS, OBJECT, CONSTRUCTOR, DAN METHOD



Oleh

Adila Bintang

2411533019

Dosen Pengampu: Nurfiah S.ST, M.Kom

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

A. Tujuan Praktikum

1. Mahasiswa dapat membuat class yang berisi beberapa atribut.
2. Menerapkan konsep encapsulation menggunakan getter dan setter

B. Pendahuluan

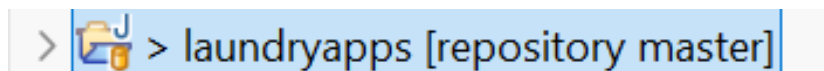
Pemrograman Berorientasi Objek (PBO) merupakan paradigma pemrograman yang berfokus pada pembuatan class dan object untuk merepresentasikan entitas dunia nyata ke dalam program komputer. Dalam Java, konsep PBO sangat penting karena membantu pengembang membuat program yang lebih terstruktur, mudah dikelola, serta dapat digunakan kembali (reusability).

Pada praktikum ini, dipelajari bagaimana membuat class, object, serta menerapkan encapsulation dengan menggunakan getter dan setter. Studi kasus yang digunakan adalah aplikasi laundry sederhana, di mana dibutuhkan beberapa class, salah satunya adalah class Customer untuk merepresentasikan data pelanggan.

Penerapan encapsulation pada class ini bertujuan agar atribut (id, nama, alamat, noHp) tidak dapat diakses langsung dari luar class, tetapi harus melalui method khusus (get dan set). Hal ini menjaga keamanan data serta menjaga prinsip modularitas dalam pemrograman.

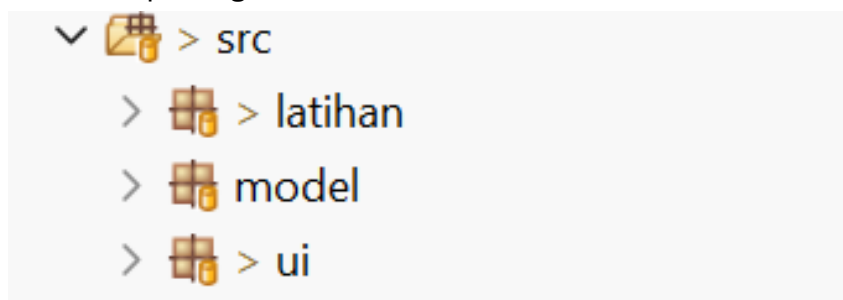
C. Langkah-Langkah

1. Membuat project dengan nama laundryapps



Langkah pertama dilakukan dengan **dibuatnya sebuah project baru** bernama *laundryapps*. Pembuatan project ini dimaksudkan agar semua file, package, dan class dapat ditempatkan dalam satu ruang kerja yang terorganisir sehingga proses pengembangan aplikasi dapat lebih mudah dilakukan.

2. Membuat package



Setelah project selesai dibuat, **sebuah package dengan nama *model* dibentuk**. Package ini digunakan sebagai wadah pengelompokan class sehingga kode

dapat disusun secara lebih terstruktur, modular, serta lebih mudah dipelihara. Dengan adanya package, pemisahan antara logika program dan antarmuka pengguna dapat diterapkan sesuai prinsip pemrograman berorientasi objek.

Package model

Class user

```
1 package model;
2
3 public class User {
4     String id, nama, username, password;
5
6     public String getId() {
7         return id;
8     }
9
10    public void setId(String id) {
11        this.id = id;
12    }
13
14    public String getNama() {
15        return nama;
16    }
17
18    public void setNama(String nama) {
19        this.nama = nama;
20    }
21
22    public String getUsername() {
23        return username;
24    }
25
26    public void setUsername(String username) {
27        this.username = username;
28    }
29
30    public String getPassword() {
31        return password;
32    }
33
34    public void setPassword(String password) {
35        this.password = password;
36    }
37
38    public static boolean login(String username, String password) {
39        boolean isLogin = false;
40
41        // Dummy user (bisa diganti dengan query ke database nantinya)
42        User user = new User();
43        user.setId("1");
44        user.setNama("Fulan");
45        user.setUsername("fulan");
46        user.setPassword("12345");
47
48        if (user.getUsername().equalsIgnoreCase(username)
49            && user.getPassword().equals(password)) {
50            isLogin = true;
51        }
52
53        return isLogin;
54    }
55 }
```

Kemudian, **class User didefinisikan**. Class ini dimaksudkan untuk merepresentasikan data pengguna atau pelanggan laundry. Atribut yang digunakan, seperti id, nama, alamat, dan noHp, **dibuat dengan modifier private** sehingga tidak dapat diakses secara langsung dari luar class. Untuk mengakses maupun mengubah nilai atribut tersebut, **method getter dan setter dipanggil**. Melalui cara ini, prinsip **encapsulation** berhasil diterapkan sehingga keamanan data dapat terjaga.

Login frame

```
Costumer.java Order.java Service.java User.java
1 package ui;
2
3
4 import java.awt.EventQueue;
5
6 import javax.swing.JFrame;
7 import javax.swing.JPanel;
8 import javax.swing.border.EmptyBorder;
9
10 import model.User;
11 import ui.MainFrame;
12
13 import javax.swing.JLabel;
14 import javax.swing.JOptionPane;
15
16 import java.awt.Font;
17
18 import javax.swing.JTextField;
19 import javax.swing.JButton;
20 import java.awt.event.ActionListener;
21 import java.awt.event.ActionEvent;
22
23 public class LoginFrame extends JFrame {
24
25     private static final long serialVersionUID = 1L;
26     private JPanel contentPane;
27     private JTextField txtUsername;
28     private JTextField txtPassword;
29     private JLabel lblNewLabel_2;
30     private JLabel lblNewLabel_3;
31
32     /**
33      * Launch the application
34      */
35
36 Costumer.java Order.java Service.java User.java
1 package ui;
2
3
4 import java.awt.EventQueue;
5
6 import javax.swing.JFrame;
7 import javax.swing.JPanel;
8 import javax.swing.border.EmptyBorder;
9
10 import model.User;
11 import ui.MainFrame;
12
13 import javax.swing.JLabel;
14 import javax.swing.JOptionPane;
15
16 import java.awt.Font;
17
18 import javax.swing.JTextField;
19 import javax.swing.JButton;
20 import java.awt.event.ActionListener;
21 import java.awt.event.ActionEvent;
22
23 public class LoginFrame extends JFrame {
24
25     private static final long serialVersionUID = 1L;
26     private JPanel contentPane;
27     private JTextField txtUsername;
28     private JTextField txtPassword;
29     private JLabel lblNewLabel_2;
30     private JLabel lblNewLabel_3;
31
32     /**
33      * Launch the application
34      */
35
36 MainFrame.java LoginFrame
60 JLabel lblNewLabel = new JLabel("Laundry Apps");
61 lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 25));
62 lblNewLabel.setBounds(88, 0, 205, 66);
63 contentPane.add(lblNewLabel);
64
65 JLabel lblNewLabel_1 = new JLabel("males aja nyuci? biar kami cuciin");
66 lblNewLabel_1.setEnabled(false);
67 lblNewLabel_1.setFont(new Font("Times New Roman", Font.PLAIN, 15));
68 lblNewLabel_1.setBounds(88, 54, 244, 26);
69 contentPane.add(lblNewLabel_1);
70
71 txtUsername = new JTextField();
72 txtUsername.setBounds(88, 129, 280, 26);
73 contentPane.add(txtUsername);
74 txtUsername.setColumns(10);
75
76 txtPassword = new JTextField();
77 txtPassword.setColumns(10);
78 txtPassword.setBounds(88, 180, 280, 26);
79 contentPane.add(txtPassword);
80
81 lblNewLabel_2 = new JLabel("Username");
82 lblNewLabel_2.setFont(new Font("Times New Roman", Font.PLAIN, 12));
83 lblNewLabel_2.setBounds(88, 105, 69, 14);
84 contentPane.add(lblNewLabel_2);
85
86 lblNewLabel_3 = new JLabel("Password");
87 lblNewLabel_3.setFont(new Font("Times New Roman", Font.PLAIN, 13));
88 lblNewLabel_3.setBounds(88, 160, 69, 14);
89 contentPane.add(lblNewLabel_3);
90
91 JButton btnLogin = new JButton("Login");
92 btnLogin.addActionListener(new ActionListener() {
93     public void actionPerformed(ActionEvent e) {
94         if (User.Login(txtUsername.getText(), txtPassword.getText(),
95             new MainFrame().setVisible(true);
96             dispose();
97         } else {
98             JOptionPane.showMessageDialog(null, "Login Gagal");
99         }
100     }
101 });
102 btnLogin.setBounds(171, 229, 110, 36);
103 contentPane.add(btnLogin);
104 }
```

Laundry Apps

males aja nyuci? biar kami cuciin

Username

Password

Login

Component	Id	Keterangan
TextField	txtUsername	Username
TextField	txtPassword	Password
Button	btnLogin	Login

Tahap berikutnya, **Login Frame dirancang**. Komponen yang dipakai meliputi JTextField txtUsername untuk username, JTextField txtPassword untuk password, serta JButton btnLogin sebagai tombol masuk. Login Frame ini digunakan sebagai pintu awal sehingga hanya pengguna yang sesuai yang dapat mengakses aplikasi. Dengan perancangan ini, interaksi antara class, object, dan antarmuka pengguna dapat ditunjukkan.

Main frame

```
IDE
ct Run Window Help
1 package ui;
2
3 import javax.swing.*;
4 import java.awt.Panel;
5 import java.awt.BorderLayout;
6 import java.awt.Font;
7 import java.awt.Color;
8
9 public class MainFrame extends JFrame {
10
11     public MainFrame() {
12         // Judul frame
13         setTitle("Halaman Utama");
14         setSize(400, 300);
15         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
16         setLocationRelativeTo(null);
17         getContentPane().setLayout(null);
18
19         JButton btnNewButton = new JButton("PESANAN");
20         btnNewButton.setFont(new Font("Times New Roman", Font.PLAIN, 13));
21         btnNewButton.setBounds(10, 91, 110, 43);
22         getContentPane().add(btnNewButton);
23
24         JButton btnNewButton_1 = new JButton("LAYANAN");
25         btnNewButton_1.setFont(new Font("Times New Roman", Font.PLAIN, 13));
26         btnNewButton_1.setBounds(141, 91, 110, 43);
27         getContentPane().add(btnNewButton_1);
28
29         JButton btnNewButton_2 = new JButton("PELANGGAN");
30         btnNewButton_2.setFont(new Font("Times New Roman", Font.PLAIN, 13));
31         btnNewButton_2.setBounds(261, 91, 115, 43);
32         getContentPane().add(btnNewButton_2);
33
34         JButton btnNewButton_3 = new JButton("PENGGUNA");
35         btnNewButton_3.setFont(new Font("Times New Roman", Font.PLAIN, 13));
36         btnNewButton_3.setBounds(10, 145, 110, 43);
37         getContentPane().add(btnNewButton_3);
38
39         JButton btnNewButton_4 = new JButton("LAPORAN");
40         btnNewButton_4.setFont(new Font("Times New Roman", Font.PLAIN, 13));
41         btnNewButton_4.setBounds(141, 145, 110, 43);
42         getContentPane().add(btnNewButton_4);
43
44         JButton btnNewButton_5 = new JButton("PROFILE");
45         btnNewButton_5.setFont(new Font("Times New Roman", Font.PLAIN, 13));
46         btnNewButton_5.setBounds(261, 145, 115, 43);
47         getContentPane().add(btnNewButton_5);
48
49         JButton btnNewButton_6 = new JButton("KELUAR");
50         btnNewButton_6.setFont(new Font("Times New Roman", Font.PLAIN, 13));
51         btnNewButton_6.setBounds(10, 205, 110, 47);
52         getContentPane().add(btnNewButton_6);
53
54         JLabel lblNewLabel = new JLabel("Laundry Apps");
55         lblNewLabel.setForeground(new Color(0, 64, 128));
56         lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 20));
57         lblNewLabel.setBounds(20, 29, 129, 36);
58         getContentPane().add(lblNewLabel);
59     }
60
61     // Untuk tes langsung MainFrame
62     public static void main(String[] args) {
63         SwingUtilities.invokeLater(() -> {
64             new MainFrame().setVisible(true);
65         });
66     }
67 }
```



Setelah login berhasil dilakukan, **Main Frame ditampilkan**. Frame ini dijadikan sebagai halaman utama aplikasi, tempat di mana fitur-fitur utama dapat diakses, misalnya pengelolaan pelanggan atau transaksi laundry. Main Frame berfungsi sebagai pusat navigasi aplikasi sehingga integrasi antarbagian program dapat diperlihatkan.

Latihan

```

1 package latihan;
2
3 public class Service {
4     String id, jenis, harga, status;
5
6     public String getId() {
7         return id;
8     }
9     public void setId(String id) {
10        this.id = id;
11    }
12    public String getJenis() {
13        return jenis;
14    }
15    public void setNama(String jenis) {
16        this.jenis = jenis;
17    }
18    public String getharga() {
19        return harga;
20    }
21    public void setharga(String harga) {
22        this.harga = harga;
23    }
24    public String getstatus() {
25        return status;
26    }
27    public void setstatus(String status) {
28        this.status = status;
29    }
30 }
31
32

```

latihan tambahan diberikan dengan cara menambahkan class lain seperti Customer, Transaksi, atau LaundryItem. Latihan ini dimaksudkan agar

pemahaman mengenai class, object, method, serta encapsulation dapat diperdalam melalui studi kasus aplikasi laundry sederhana.

D. Kesimpulan

Dari praktikum ini dapat disimpulkan bahwa **pembuatan class dengan atribut serta penerapan prinsip encapsulation melalui getter dan setter berhasil dilakukan. Pengelolaan kode melalui package telah diterapkan** sehingga struktur program dapat dijaga dengan baik. Selain itu, **object berhasil digunakan** untuk merepresentasikan entitas nyata berupa pelanggan laundry.

Melalui implementasi GUI sederhana, yaitu Login Frame dan Main Frame, **integrasi antara konsep OOP dan antarmuka pengguna berhasil diperlihatkan.** Dengan demikian, praktikum ini menunjukkan bahwa pemrograman berorientasi objek **dapat digunakan untuk membangun aplikasi yang modular, aman, terstruktur, dan mudah dikembangkan di masa mendatang.**