

LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS OBJEK
CLASS, OBJECT, CONSTRUCTOR dan METHOD



OLEH:
DEVINA AMANDA PUTRI
(2411533009)

DOSEN PENGAMPU:
NURFIAH, S.ST, M.KOM

FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS

2025

A. PENDAHULUAN

Pemrograman Berbasis Objek (PBO) merupakan salah satu paradigma pemrograman yang berfokus pada penggunaan objek untuk memecahkan permasalahan. Konsep ini muncul karena program yang ditulis secara prosedural sering kali sulit untuk dikembangkan lebih lanjut dan cenderung tidak terstruktur ketika program menjadi besar. Dengan PBO, permasalahan dunia nyata dapat dimodelkan ke dalam class dan object, sehingga program lebih mudah dipahami, dipelihara, dan dikembangkan.

Class adalah blueprint atau cetak biru dari sebuah objek yang di dalamnya berisi atribut (properties) dan perilaku (method). Class bersifat abstrak, artinya class tidak dapat digunakan langsung, tetapi harus diinstansiasi menjadi sebuah object. Object inilah yang menjadi representasi nyata dari entitas di dunia nyata. Misalnya, dalam konteks aplikasi laundry, pelanggan (customer) dan layanan (service) dapat direpresentasikan sebagai object dari class tertentu.

Selain class dan object, terdapat konsep penting lain yaitu encapsulation, constructor, dan method.

- Encapsulation berarti membungkus data agar lebih aman dengan cara menyembunyikan atribut menggunakan modifier private, kemudian menyediakan akses melalui method setter dan getter.
- Constructor adalah blok kode khusus yang dipanggil saat sebuah object dibuat. Constructor biasanya digunakan untuk memberikan nilai awal pada atribut object.
- Method adalah fungsi atau prosedur yang didefinisikan dalam sebuah class. Method digunakan untuk melakukan aksi tertentu, baik untuk mengolah data maupun memberikan hasil. Dalam Java, terdapat method bawaan (seperti `toString()` atau `equals()`) dan method buatan programmer sendiri.

Dalam praktikum ini, penerapan konsep PBO difokuskan pada pembuatan Aplikasi Laundry. Studi kasus ini dipilih karena laundry merupakan layanan yang dekat dengan kehidupan sehari-hari sehingga mudah dipahami. Aplikasi ini mencakup class User untuk login, class Customer untuk data pelanggan, class Service untuk data layanan laundry, dan class Order untuk data transaksi pesanan. Dengan membuat beberapa class tersebut, dapat dilihat bagaimana class dan object bekerja sama membentuk program yang lebih kompleks dan terstruktur.

B. TUJUAN

Tujuan praktikum ini yaitu mahasiswa mampu membuat class, object, encapsulation, constructor, dan method menggunakan bahasa Java serta menerapkannya pada aplikasi manajemen laundry. Adapun capaian yang diharapkan adalah:

- Membuat class, object, encapsulation, constructor, dan method untuk aplikasi laundry (User, Costumer, Service, dan Order).
- Membuat desain antarmuka sederhana untuk aplikasi laundry seperti halaman login dan halaman utama.
- Memahami penggunaan method pada JFrame/tampilan aplikasi.
- Mampu membuat fungsi berpindah dari tampilan login ke halaman utama.

C. LANGKAH KERJA

1. Buat project baru pada eclipse dengan nama laundry apps, lalu buatkan 2 package dengan nama model dan ui.
2. Pada package model, buat class dengan nama User.java lalu tambahkan attribute pada class user yang berisi id, nama, username, dan password dengan tipe data string.



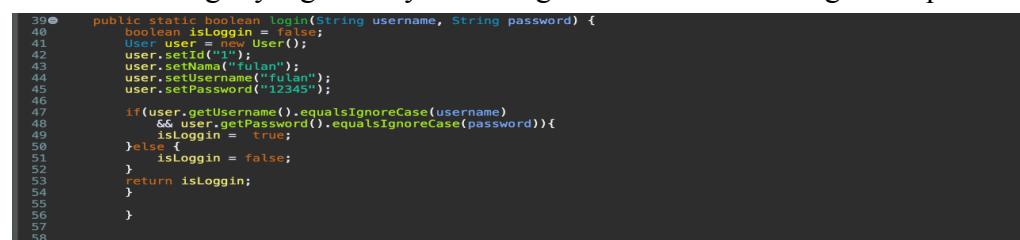
```
File Search Project Run Window Help
eclipse-workspace - laundryapps/src/model/User.java - Eclipse IDE
User.java LoginFrame.java MainFrame.java Customer.java Service.java Order.java 52
1 package model;
2
3 public class User {
4     String id, nama, username, password;
5 }
```

3. Buat setter dan getter untuk mengakses serta mengubah nilai atribut sesuai prinsip encapsulation.



```
5
6     |
7     public String getId() {
8         return id;
9     }
10    public void setId(String id) {
11        this.id = id;
12    }
13    public String getNama() {
14        return nama;
15    }
16    public void setNama(String nama) {
17        this.nama = nama;
18    }
19    public String getUsername() {
20        return username;
21    }
22    public void setUsername(String username) {
23        this.username = username;
24    }
25    public String getPassword() {
26        return password;
27    }
28    public void setPassword(String password) {
29        this.password = password;
30    }
31    }
32    }
33    }
34    }
35    }
36    }
37    }
38    }
```

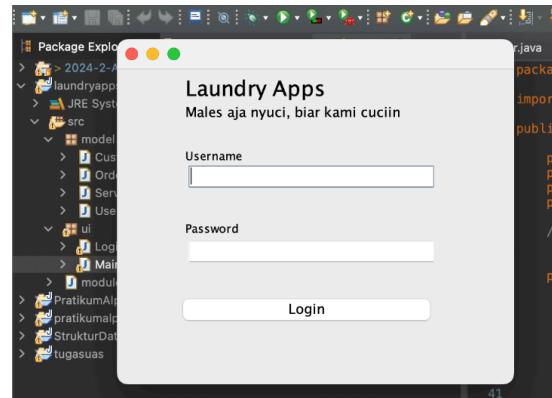
4. Buat method login yang nantinya akan digunakan ketika user login ke aplikasi.



```
39    public static boolean login(String username, String password) {
40        boolean isLogin = false;
41        User user = new User();
42        user.setId("1");
43        user.setNama("fulan");
44        user.setUsername("fulan");
45        user.setPassword("12345");
46
47        if(user.getUsername().equalsIgnoreCase(username)
48            && user.getPassword().equalsIgnoreCase(password)){
49            isLogin = true;
50        }else {
51            isLogin = false;
52        }
53        return isLogin;
54    }
55    }
56    }
57    }
58    }
```

5. Selanjutnya membuat tampilan Login menggunakan JFrame pada package ui, buat JFrame baru dengan nama LoginFrame. Tambahkan komponen JLabel, JTextField untuk username dengan Id txtUsername dan JTextField untuk

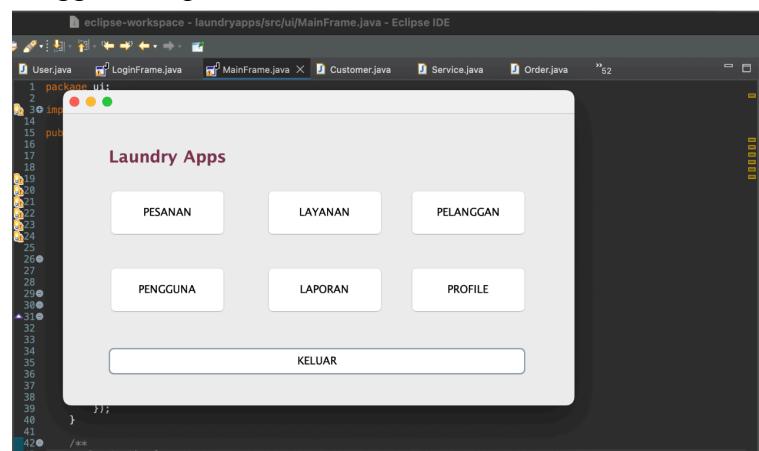
password dengan Id txtPassword, JButton dengan Id btnLogin untuk login.



6. Lalu klik kanan pada button login untuk menambahkan actionPerformed, kemudian panggil method login pada class User dengan mengirimkan parameter dari txtUsername dan txtPassword, jika benar maka akan tampil MainFrame.

```
77     JButton btnLogin = new JButton("Login");
78     btnLogin.addActionListener(new ActionListener() {
79         public void actionPerformed(ActionEvent e) {
80             if(User.login(txtUsername.getText(), txtPassword.getText())) {
81                 new MainFrame().setVisible(true);
82                 dispose();
83             } else
84                 JOptionPane.showMessageDialog(null, "Login Gagal");
85         }
86     });
87     btnLogin.setBounds(56, 215, 249, 29);
88     contentPane.add(btnLogin);
```

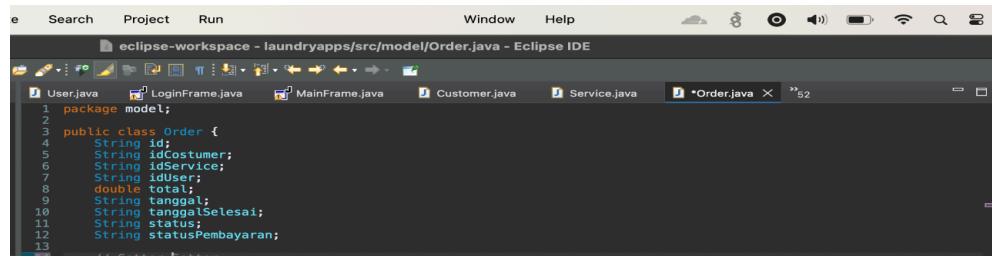
7. Selanjutnya membuat MainFrame menggunakan JFrame pada package ui dengan komponen JLabel dan JButton untuk Pesanan, Layanan, Pelanggan, Pengguna, Laporan, Profile.



D. TUGAS

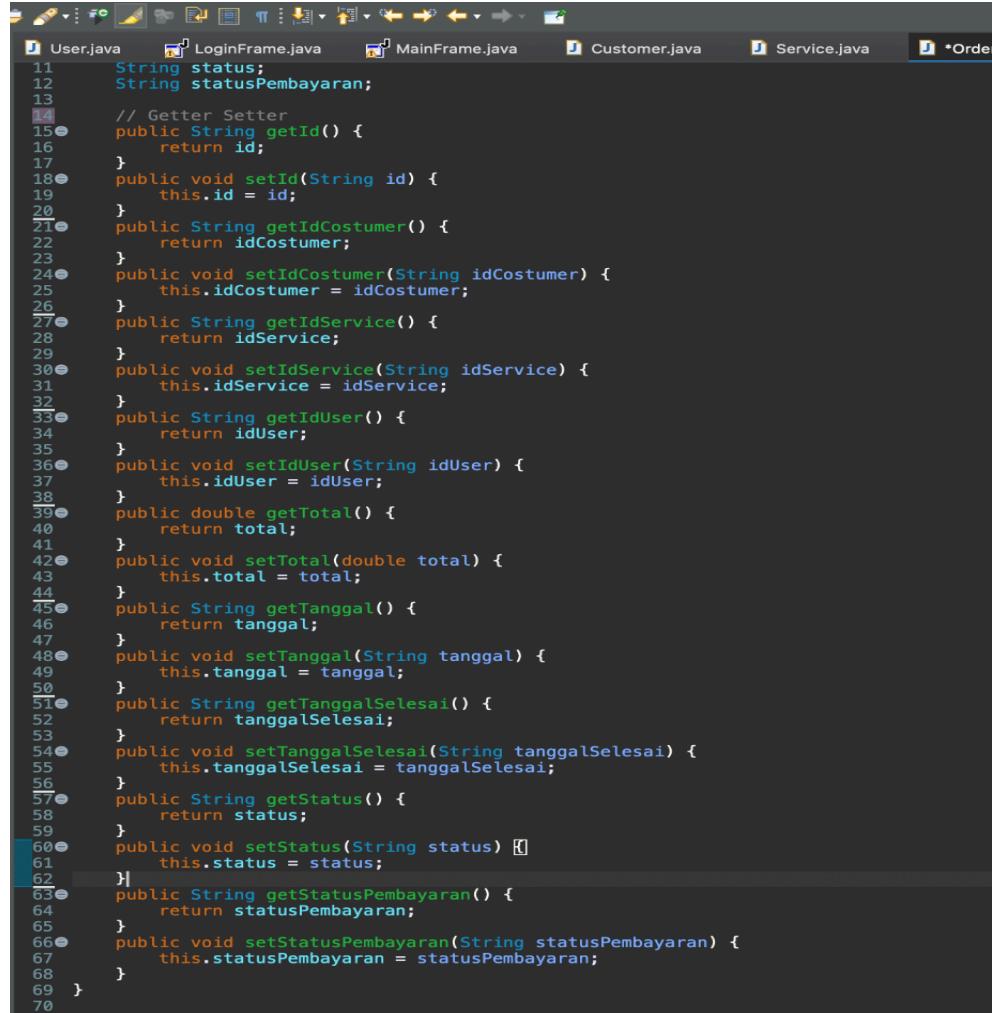
1. Buat class Order pada package model, class ini merepresentasikan pesanan laundry.

2. Tambahkan attribute id, id_costumer, id_service, id_user, total, tanggal, tanggal_selesai, status, status_pembayaran



```
1 package model;
2
3 public class Order {
4     String id;
5     String idCostumer;
6     String idService;
7     StringidUser;
8     double total;
9     String tanggal;
10    String tanggalSelesai;
11    String status;
12    String statusPembayaran;
13 }
```

3. Lalu buat getter setter sesuai dengan prinsip encapsulation.



```
11     String status;
12     String statusPembayaran;
13
14     // Getter Setter
15     public String getId() {
16         return id;
17     }
18     public void setId(String id) {
19         this.id = id;
20     }
21     public String getIdCostumer() {
22         return idCostumer;
23     }
24     public void setIdCostumer(String idCostumer) {
25         this.idCostumer = idCostumer;
26     }
27     public String getIdService() {
28         return idService;
29     }
30     public void setIdService(String idService) {
31         this.idService = idService;
32     }
33     public String getIdUser() {
34         return idUser;
35     }
36     public void setIdUser(String idUser) {
37         this.idUser = idUser;
38     }
39     public double getTotal() {
40         return total;
41     }
42     public void setTotal(double total) {
43         this.total = total;
44     }
45     public String getTanggal() {
46         return tanggal;
47     }
48     public void setTanggal(String tanggal) {
49         this.tanggal = tanggal;
50     }
51     public String getTanggalSelesai() {
52         return tanggalSelesai;
53     }
54     public void setTanggalSelesai(String tanggalSelesai) {
55         this.tanggalSelesai = tanggalSelesai;
56     }
57     public String getStatus() {
58         return status;
59     }
60     public void setStatus(String status) {
61         this.status = status;
62     }
63     public String getStatusPembayaran() {
64         return statusPembayaran;
65     }
66     public void setStatusPembayaran(String statusPembayaran) {
67         this.statusPembayaran = statusPembayaran;
68     }
69 }
```

E. KESIMPULAN

Dari praktikum ini dapat disimpulkan bahwa Pemrograman Berbasis Objek (PBO) memberikan cara yang lebih terstruktur dalam membangun program dengan memanfaatkan konsep class dan object. Melalui studi kasus aplikasi laundry, kita bisa melihat bagaimana dunia nyata dapat dimodelkan ke dalam kode program. Setiap class memiliki peran yang jelas, misalnya User untuk login, Costumer untuk menyimpan data pelanggan, Service untuk jenis layanan, dan Order untuk transaksi pesanan. Dengan adanya constructor, encapsulation, dan method, program menjadi lebih aman, rapi, serta mudah dikembangkan. Secara keseluruhan, praktikum ini membantu mahasiswa memahami dasar PBO sekaligus melatih keterampilan menghubungkan teori dengan implementasi nyata dalam bentuk aplikasi sederhana.