

LAPORAN PRATIKUM PBO

“ CLASS, OBJECT, CONSTRUCTOR dan METHOD ”



OLEH:

HABIBI PUTRA RIZQULLAH

2411531001

MATA KULIAH : PEMOGRAMAN BERIOENTASI OBJEK

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

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

2025

A. Pendahuluan

Pemrograman Berorientasi Objek (OOP) adalah cara membuat program komputer yang menggunakan kelas dan objek sebagai komponen utama untuk membangun perangkat lunak. Metode ini ternyata lebih baik daripada cara pemrograman lama, dengan ide utamanya adalah menempatkan informasi dan tindakan yang menggunakan informasi tersebut di satu tempat, sehingga hanya tindakan tertentu yang dapat mengaksesnya. Dengan melakukan ini, OOP memastikan data aman dan terorganisir dengan baik, sekaligus membantu programmer membuat program yang terpisah dan terstruktur dengan baik.

OOP telah berkembang menjadi salah satu metode paling umum dalam bahasa pemrograman saat ini, seperti Java, karena mendukung ide-ide penting seperti menjaga privasi, mewariskan sesuatu, memiliki banyak bentuk, dan menggunakan kembali kode. Manfaat-manfaat ini membuat pembuatan perangkat lunak lebih mudah ditangani, diperbarui, dan dapat diubah untuk memenuhi kebutuhan yang mendesak. Oleh karena itu, memahami OOP sangat penting, terutama dalam praktik ini, karena memberikan Anda dasar-dasar untuk membuat program yang mengikuti aturan pembangunan perangkat lunak modern.

B. Tujuan

Tujuan dari pelaksanaan praktikum ini adalah sebagai berikut:

1. Memahami konsep dasar pemrograman berorientasi objek, khususnya terkait class, object, constructor, serta method.
2. Mampu mengimplementasikan kode program dalam bahasa Java untuk membuat class, object, dan constructor.
3. Mendesain serta mengembangkan antarmuka grafis pengguna (Graphical User Interface/GUI) sebagai salah satu bentuk penerapan konsep OOP.

C. Teori Dasar

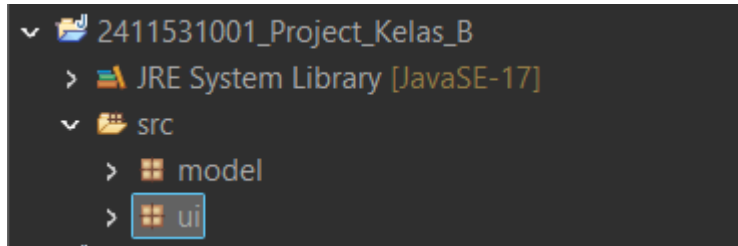
Pemrograman Berorientasi Objek (Object-Oriented Programming/OOP) adalah cara pemrograman yang menekankan penggunaan kelas dan objek sebagai bagian utama dalam pembuatan perangkat lunak. Ide ini muncul untuk mengatasi permasalahan pada pemrograman prosedural (step-by-step programming), yang semakin sulit dikelola seiring meningkatnya kompleksitas program. Dengan OOP, data dan tindakan yang mengelola data tersebut digabungkan menjadi satu kesatuan, sehingga memberikan struktur yang lebih baik, pemeliharaan yang lebih mudah, dan percepatan dalam pengembangan perangkat lunak.

Konsep utama dalam OOP meliputi: class (rancangan), object (instansiasi dari class), constructor (penginisialisasi object), method (tindakan/operasi), encapsulation (pembungkusan data), inheritance (pewarisan), dan polymorphism (kemampuan memiliki banyak bentuk). Java merupakan salah satu bahasa yang paling populer dalam penerapan OOP karena mendukung penuh prinsip-prinsip tersebut, termasuk dalam pembuatan Graphical User Interface (GUI).

D. Langkah – Langkah Pengerjaan

1. Membuat Class User

- Pertama, buatlah sebuah project baru di Eclipse dan beri nama laundryapps.
- Setelah project berhasil dibuat, tambahkan dua package pada direktori src dengan nama model dan ui.



Package model digunakan untuk menyimpan class yang berhubungan dengan data atau logika aplikasi laundry, sedangkan **Package ui** dipakai untuk menampung class tampilan atau antarmuka aplikasi.

- Langkah berikutnya adalah membuat sebuah class baru di dalam package model dengan nama kelas yaitu **User**.
- Setelah itu, tambahkan atribut pada class berupa **id**, **nama**, **username**, dan **password**, yang seluruhnya menggunakan tipe data **String**.

```
1 package model;
2
3 public class User {
4     String id, nama, username, password;
5 }
```

- Membuat **setter** dan **getter** pada atribut User, yaitu method yang berfungsi untuk memberikan atau mengubah nilai pada sebuah objek serta menampilkan nilai dari objek tersebut.

```
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 getName() {
15        return nama;
16    }
17
18    public void setName(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 }
```

- Selanjutnya, buat sebuah method login pada class User yang digunakan untuk proses autentikasi pengguna. Method ini bersifat public static dengan tipe kembalian boolean, serta menerima parameter berupa username dan password.
- Di dalam method tersebut, deklarasikan variabel isLoggedIn bertipe boolean dengan nilai awal false sebagai penanda status login.

```

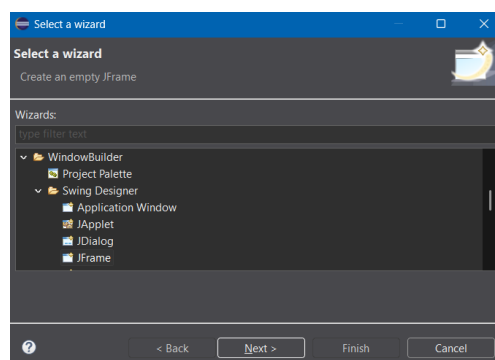
37
38 public static boolean login(String username, String password) {
39     boolean isLoggedIn = false;
40     User user = new User();
41     user.setId("1");
42     user.setNama("fulan");
43     user.setUsername("fulan");
44     user.setPassword("12345");
45
46     if (user.getUsername().equalsIgnoreCase(username)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

```

- Buat objek baru dari class User dengan perintah `User user = new User();`. Selanjutnya, isi atribut pada objek tersebut menggunakan setter:
 - `user.setId("1")` : memberi nilai id dengan "1".
 - `user.setNama("fulan")` : memberi nilai nama dengan "fulan".
 - `user.setUsername("fulan")` : memberi nilai username dengan "fulan".
 - `user.setPassword("12345")` : memberi nilai password dengan "12345".
- Lakukan validasi dengan membandingkan parameter username dan password yang dimasukkan dengan nilai atribut username dan password pada objek **User**. Jika sesuai, variabel isLoggedIn diubah menjadi true; jika tidak sesuai, tetap bernilai false.
- Nilai akhir dari variabel isLoggedIn kemudian dikembalikan oleh method login untuk menunjukkan apakah proses login berhasil atau gagal.

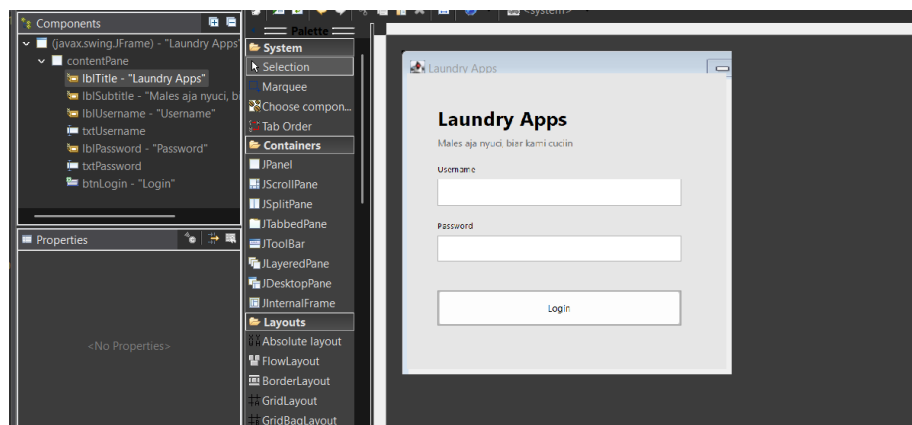
2. Membuat Tampilan Login menggunakan JFrame

- Buat sebuah JFrame baru pada package ui dengan nama LoginFrame

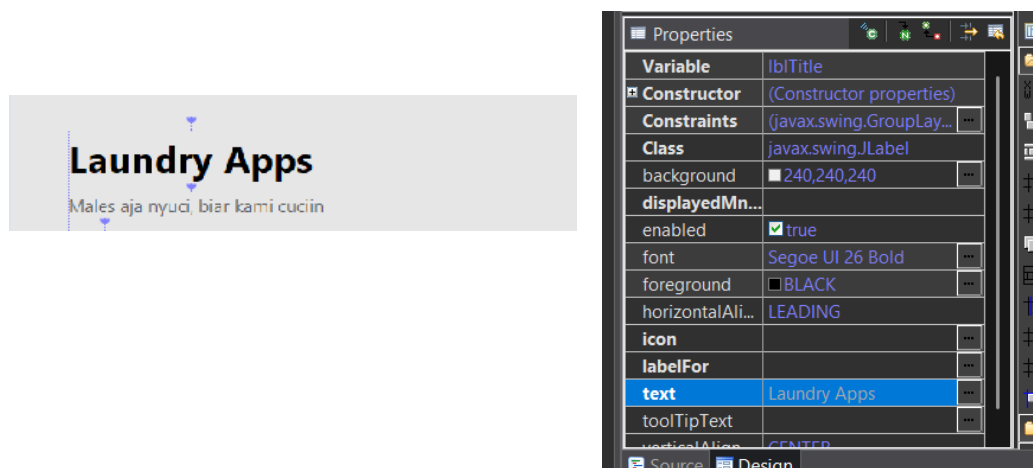


- Kemudian rancang tampilan form login sesuai dengan design pada gambar dibawah.

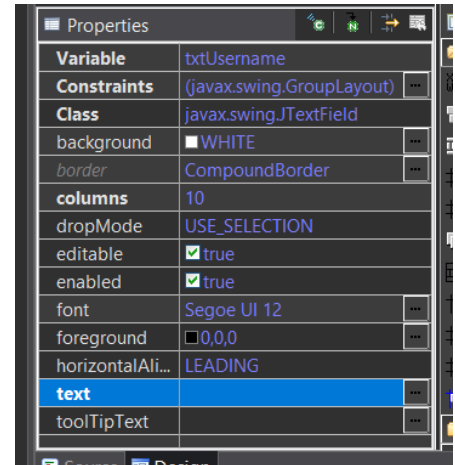
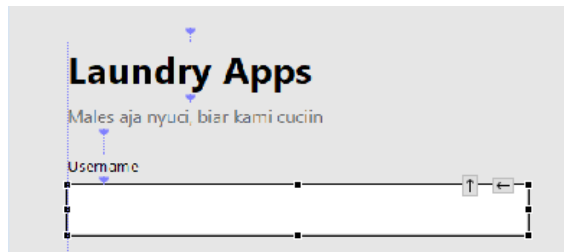
- Langkah selanjutnya membuat sebuah JFrame baru pada package **ui** dengan nama **LoginFrame**. JFrame ini akan menjadi form utama untuk menampilkan halaman login. Judul form diatur menjadi “*Laundry Apps*” agar sesuai dengan nama aplikasi.



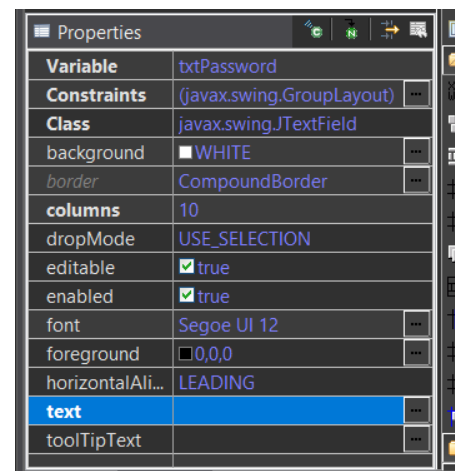
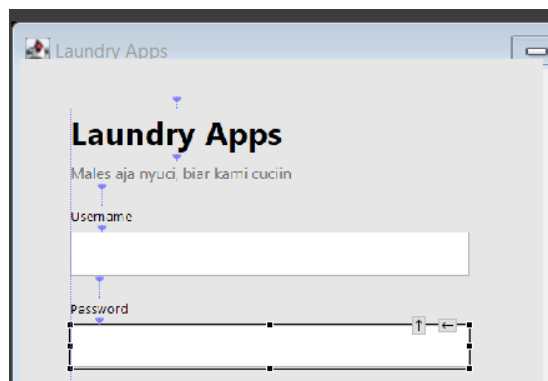
- Setelah JFrame dibuat, tambahkan label sebagai judul utama dengan teks “*Laundry Apps*”. Kemudian, tambahkan label lain sebagai subjudul dengan teks “*Males aja nyuci, biar kami cuciin*”. Judul ditampilkan dengan ukuran huruf lebih besar dan tebal, sedangkan subjudul dibuat lebih kecil untuk memberikan keterangan tambahan.



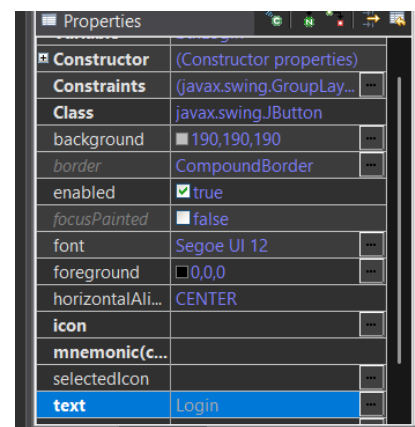
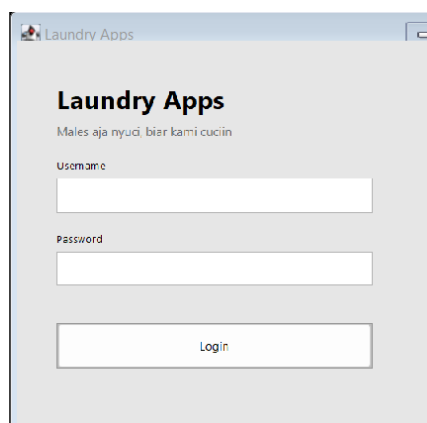
- Selanjutnya, tambahkan label dengan teks “*Username*” untuk memberi keterangan pada kolom input. Di bawah label ini, tambahkan komponen JTextField dengan nama variabel **txtUsername**. Komponen ini berfungsi sebagai tempat pengguna memasukkan username.



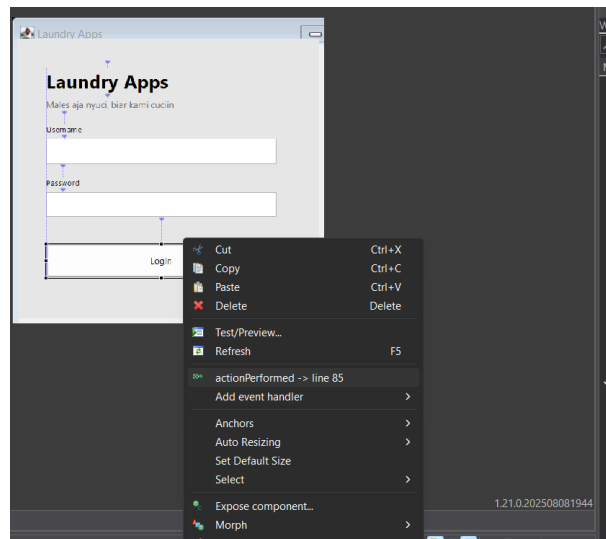
- Kemudian, tambahkan label dengan teks “*Password*” sebagai keterangan untuk kolom sandi. Tambahkan juga JPasswordField di bawah label tersebut dengan nama variabel **txtPassword**, yang berfungsi untuk menyembunyikan karakter password saat diketik oleh pengguna.



- Langkah berikutnya adalah menambahkan sebuah tombol (JButton) dengan teks “Login”. Tombol ini diberi variabel **btnLogin** agar nantinya dapat digunakan dalam proses pemrograman untuk menghubungkan aksi login.



- 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.

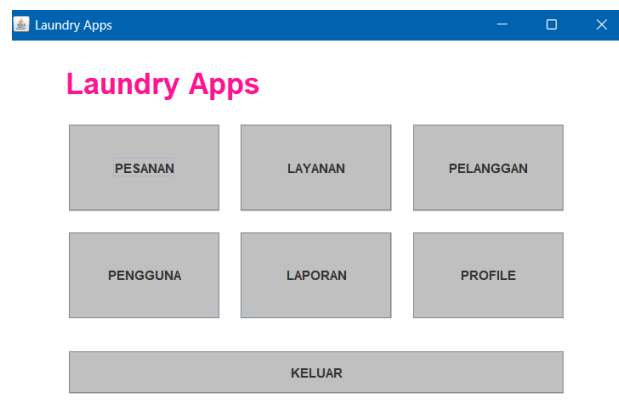
```

83 // Tombol Login
84 btnLogin = new JButton("Login");
85 btnLogin.addActionListener(new ActionListener() {
86     public void actionPerformed(ActionEvent e) {
87         if(User.Login(txtUsername.getText(), txtPassword.getText())) {
88             new MainFrame().setVisible(true);
89             dispose();
90         } else {
91             JOptionPane.showMessageDialog(null, "Login Gagal");
92         }
93     }
94 });

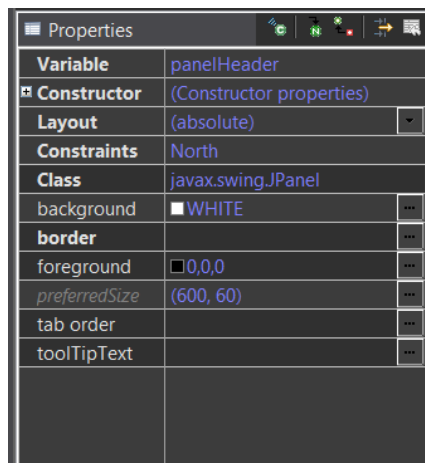
```

3. Membuat Tampilan Halaman Utama dengan JFrame

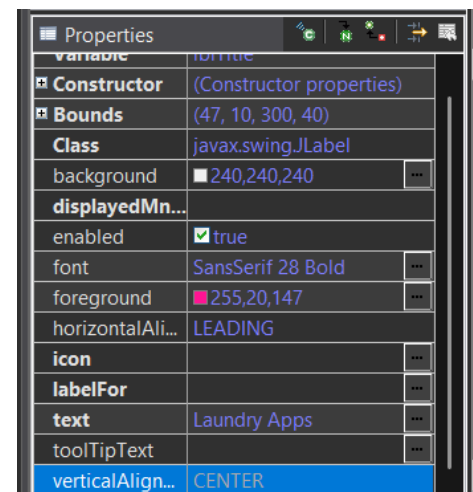
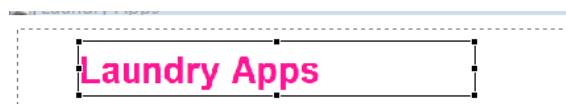
- Buat sebuah JFrame baru dengan nama MainFrame atau HalamanUtama di dalam package ui. JFrame ini akan menjadi halaman utama aplikasi setelah proses login berhasil. Judul form diatur dengan teks “Laundry Apps” .
- Selanjutnya desain halaman utama seperti gambar berikut :



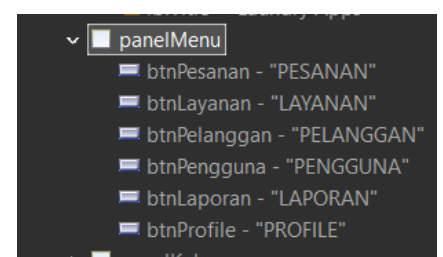
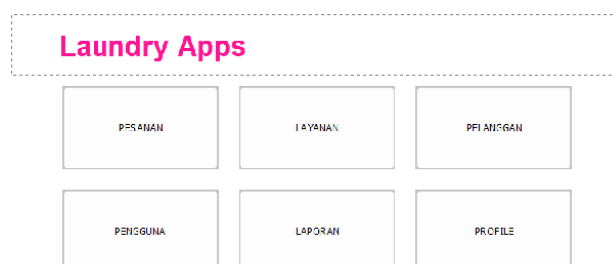
- Pada bagian atas form, tambahkan sebuah panel bernama **panelHeader**. Di dalam panel ini,



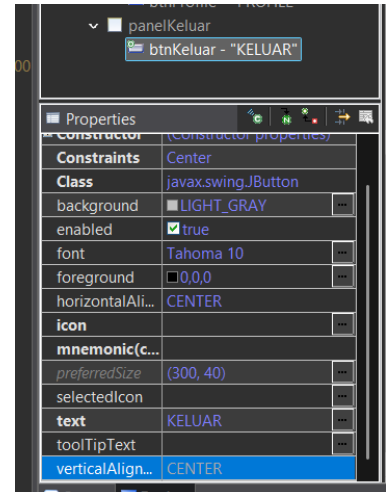
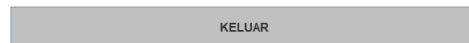
- Lalu letakkan sebuah label dengan teks “*Laundry Apps*” menggunakan ukuran font yang lebih besar, tebal, dan berwarna agar menjadi identitas utama aplikasi



- Selanjutnya, buat sebuah panel bernama **panelMenu** yang digunakan untuk menampung tombol-tombol menu utama aplikasi. Pada panel ini, tambahkan beberapa tombol yaitu **Pesanan**, **Layanan**, **Pelanggan**, **Pengguna**, **Laporan**, dan **Profile**. Setiap tombol akan mewakili fitur yang berbeda pada aplikasi.



- Tambahkan panel khusus bernama **panelKeluar** di bagian bawah halaman utama. Panel ini digunakan untuk menampung tombol **Keluar**, yang berfungsi menutup aplikasi atau kembali ke halaman login.



4. Tugas / Latihan

1. Buatlah class dengan nama **Costumer** dengan attribute **id**, **nama**, **Alamat** dan **nomor hp**, buatkan setter dan getter untuk menambahkan object **Costumer**.
 - Buatlah sebuah package baru dengan nama **Tugas**
 - buat sebuah class baru dengan nama **Costumer** di dalam package **Tugas**.
 - Di dalam class **Costumer**, tambahkan beberapa atribut dengan tipe data **String** yaitu **id**, **nama**, **alamat**, dan **nomorHp**. Atribut-atribut ini digunakan untuk menyimpan informasi dasar tentang seorang pelanggan.

```

1 package Tugas;
2
3 public class Costumer {
4     String id, nama, alamat, nomorHp;
5 }

```

- Setelah mendefinisikan atribut, buat method setter dan getter untuk masing-masing atribut. Method setter digunakan untuk mengisi atau mengubah nilai atribut, sedangkan method getter digunakan untuk mengambil atau menampilkan nilai dari atribut tersebut.

```

6● public String getId() {
7     return id;
8 }
9
10● public void setId(String id) {
11     this.id = id;
12 }
13
14● public String getName() {
15     return nama;
16 }
17
18● public void setName(String nama) {
19     this.nama = nama;
20 }
21
22● public String getAddress() {
23     return alamat;
24 }
25
26● public void setAddress(String alamat) {
27     this.alamat = alamat;
28 }
29
30● public String getNomorHp() {
31     return nomorHp;
32 }
33
34● public void setNomorHp(String nomorHp) {
35     this.nomorHp = nomorHp;
36 }
37

```

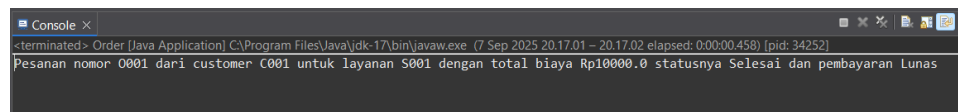
- Langkah terakhir adalah menambahkan method **main** sebagai tempat uji coba class. Pada method ini, buat sebuah objek baru dari class **Costumer** dengan nama **cost**. Kemudian isi data pelanggan menggunakan method setter, seperti ID pelanggan, nama, alamat, dan nomor HP. Setelah itu, gunakan method getter untuk menampilkan data pelanggan menggunakan **System.out.println()**.

```

38● public static void main(String args[]) {
39     Costumer cost = new Costumer();
40     cost.setId("C001");
41     cost.setName("Habibi");
42     cost.setAlamat("Jl. Limau Manis");
43     cost.setNomorHp("082173241590");
44
45     System.out.println("Customer dengan ID "
46         + cost.getId() + " bernama " + cost.getName()
47         + " tinggal di " + cost.getAddress() + " dengan nomor HP " + cost.getNomorHp());
48 }
49

```

- Hasilnya akan menampilkan :



```

Console x
<terminated> Order [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (7 Sep 2025 20:17:01 - 20:17:02 elapsed: 0:00:00.458) [pid: 34252]
Pesanan nomor 0001 dari customer C001 untuk layanan S001 dengan total biaya Rp10000.0 statusnya Selesai dan pembayaran Lunas

```

2. **Buatlah class dengan nama Service dengan attribute id, jenis, harga dan status, buatkan setter dan getter untuk menambahkan object Service.**

- Pertama buat sebuah class baru dengan nama Service.
- Pada class Service, tambahkan beberapa atribut yaitu id, jenis, dan status yang bertipe String, serta atribut harga yang bertipe double. Atribut-atribut tersebut berfungsi untuk menyimpan informasi kode layanan, jenis layanan, tarif layanan, dan status ketersediaan layanan.

```
1 package Tugas;  
2  
3 public class Service {  
4     String id, jenis, status;  
5     double harga;  
6 }
```

- Selanjutnya, buat method **setter** dan **getter** untuk setiap atribut. Setter digunakan untuk memberikan nilai pada atribut seperti id, jenis, harga, dan status. Sementara itu, getter digunakan untuk mengambil dan menampilkan kembali nilai dari atribut-atribut tersebut.

```
7  
8 public String getId() {  
9     return id;  
10 }  
11  
12 public void setId(String id) {  
13     this.id = id;  
14 }  
15  
16 public String getJenis() {  
17     return jenis;  
18 }  
19  
20 public void setJenis(String jenis) {  
21     this.jenis = jenis;  
22 }  
23  
24 public double getHarga() {  
25     return harga;  
26 }  
27  
28 public void setHarga(double harga) {  
29     this.harga = harga;  
30 }  
31  
32 public String getStatus() {  
33     return status;  
34 }  
35  
36 public void setStatus(String status) {  
37     this.status = status;  
38 }
```

- Langkah terakhir adalah menambahkan method main untuk melakukan uji coba class. Di dalamnya, buat sebuah objek dari class Service dengan nama layanan. Isi nilai atribut menggunakan setter, misalnya ID layanan S001, jenis layanan Dry Clean, harga 10000.0, dan status Tersedia. Setelah itu, tampilkan informasi layanan ke layar menggunakan System.out.println() dengan memanfaatkan getter dari masing-masing atribut.

```

37 }
38
39 public static void main(String args[]) {
40     Service layanan = new Service();
41     layanan.setId("S001");
42     layanan.setJenis("Dry Clean");
43     layanan.setHarga(10000.0);
44     layanan.setStatus("Tersedia");
45
46     System.out.println("Layanan " + layanan.getJenis()
47         + " dengan kode " + layanan.getId()
48         + " tersedia dengan tarif Rp" + layanan.getHarga()
49         + " status " + layanan.getStatus());
50 }
51 }

```

- Hasilnya akan menampilkan :

```

terminated> Order (Java Application) C:\Program Files\Java\jdk-17\bin\javaw.exe. (7 Sep 2025 20:24:27 - 20:24:27 elapsed: 0:00:00.417) [pid: 17736]
Pesanan nomor 0001 dari customer C001 untuk layanan S001 dengan total biaya Rp10000.0 statusnya Selesai dan pembayaran Lunas

```

3. Buatlah class dengan nama Order dengan attribute id, id_costumer, id_service, id_user, total, tanggal, tanggal_selesai, status, status_pembayaran , buatkan setter dan getter untuk menambahkan object Order

- Buatlah sebuah class baru dengan nama **Order** di dalam package **Tugas**.
- Di dalam class **Order**, tambahkan beberapa atribut. Atribut bertipe **String** meliputi **id**, **idCustomer**, **idService**, **idUser**, **tanggal**, **tanggalSelesai**, **status**, dan **statusPembayaran**. Selain itu, tambahkan juga atribut **total** dengan tipe data **double**. Atribut-atribut ini digunakan untuk menyimpan detail informasi pesanan yang dilakukan pelanggan.

```

1 package Tugas;
2
3 public class Order {
4     String id, idCustomer, idService, idUser, tanggal, tanggalSelesai, status, statusPembayaran;
5     double total;
6 }

```

- Setelah mendefinisikan atribut, buat method **setter** dan **getter** untuk masing-masing atribut. Setter berfungsi untuk memberikan atau mengubah nilai atribut, sedangkan getter digunakan untuk mengambil nilai atribut.

```

70 public String getId() {
71     return id;
72 }
73
74 public void setId(String id) {
75     this.id = id;
76 }
77
78 public String getIdCustomer() {
79     return idCustomer;
80 }
81
82 public void setIdCustomer(String idCustomer) {
83     this.idCustomer = idCustomer;
84 }
85
86 public String getIdService() {
87     return idService;
88 }
89
90 public void setIdService(String idService) {
91     this.idService = idService;
92 }
93
94 public String getIdUser() {
95     return idUser;
96 }
97
98 public void setIdUser(String idUser) {
99     this.idUser = idUser;
100 }
101
102 public double getTotal() {
103     return total;
104 }

```

```

43 public void setTotal(double total) {
44     this.total = total;
45 }
46
47 public String getTanggal() {
48     return tanggal;
49 }
50
51 public void setTanggal(String tanggal) {
52     this.tanggal = tanggal;
53 }
54
55 public String getTanggalSelesai() {
56     return tanggalSelesai;
57 }
58
59 public void setTanggalSelesai(String tanggalSelesai) {
60     this.tanggalSelesai = tanggalSelesai;
61 }
62
63 public String getStatus() {
64     return status;
65 }
66
67 public void setStatus(String status) {
68     this.status = status;
69 }
70
71 public String getStatusPembayaran() {
72     return statusPembayaran;
73 }
74
75 public void setStatusPembayaran(String statusPembayaran) {
76     this.statusPembayaran = statusPembayaran;
77 }
78

```

- Langkah selanjutnya, menambahkan method main sebagai tempat uji coba class. Pada method ini, buat sebuah objek baru dari class **Order** dengan nama **pesanan**. Isi beberapa atribut dengan nilai contoh, seperti ID pesanan *0001*, ID customer *C001*, ID service *S001*, total biaya *10000.0*, status *Selesai*, dan status pembayaran *Lunas*. Setelah itu, gunakan **System.out.println()** untuk menampilkan informasi pesanan tersebut ke console dengan memanggil getter dari setiap atribut yang telah diisi.

```

79 public static void main(String args[]) {
80     Order pesanan = new Order();
81     pesanan.setId("0001");
82     pesanan.setIdCustomer("C001");
83     pesanan.setIdService("S001");
84     pesanan.setTotal(10000.0);
85     pesanan.setStatus("Selesai");
86     pesanan.setStatusPembayaran("Lunas");
87
88     System.out.println("Pesanan nomor " + pesanan.getId()
89         + " dari customer " + pesanan.getIdCustomer()
90         + " untuk layanan " + pesanan.getIdService()
91         + " dengan total biaya Rp" + pesanan.getTotal()
92         + " statusnya " + pesanan.getStatus()
93         + " dan pembayaran " + pesanan.getStatusPembayaran());
94 }
95 }

```

- Hasilnya akan menampilkan :

Pesanan nomor 0001 dari customer C001 untuk layanan S001 dengan total biaya Rp10000.0 statusnya Selesai dan pembayaran Lunas

E. Kesimpulan

Dengan mengintegrasikan kode program yang telah dibuat ke dalam GUI yang dirancang menggunakan Window Builder pada Eclipse IDE, saya dapat memahami konsep dasar pemrograman berorientasi objek, khususnya terkait class, object, constructor, dan method. Selain itu, saya mampu mengimplementasikan kode Java untuk membuat class dan object, serta merancang tampilan GUI sederhana sebagai penerapan konsep OOP. Dengan demikian, tujuan praktikum ini telah tercapai sesuai yang diharapkan.