

LAPORAN PRAKTIKUM PBO

PEKAN KE – 1

CLASS DAN OBJECT



Oleh:

Daffa Aira Adrin

NIM 2411531006

MATA KULIAH PRAKTIKUM PEMROGRAMAN BERORIENTASI

OBJEK

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

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG

A. Pendahuluan

a) Latar Belakang

Pemrograman berorientasi objek (Object-Oriented Programming / OOP) adalah suatu metodologi dalam membuat suatu program menggunakan **object** dan **class**. OOP merupakan metodologi yang paling banyak digunakan pada bahasa pemrograman modern, termasuk bahasa pemrograman Java. Metodologi ini memudahkan pengembang untuk membuat sebuah program yang modular, dan terstruktur. Tujuan utama dari OOP adalah menggabungkan data dan fungsi yang beroperasi pada data tersebut agar bagian kode lain yang dapat mengakses data tersebut kecuali fungsi yang telah ditentukan.

b) Tujuan

Tujuan dari pelaksanaan praktikum ini sebagai berikut:

- 1) Memahami konsep dasar dari class, object, dan constructor.
- 2) Memahami dan mengimplementasikan kode untuk membuat sebuah class, object, dan constructor di bahasa pemrograman Java.
- 3) Mendesain sebuah GUI sebagai implementasi OOP.

c) Landasan Teori

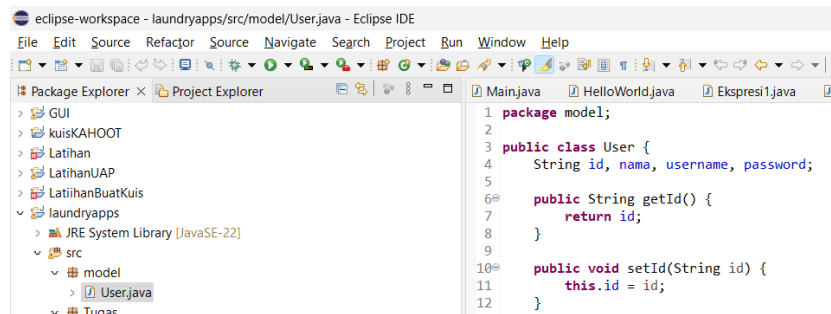
Class di dalam Java adalah templat untuk membuat entitas dunia nyata yang memiliki atribut dan properti mirip yang disebut dengan objek. **Objek** di dalam Java adalah sebuah instansi dari sebuah **class** yang memiliki sejumlah atribut, dan **method**.

Constructor adalah sebuah **method** bertipe khusus yang digunakan untuk menginisialisasikan sebuah objek. Fungsi utamanya adalah menetapkan keadaan awal dari sebuah objek.

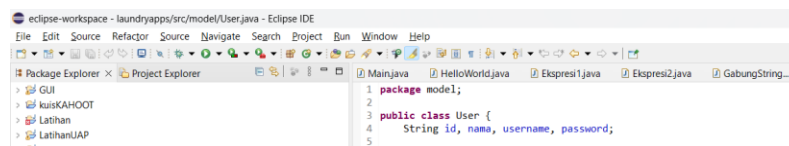
B. Langkah-Langkah Pengerjaan

a. Class User

1. Buat proyek baru pada **Package Explorer** dan beri nama “laundryapps”. Kemudian, pada **project** buat **package** baru dengan nama “model”.
2. Buat **class** baru dengan nama “User”.



3. Setelah itu, deklarasikan atribut **class** yang terdiri id, nama, username, dan password dengan **datatype String**.

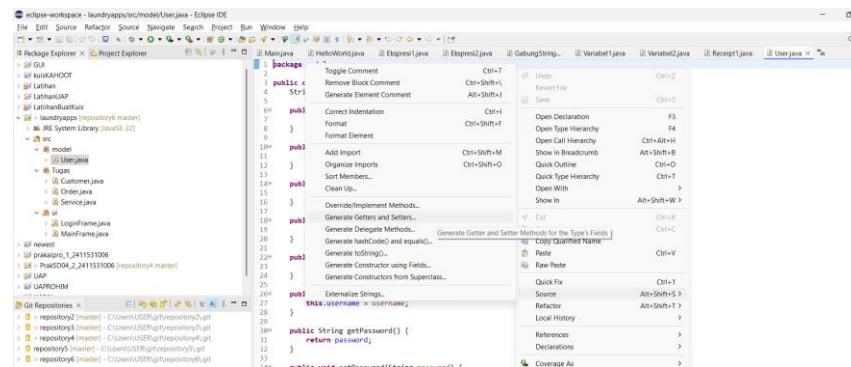


4. Buatkan **method accessor** dan **mutator** untuk setiap atribut dengan menggunakan shortcut yang terdapat pada Eclipse IDE.

```

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 }

```



5. Setelah itu, buat **static method** dengan nama “login” dengan **access type public**, dan **return type Boolean**. Tuliskan 2 buah parameter, yaitu “username” dengan **datatype String**, dan “password” dengan **datatype String**.

6. Inisialisasikan variabel “isLogin” dengan **datatype boolean** dengan nilai **false** di dalam **method “login”**.

7. Instansiasikan objek dari **class “User”** dengan nama “user”.

```

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     User user = new User();

```

8. Inisialisasikan atribut “id” dari objek dengan menggunakan **method mutator “setID”** dengan nilai “1”.

9. Inisialisasikan atribut “nama” dari objek dengan menggunakan **method mutator “setNama”** dengan nilai “fulan”.

10. Inisialisasikan atribut “username” dari objek dengan menggunakan **method mutator “setUsername”** dengan nilai “fulan”.

11. Inisialisasikan atribut “password” dari objek dengan menggunakan **method mutator** “Password” dengan nilai “12345”.

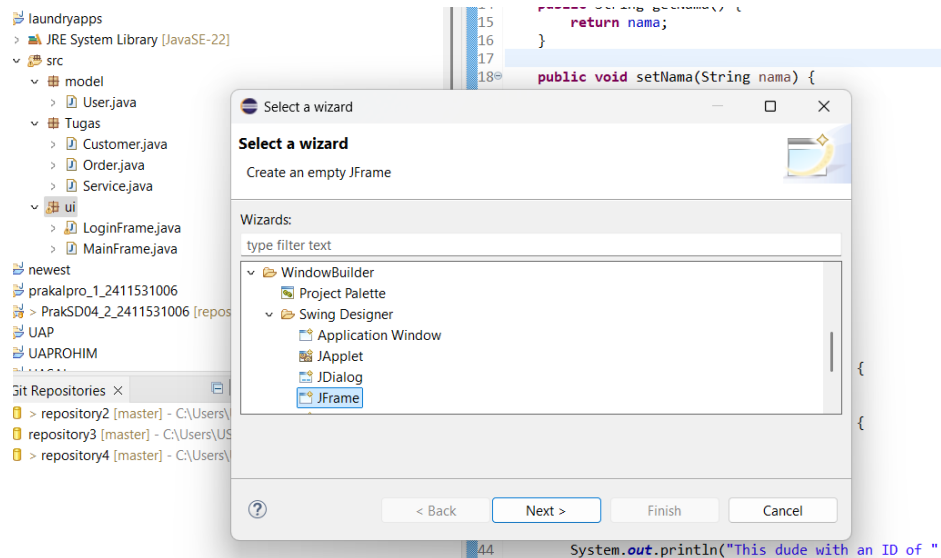
```
44 public void setPassword(String password) {  
45     this.password = password;  
46 }  
47  
48 public static boolean login(String username, String password) {  
49     boolean isLogin = false;  
50     User user = new User();  
51     user.setId("1");  
52     user.setName("fulan");  
53     user.setUsername("fulan");  
54     user.setPassword("12345");  
55 }
```

12. Setelah itu, tuliskan **if statement** untuk memeriksa apakah pengguna memiliki akun dan password yang sesuai. Jika **username**, dan **password** yang diinputkan benar, maka ubah nilai variabel “isLogin” menjadi “true”. Jika salah, ubah nilai variabel “isLogin” menjadi “false”.
13. Kemudian, kembalikan variabel “isLogin”.

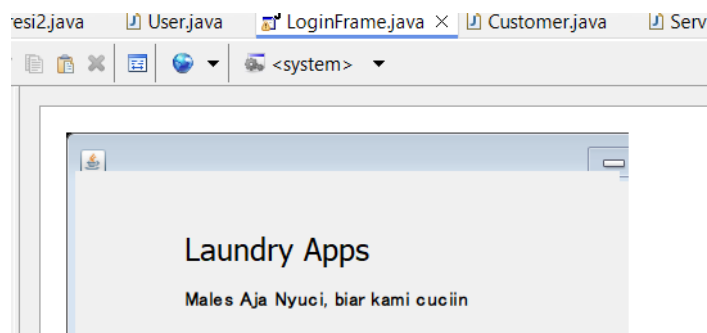
```
34 public void setPassword(String password) {  
35     this.password = password;  
36 }  
37  
38 public static boolean login(String username, String password) {  
39     boolean isLogin = false;  
40     User user = new User();  
41     user.setId("1");  
42     user.setName("fulan");  
43     user.setUsername("fulan");  
44     user.setPassword("12345");  
45  
46     if (user.getUsername().equalsIgnoreCase(username) && user.getPassword().equalsIgnoreCase(password)) {  
47         isLogin = true;  
48     } else {  
49         isLogin = false;  
50     }  
51     return isLogin;  
52 }  
53  
54  
55 }
```

b. Login Interface

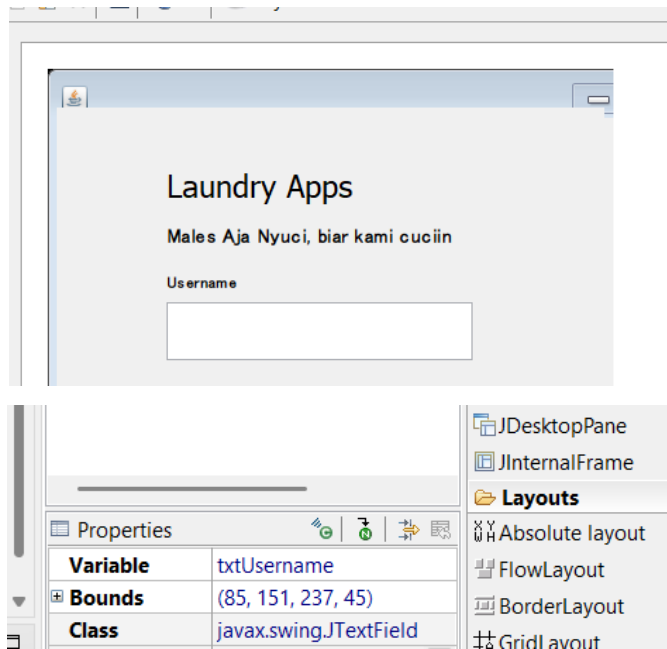
1. Buat **package** baru dengan nama “ui”
2. Buat JFrame baru pada **package** “ui” dan berikan nama “LoginFrame”.
3. Pada tab **design**, ubah **layout** JPanel dengan **absolute layout** agar bisa memindahkan elemen dengan bebas tanpa dibatasi dengan **constraint** yang ada pada **layout** yang lain, dan atur ukuran JPanel sesuai dengan kebutuhan.



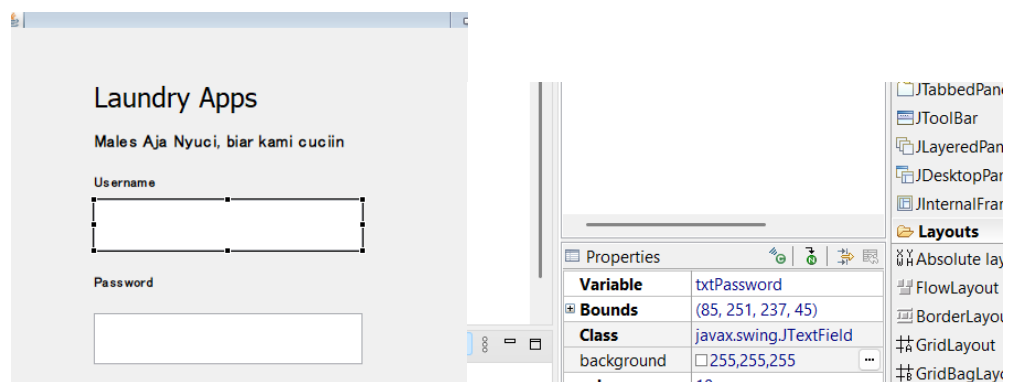
4. Masukkan JLabel, dan atur JLabel agar menampilkan tulis “Laundry Apps”.
5. Tekan JLabel yang baru dibuat, kemudian lakukan **copy-paste**, dan atur JLabel agar menampilkan tulisan “Males Aja Nyuci, biar kami cuciin”, dan letakkan di bawah JLabel yang pertama.



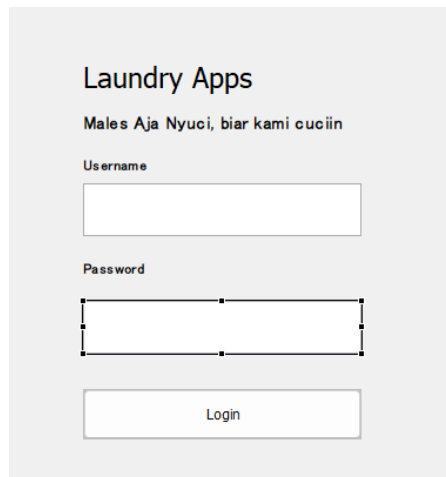
6. Lakukan hal yang sama untuk satu buah JLabel lagi, dan atur JLabel agar menampilkan teks “username”, dan letakkan di bawah JLabel kedua. Atur ketiga JLabel agar ukuran **font** memiliki ukuran terbesar, dan bertambah kecil untuk JLabel yang selanjutnya.
7. Setelah itu, tambahkan JTextField ke dalam JPanel, dan letakkan di bawah JLabel yang menampilkan tulisan “username”, dan atur ukuran agar bisa menampilkan teks dengan jelas. Ubah nama variabel dari JTextField pada **window properties** menjadi “txtUsername”.



8. Kemudian, tambahkan JLabel, dan atur JLabel agar menampilkan tulisan “password”. Letakkan JLabel di bawah JTextField pertama. Atur JLabel agar memiliki ukuran font yang sama dengan JLabel yang menampilkan tulisan “username”.
9. Setelah itu, tambahkan JTextField ke dalam JPanel, dan letakkan di bawah JLabel yang menampilkan tulisan “password”, dan atur ukuran agar bisa menampilkan teks dengan jelas. Ubah nama variabel dari JTextField pada **window properties** menjadi “txtPassword”.



10. Kemudian, tambahkan JButton pada JPanel. Atur JButton agar menampilkan tulisan “Login”, dan memiliki ukuran yang sama dengan JTextField.



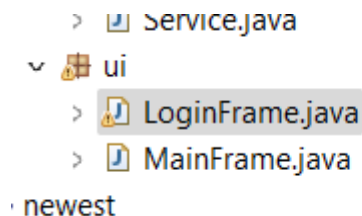
11. Pada **tab source**, Tambahkan ActionListener pada tombol **btnLogin** untuk memanggil method `User.login()` dengan parameter `txtUsername.getText()` dan `txtPassword.getText()`. Jika return true, maka tampilkan `MainFrame` dan tutup `LoginFrame`. Jika false, tampilkan pesan dialog “Login Gagal”.

```
txtPassword.setBounds(85, 251, 237, 45);
contentPane.add(txtPassword);

JButton btnLogin = new JButton("Login");
btnLogin.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(User.login(txtUsername.getText(), txtPassword.getText())) {
            new MainFrame().setVisible(true);
            dispose();
        } else {
            JOptionPane.showMessageDialog(null, "Login Gagal");
        }
    }
});
btnLogin.setFont(new Font("Tahoma", Font.PLAIN, 12));
btnLogin.setBackground(new Color(192, 192, 192));
btnLogin.setBounds(85, 326, 237, 43);
contentPane.add(btnLogin);
```

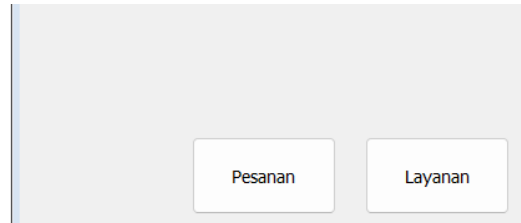
c. Main Interface

1. Buat JFrame baru pada package “ui” dan beri nama `MainFrame`.

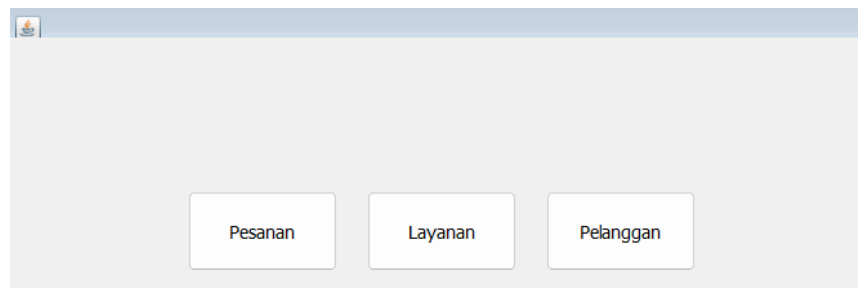


2. Pada tab design, ubah layout `JPanel` dengan absolute layout agar bisa memindahkan elemen dengan bebas tanpa dibatasi constraint dari layout manager yang lain, lalu atur ukuran sesuai kebutuhan.
3. Tambahkan `JButton` dengan tulisan “**Pesanan**”, kemudian letakkan di bagian atas tengah `JPanel`.

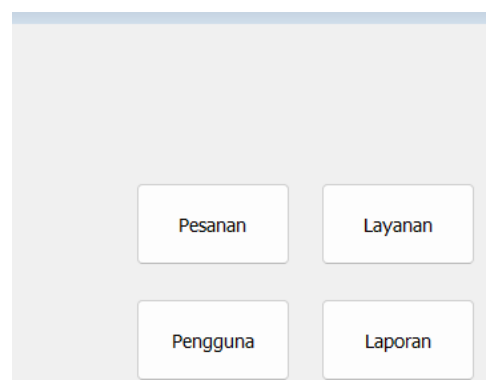
4. Tambahkan JButton dengan tulisan **“Layanan”**, kemudian letakkan di samping kanan tombol Pesanan.



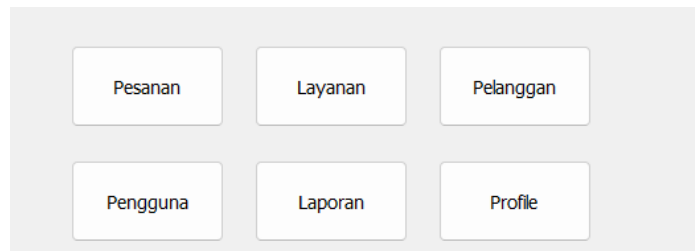
5. Tambahkan JButton dengan tulisan **“Pelanggan”**, kemudian letakkan di samping kanan tombol Layanan.



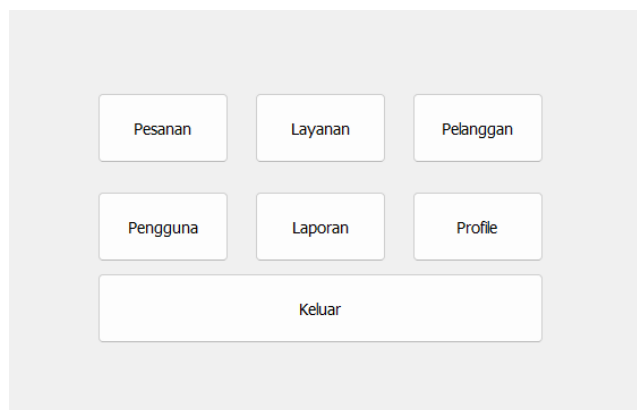
6. Tambahkan JButton dengan tulisan **“Pengguna”**, kemudian letakkan di bawah tombol Pesanan.
7. Tambahkan JButton dengan tulisan **“Laporan”**, kemudian letakkan di samping kanan tombol Pengguna.



8. Tambahkan JButton dengan tulisan **“Profile”**, kemudian letakkan di samping kanan tombol Laporan.



9. Tambahkan JButton dengan tulisan **“Keluar”**, kemudian letakkan di bawah tombol-tombol sebelumnya, dengan ukuran lebih besar yang sejajar dengan lebar tombol di atasnya.



d. Tugas

1. Buat package baru dengan nama **“Tugas”**.
2. Buat class baru dengan nama **Customer** pada **package Tugas**.

```

1 package Tugas;
2
3 public class Customer {
4     String id, nama, Alamat, nomorHp;
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() {

```

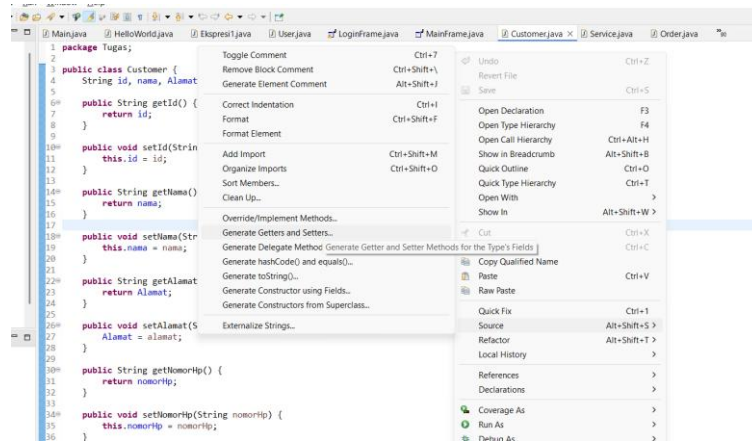
3. Deklarasikan atribut class berupa id, nama, alamat, dan nomorHp dengan **tipe data String**.

```

1 package Tugas;
2
3 public class Customer {
4     String id, nama, Alamat, nomorHp;
5

```

4. Buat method accessor (getter) dan mutator (setter) untuk setiap atribut menggunakan shortcut di Eclipse IDE.



5. Pada method main, instansiasikan objek dari class “Customer” dengan nama variabel “dude”. Setelah itu, gunakan **method setter** untuk memberikan nilai pada setiap atribut (id, nama, alamat, nomorHp).
6. Cetak informasi objek “Customer” tersebut ke **console** menggunakan `System.out.println()` dengan mengakses nilai atribut melalui **method getter**.

```

public void setNomorHp(String nomorHp) {
    this.nomorHp = nomorHp;
}

public static void main(String args[]) {
    Customer dude = new Customer();
    dude.setId("dude123");
    dude.setNama("Abdullah");
    dude.setAlamat("Jl. Cipaganti");
    dude.setNomorHp("0812345");

    System.out.println("This dude with an ID of "
        + dude.getId() + "lives in " + dude.getAlamat() + " his name is "
        + dude.getNama() + " with a phone number of " + dude.getNomorHp());
}
}

```

C. KESIMPULAN

Dari praktikum ini dapat disimpulkan bahwa **class** dan **object** merupakan dasar dari **object-oriented programming (OOP)**. Penggunaan OOP dilakukan dengan pembuatan **class** dengan nama “User”, atribut, **method accessor-mutator**, dan **object**. Implementasi OOP dilakukan dengan validasi input dengan penggunaan **if-statement**, dan diintegrasikan dengan GUI yang dirancang menggunakan **Window Builder** pada **Eclipse IDE**.