

PRAKTIKUM 1

CLASS, OBJECT, CONSTRUCTOR dan METHOD

1.1 Tujuan

Tujuan praktikum ini yaitu mahasiswa mampu membuat class, object, encapsulation, constructor dan method menggunakan Bahasa java dan menerapkannya pada aplikasi manajemen laundry, Adapun pada praktikum ini mahasiswa akan mempelajari beberapa poin yaitu :

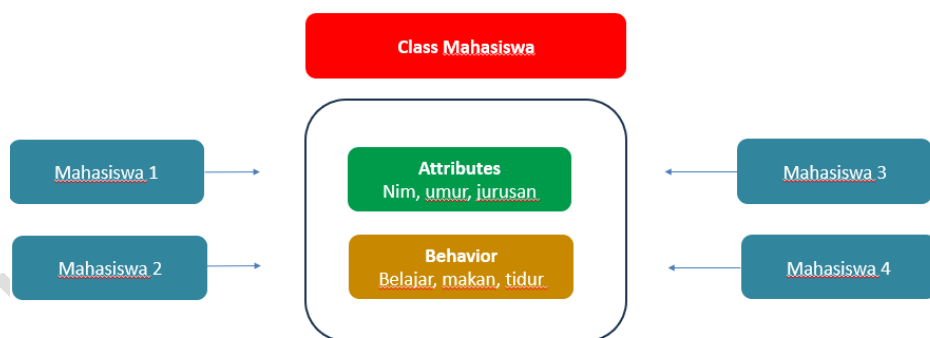
- Membuat class, object, encapsulation, constructor dan method untuk aplikasi laundry seperti class User, Customer, Service dan Order
- Membuat desain antarmuka aplikasi Laundry Login dan Halaman Utama
- Mampu menggunakan method pada JFrame/Tampilan aplikasi
- Mampu membuat fungsi berpindah dari tampilan login ke halaman utama

1.2 Alat

- Computer / laptop yang telah terinstall JDK dan Eclipse

1.3 Teori

Class merupakan sekumpulan objek yang memiliki karakteristik/sifat/behavior dan properties/atribut yang sama, class juga bisa disebut sebagai template/blueprint dari objek yang akan dibuat. Misalkan misalkan ada class mahasiswa, yaitu nama-nama mahasiswa misalkan fulan merupakan sebuah object.



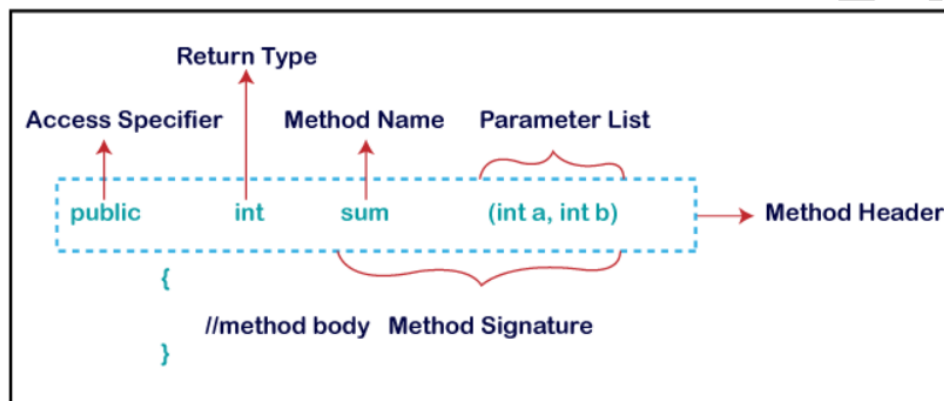
Sebuah class bukan sebuah entitas didalam dunia nyata melainkan sebagai blueprint untuk membuat suatu objek, class didalam java dapat berisi Data Member, Method, Constructor, Nested Class dan Interface.

Object merupakan representasi entitis yang ada pada dunia nyata, sebuah object terdiri dari :

- State merepresentasikan attribute dari sebuah object
- Behavior merepresentasikan method dari sebuah object
- Identity nama yang unik sebuah objek dan digunakan untuk interaksi dengan objek lain.



Method merupakan sebuah blok kode yang dapat dijalankan berulang kali, sehingga lebih terorganisir dan dapat digunakan Kembali, contoh method yang disediakan oleh java adalah method `main()`, `equals()`, `toString()` dll. Pola sebuah method dapat dilihat seperti gambar berikut :



Aturan pembuatan method pada java :

- Penamaan method harus menggunakan **verb** dan dimulai dengan **lowercase** letter
- Jika nama method lebih dari satu kata maka kata pertama harus **verb** dan selanjutnya **adjective** atau **noun** seperti `sum()`, `area()`
- jika nama method terdiri lebih dari 1 kata, maka kata pertama **lowercase** dan kata selanjutnya **uppercase** seperti `areaOfCircle()`

Method pada java terdiri dari beberapa jenis, berikut jenis-jenis method yang ada pada Java.

- **Predefined Method** (standard library method or built-in method) method yang sudah disediakan oleh java seperti `length()`, `equals()`, `compareTo()`, `print()`
- **User-defined Method** yaitu method yang dibuat oleh user atau Programmer, method dibuat sesuai dengan kebutuhan
- **Static method** yaitu Method yang menggunakan `static` sebelum nama method, keutamaan method `static` yaitu dapat memanggil/menggunakan method tanpa harus membuat object terlebih dahulu.

- Instance method merupakan method non static yaitu harus membuat object terlebih dahulu Ketika akan menggunakannya. Instance method terdiri dari 2 jenis yaitu accessor method dan mutator method.
 - Accessor Method digunakan untuk membaca instance, menggunakan kata kunci **get** atau disebut **getter**.

```
public String getNim() {
    return nim;
}
```

- Mutator Method digunakan untuk membaca dan mengubah nilai, menggunakan kata kunci **set** atau disebut juga dengan **setter** dan akan mengembalikan value yang bersifat private.

```
public void setNim(String nim) {
    this.nim = nim;
}
```

- Abstract Method yaitu method yang tidak memiliki body, dideklarasikan didalam class abstract dengan menggunakan kata kunci **abstract**.
- Factory Method yaitu method yang digunakan untuk mengembalikan object ke class yang bersangkutan, semua method static disebut factory method contohnya **NumberFormat**
obj = NumberFormat.getNumberInstance();

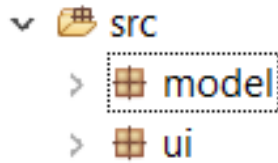
Constructor yaitu blok kode yang mirip dengan method, menggunakan kata kunci **new** Ketika akan membuat sebuah object, nama constructor harus sama dengan nama class.

```
public Costumer(String id, String nama, String alamat, String hp) {
    super();
    this.id = id;
    this.nama = nama;
    this.alamat = alamat;
    this.hp = hp;
}
```

1.4 Langkah-langkah

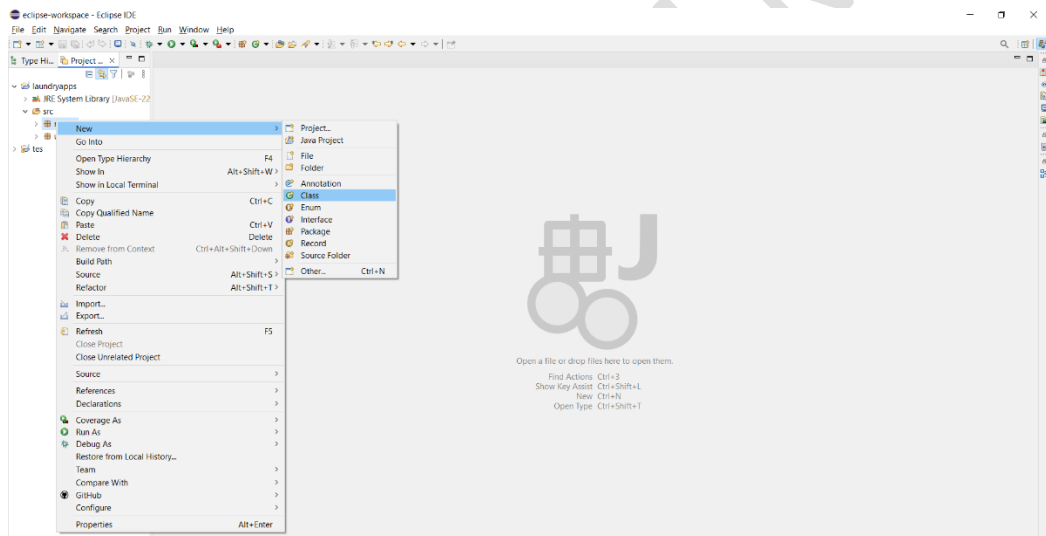
Membuat Class User

- Buat project baru pada eclipse dengna nama **laundryapps**
- Selanjutnya buat 2 buah package pada directory src dengan nama **model** dan **ui**



Package **model** berfungsi untuk menyimpan class-class yang diperlukan untuk pembuatan aplikasi laundry, sedangkan package **ui** digunakan untuk menyimpan tampilan atau antar muka aplikasi laundry.

- Selanjutnya buat class baru pada package model dengan nama **User**.



- Membuat attribute class user yaitu id, nama, username dan password

```
String id, nama, username, password;
```

- Membuat **setter** dan **getter** attribute user, method ini berfungsi untuk memasukkan/mengubah dan menampilkan value dari suatu object.

```
public String getId() {  
    return id;  
}  
  
public void setId(String id) {  
    this.id = id;  
}  
  
public String getNama() {  
    return nama;  
}  
  
public void setNama(String nama) {  
    this.nama = nama;  
}  
  
public String getUsername() {  
    return username;  
}  
  
public void setUsername(String username) {  
    this.username = username;  
}  
  
public String getPassword() {  
    return password;  
}  
  
public void setPassword(String password) {  
    this.password = password;  
}
```

- Membuat Method login yang nantinya akan digunakan Ketika pengguna akan login ke aplikasi.

```
public static boolean login(String username, String password) {  
    boolean isLoggin = false;  
    User user = new User();  
    user.setId("1");  
    user.setNama("fulan");  
    user.setUsername("fulan");  
    user.setPassword("12345");  
  
    if(user.getUsername().equalsIgnoreCase(username)  
        && user.getPassword().equalsIgnoreCase(password)) {  
        isLoggin = true;  
    }else {  
        isLoggin = false;  
    }  
    return isLoggin;  
}
```

Method login ini merupakan method static dikarenakan menggunakan kata kunci static yang mana dapat diakses dari class lain tanpa harus membuat sebuah object terlebih

dahulu. Method login ini menggunakan tipe data Boolean yang mana akan mengembalikan nilai Boolean Ketika digunakan, kemudian memiliki 2 buah parameter yaitu username dan password. Pertama akan variable isLogin diberikan nilai false, kemudian dibuatkan sebuah object user baru, selanjutnya akan dilakukan pencocokan antara data username dan password pada object dengan data username dan password yang dikirimkan dari class lain, jika cocok maka isLogin akan diubah menjadi true dan terakhir method akan mengembalikan nilai isLogin.

Source code lengkap class User seperti gambar dibawah ini :

```

package model;

public class User {
    String id, nama, username, password;

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

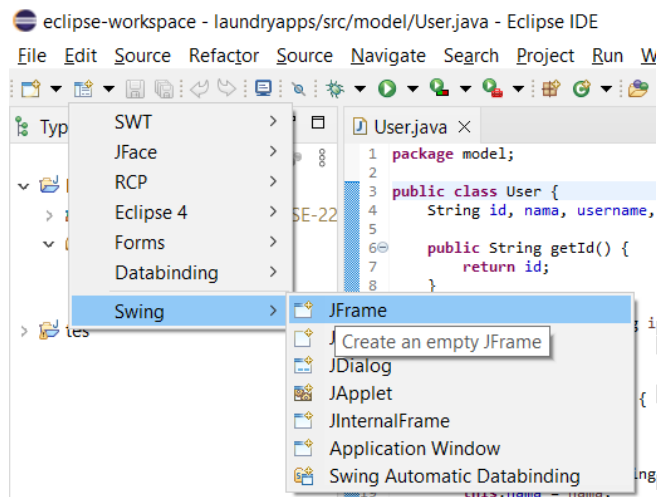
    public static boolean login(String username, String password) {
        boolean isLoggin = false;
        User user = new User();
        user.setId("1");
        user.setNama("fulan");
        user.setUsername("fulan");
        user.setPassword("12345");

        if(user.getUsername().equalsIgnoreCase(username)
            && user.getPassword().equalsIgnoreCase(password)) {
            isLoggin = true;
        }else {
            isLoggin = false;
        }
        return isLoggin;
    }
}

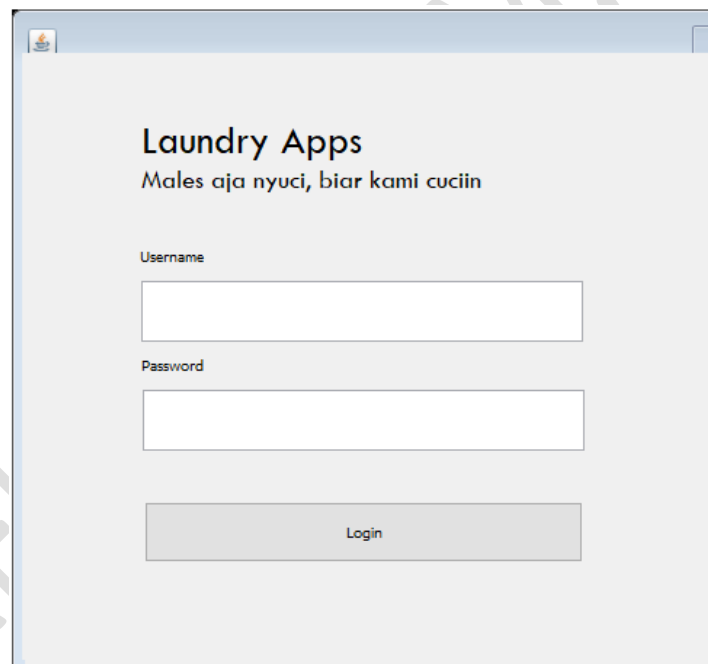
```

Membuat Tampilan Login menggunakan JFrame

- buat JFrame baru pada package ui dengan nama LoginFrame



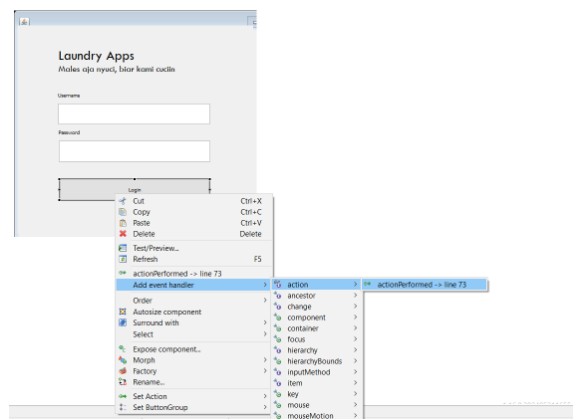
- kemudian desain tampilan login seperti gambar dibawah ini.



Ubah id JTextField username, password dan JButton seperti table dibawah ini.

Component	Id	Keterangan
JTextField	txtUsername	Username
JTextField	txtPassword	Password
JButton	btnLogin	Login

- selanjutnya klik kanan pada button login, pilih add event handler → action → actionPerformed

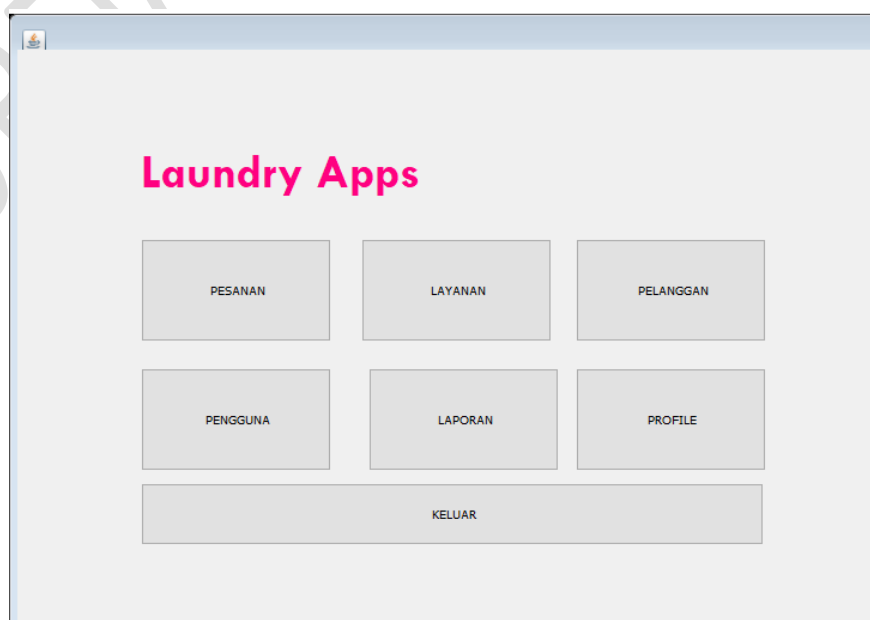


Kemudian panggil method login pada class User dengan mengirimkan parameter yang diambil dari txtUsername dan txtPassword, jika cocok maka akan tampil halaman utama.

```
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");
        }
    }
});
```

Membuat Tampilan Halaman Utama dengan JFrame

- buat sebuah JFrame baru pada package ui dengan nama MainFrame
- selanjutnya desain halaman utama seperti gambar berikut :



1.5 Latihan /Tugas

- Buatlah class dengan nama Costumer dengan attribute id, nama, Alamat dan nomor hp, buatlah setter dan getter untuk menambahkan object Costumer
- Buatlah class dengan nama Service dengan attribute id, jenis, harga dan status, buatlah setter dan getter untuk menambahkan object Service
- Buatlah class dengan nama Order dengan attribute id, id_costumer, id_service, id_user, total, tanggal, tanggal_selesai, status, status_pembayaran , buatlah setter dan getter untuk menambahkan object Order