

# **LAPORAN PRAKTIKUM**

## **TEKNIK PEMROGRAMAN**

PRAKTIKUM 3

OBJECT, CLASS, ENCAPSULATION

MINGGU KE-3



# **POLBAN**

NAMA: FAUZI ISMAIL

NIM: 241524042

KELAS: D4-1B

**PROGRAM STUDI SARJANA TERAPAN**

**TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI BANDUNG**

**2025**

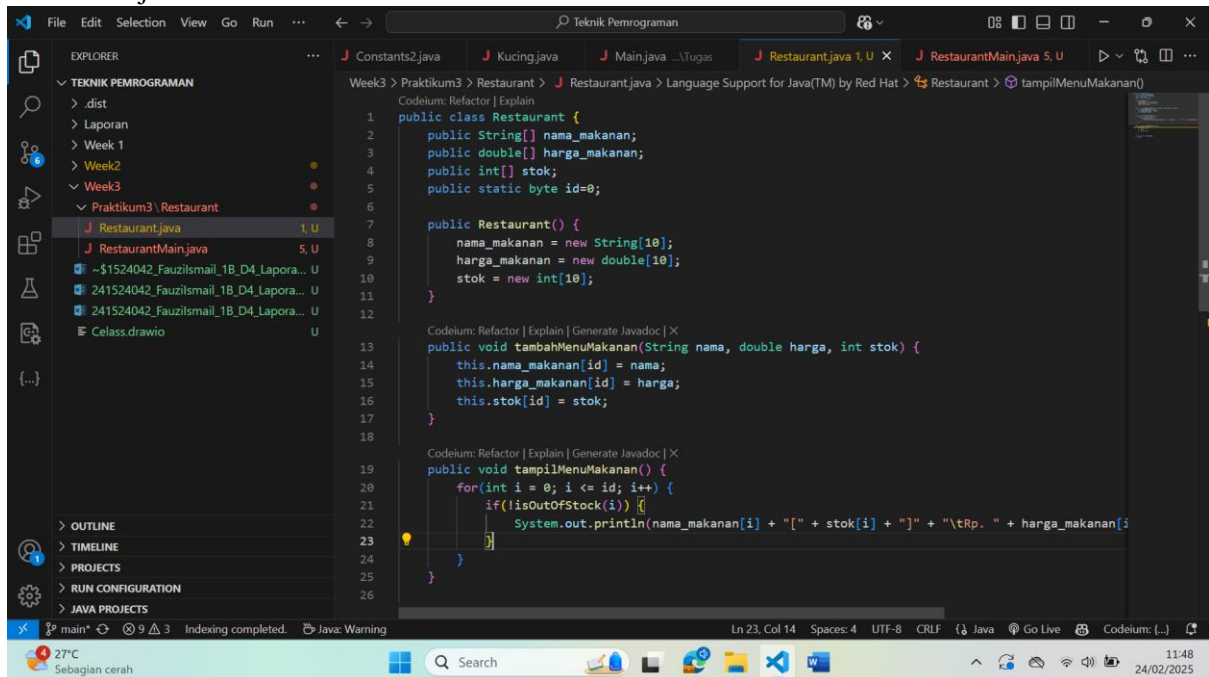
## DAFTAR ISI

DAFTAR ISI.....	2
<a href="https://github.com/mailvlous/teknikPemrograman/tree/main/Week3/Praktikum3">https://github.com/mailvlous/teknikPemrograman/tree/main/Week3/Praktikum3</a> .....	3
Kasus 1.....	3
Kasus 2.....	6
Uji Coba Kasus 2 .....	12
Link Github.....	12

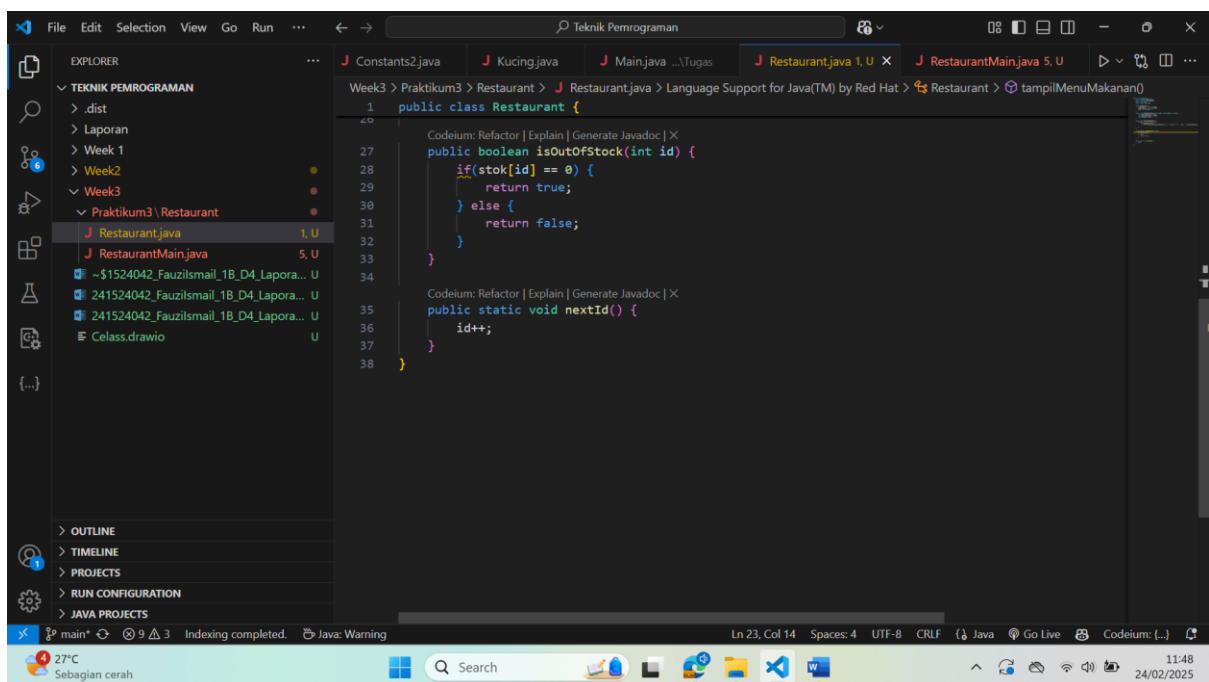
<https://github.com/mailvlous/teknikPemrograman/tree/main/Week3/Praktikum3>

## Kasus 1

Restaurant.java sebelum diubah:

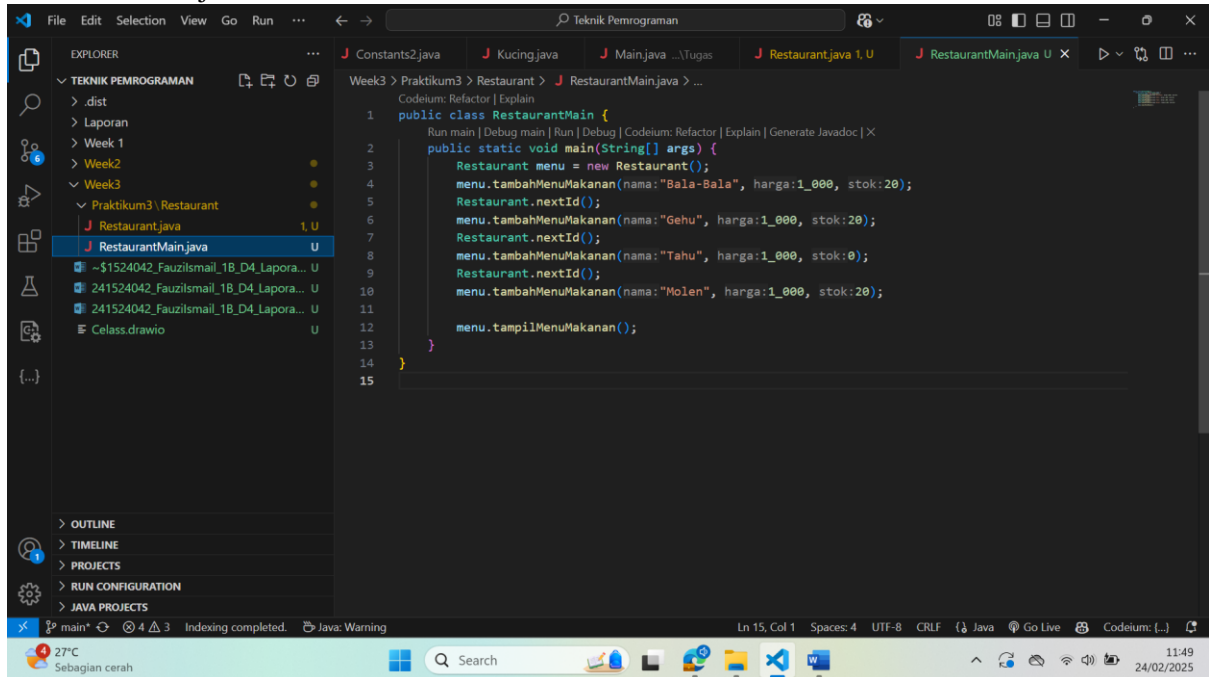


```
1 public class Restaurant {
2     public String[] nama_makanan;
3     public double[] harga_makanan;
4     public int[] stok;
5     public static byte id=0;
6
7     public Restaurant() {
8         nama_makanan = new String[10];
9         harga_makanan = new double[10];
10        stok = new int[10];
11    }
12
13    Codeium: Refactor | Explain | Generate Javadoc | X
14    public void tambahMenuMakanan(String nama, double harga, int stok) {
15        this.nama_makanan[id] = nama;
16        this.harga_makanan[id] = harga;
17        this.stok[id] = stok;
18    }
19
20    Codeium: Refactor | Explain | Generate Javadoc | X
21    public void tampilMenuMakanan() {
22        for(int i = 0; i <= id; i++) {
23            if(!isOutOfStock(i)) {
24                System.out.println(nama_makanan[i] + "[" + stok[i] + "]" + "\tRp. " + harga_makanan[i]);
25            }
26        }
27    }
```



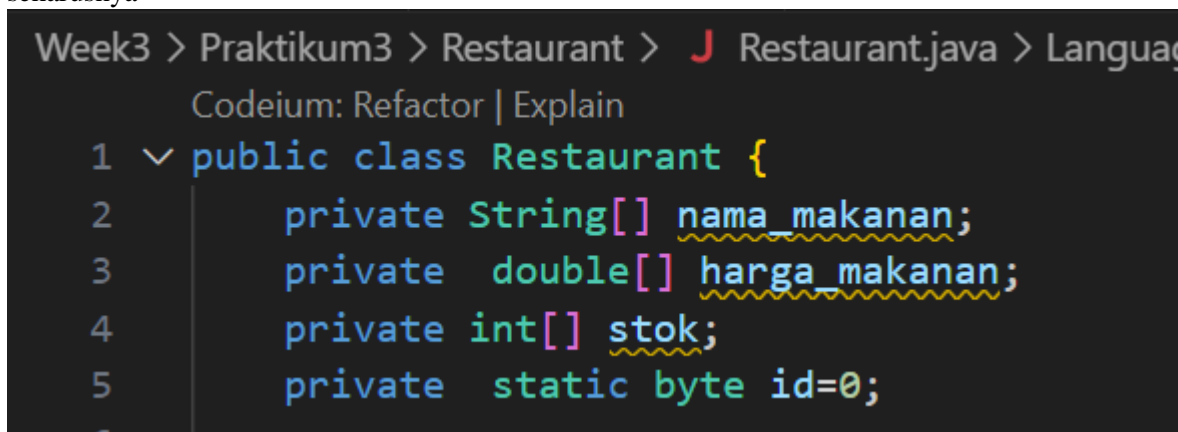
```
27 public class Restaurant {
28     public String[] nama_makanan;
29     public double[] harga_makanan;
30     public int[] stok;
31     public static byte id=0;
32
33     public Restaurant() {
34         nama_makanan = new String[10];
35         harga_makanan = new double[10];
36         stok = new int[10];
37     }
38
39     Codeium: Refactor | Explain | Generate Javadoc | X
40     public boolean isOutOfStock(int id) {
41         if(stok[id] == 0) {
42             return true;
43         } else {
44             return false;
45         }
46     }
47
48     Codeium: Refactor | Explain | Generate Javadoc | X
49     public static void nextId() {
50         id++;
51     }
52 }
```

RestautantMain.java sebelum diubah:



A. Terdapat beberapa kesalahan yang menyebabkan class Restaurant tidak memenuhi konsep OOP yang benar:

1. Tidak memenuhi konsep Encapsulation karena penginisiasian attributes masih dalam public, seharusnya



2. Sebenarnya kita dapat menyederhanakan lagi untuk nextId() sehingga tidak perlu dipanggil lagi di RestaurantMain.java. Seperti ini:

```
Codeium: Refactor | Explain | Generate Javadoc | X
17 public void tambahMenuMakanan(String nama, double harga, int stok) {
18     this.nama_makanan[id] = nama;
19     this.harga_makanan[id] = harga;
20     this.stok[id] = stok;
21     nextId();
22 }
23

Codeium: Refactor | Explain
1  v public class RestaurantMain {
    Run main | Debug main | Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
2  v public static void main(String[] args) {
3      Restaurant menu = new Restaurant();
4      menu.tambahMenuMakanan(nama:"Bala-Bala", harga:1_000, stok:20);
5
6      menu.tambahMenuMakanan(nama:"Gehu", harga:1_000, stok:20);
7
8      menu.tambahMenuMakanan(nama:"Tahu", harga:1_000, stok:0);
9
10     menu.tambahMenuMakanan(nama:"Molen", harga:1_000, stok:20);
11
12     menu.tampilMenuMakanan();
13 }
14 }
15
```

B. Untuk pengembangan aplikasi dengan menambah seperti fitur pemesanan dan juga mengurangi setiap stok yang ada sebenarnya sangat memungkinkan. Saya mungkin akan menambah class baru seperti class pemesan, dan beberapa method baru seperti pengurangan stok. Juga saya akan menambahkan method baru pada pemesan seperti memesanMakanan() yang mempunyai parameter berapa jumlah makanan yang akan dipesan. Mungkin juga saya akan menambah class menu yang mempunyai atribut nama, stok, dll, serta method getter setter.

## Kasus 2

Untuk kasus 2 saya membuat 5 class yakni:

### 1. Class Film

```
Week3 > Praktikum3 > Film > J Film.java > Language Support for Java(TM) by Red Hat > Film
Codeium: Refactor | Explain
1 public class Film {
2     private String judul;
3     private String genre;
4     private String rating;
5     private String durasi;
6
7
8     public Film(String judul, String genre, String rating, String durasi) {
9         this.judul = judul;
10        this.genre = genre;
11        this.rating = rating;
12        this.durasi = durasi;
13    }
14
15    // setter
16    public void setJudul(String judul) {
17        this.judul = judul;
18    }
19
20    public void setGenre(String genre) {
21        this.genre = genre;
22    }
23
```

```
Codeium: Refactor | Explain | Generate Javadoc | X
public void setRating(String rating) {
    this.rating = rating;
}

Codeium: Refactor | Explain | Generate Javadoc | X
public void setDurasi(String durasi) {
    this.durasi = durasi;
}
```

```
32 // getter
33 Codeium: Refactor | Explain | X
34 public String getJudul() {
35     return judul;
36 }
37 Codeium: Refactor | Explain | Generate Javadoc | X
38 public String getGenre() {
39     return genre;
40 }
41 Codeium: Refactor | Explain | Generate Javadoc | X
42 public String getRating() {
43     return rating;
44 }
45 Codeium: Refactor | Explain | Generate Javadoc | X
46 public String getDurasi() {
47     return durasi;
48 }
49 }
50
```

Class film mempunyai attributes:

- Judul
- Genre
- Rating
- Durasi

Class film mempunyai method:

- setJudul()
- setGenre()
- setRating()
- setDurasi()
- getJudul()
- getGenre()
- getRating()
- getDurasi()

## 2. Class Tiket

```
Codeium: Refactor | Explain
1 public class Tiket {
2     private String judul;
3     private int harga;
4     private int jumlah;
5
6     public Tiket(int id, String judul, int harga, int jumlah, int stok) {
7         this.judul = judul;
8         this.harga = harga;
9         this.jumlah = jumlah;
10    }
11
12    // setter
13    Codeium: Refactor | Explain | X
14    public void setJudul(Film judul) {
15        this.judul = judul.getJudul();
16    }
17
18    Codeium: Refactor | Explain | Generate Javadoc | X
19    public void setHarga(int harga) {
20        this.harga = harga;
21    }
22
23    Codeium: Refactor | Explain | Generate Javadoc | X
24    public void setJumlah(int jumlah) {
25        this.jumlah = jumlah;
26    }
27
28    // getter
29    Codeium: Refactor | Explain | X
30    public String getJudul() {
31        return judul;
32    }
33
34    Codeium: Refactor | Explain | Generate Javadoc | X
35    public int getHarga() {
36        return harga;
37    }
38
39    Codeium: Refactor | Explain | Generate Javadoc | X
40    public int getJumlah() {
41        return jumlah;
42    }
43 }
```

Class Tiket mempunyai attributes:

- judul
- harga
- jumlah

Class Tiket mempunyai method:

- setJudul()
- setHarga()
- setJumlah()
- getJudul()
- getHarga()
- getJumlah()



### 3. Class Pelanggan

```
Week3 > Praktikum3 > Film > J Pelanggan.java > Language Support for Java(TM) by Red Hat > Pelanggan
Codeium: Refactor | Explain
1 public class Pelanggan {
2     private String nama;
3     private int umur;
4     private String email;
5     private Tiket tiket;
6
7     public Pelanggan() {}
8
9     public Pelanggan(String nama, int umur, String email) {
10         this.nama = nama;
11         this.umur = umur;
12         this.email = email;
13     }
14 }
```

Dalam Class Pelanggan terdapat konstruktor kosong untuk menyimpan user input/scanner

```
15 // Setter
Codeium: Refactor | Explain | X
16 public void setName(String nama) {
17     this.nama = nama;
18 }
19
Codeium: Refactor | Explain | Generate Javadoc | X
20 public void setUmur(int umur) {
21     this.umur = umur;
22 }
23
Codeium: Refactor | Explain | Generate Javadoc | X
24 public void setEmail(String email) {
25     this.email = email;
26 }
27 }
```

```
28 // Getter
Codeium: Refactor | Explain | X
29 public String getName() {
30     return nama;
31 }
32
Codeium: Refactor | Explain | Generate Javadoc | X
33 public int getUmur() {
34     return umur;
35 }
36
Codeium: Refactor | Explain | Generate Javadoc | X
37 public String getEmail() {
38     return email;
39 }
40
Codeium: Refactor | Explain | Generate Javadoc | X
41 public Tiket getTiket() {
42     return tiket;
43 }
44
45 }
46 }
```

Dalam Class Pelanggan memiliki attributes:

- nama
- umur
- email
- tiket

Dalam Class Pelanggan memiliki methods:

- setName()
- setUmur()
- setEmail()
- getName()
- getUmur()
- getEmail()
- getTiket()

#### 4. Class Pemesanan

```
Week3 > Praktikum3 > Film > Pemesanan.java > Language Support for Java(TM) by Red Hat > Pemesanan
Codeium: Refactor | Explain
1 public class Pemesanan {
2     private Pelanggan pelanggan;
3     private Tiket tiket;
4
5     public Pemesanan(Pelanggan pelanggan, Tiket tiket) {
6         this.pelanggan = pelanggan;
7         this.tiket = tiket;
8     }
9 }
```

Terdapat 2 attribute berdasarkan Class sebelumnya, yaitu Class Pelanggan dan Class Tiket

```
9
Codeium: Refactor | Explain | Generate Javadoc | X
10 public Pelanggan getPelanggan() {
11     return pelanggan;
12 }
13
Codeium: Refactor | Explain | Generate Javadoc | X
14 public Tiket getTiket() {
15     return tiket;
16 }
17 }
```

```
Codeium: Refactor | Explain | Generate Javadoc | X
18 public void tampilPemesanan(Pelanggan pelanggan, Tiket tiket) {
19     System.out.println("\n=== Informasi Pemesanan ===");
20     System.out.println("Nama Pelanggan : " + pelanggan.getName());
21     System.out.println("Email      : " + pelanggan.getEmail());
22     System.out.println("Film       : " + tiket.getJudul());
23     System.out.println("Harga Tiket : " + tiket.getHarga());
24     System.out.println("Jumlah Tiket : " + tiket.getJumlah());
25 }
26
27 }
```

Dalam Class Pemesanan terdapat 2 attributes:

- pelanggan berdasar Class Pelanggan
- tiket berdasar Class Tiket

Dalam Class Pemesanan terdapat methods:

- getPelanggan()
- getTiket()
- tampilPemesanan()

## 5. Class Main

Class Main berfungsi untuk menampilkan dan menggunakan class-class yang tadi sudah kita buat.

```
Week3 > Praktikum3 > Film > J Main.java > Java > Main > main(String[] args)
1
2 import java.util.Scanner;
3
4
5 public class Main {
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in);
8
9         System.out.println("Masukkan nama pelanggan");
10        String nama = input.nextLine();
11        System.out.println("Masukkan email pelanggan");
12        String email = input.nextLine();
13        System.out.println("Masukkan umur pelanggan");
14        int umur = input.nextInt();
15
16        input.nextLine();
17
18        Pelanggan pelanggan = new Pelanggan(nama, umur, email);
19    }
```

Pertama kita perlu mengimport Scanner karena ada beberapa function yang memerlukan inputan dari keyboard. Lalu saya buat perintah untuk memasukkan nama pelanggan, email, dan umur. Kemudian semua itu dimasukkan kedalam Class Constructor Pelanggan.

```
20        System.out.println("Masukkan judul film");
21        String judul = input.nextLine();
22        System.out.println("Masukkan jumlah tiket");
23        int jumlah = input.nextInt();
24        System.out.println("Masukkan harga tiket");
25        int harga = input.nextInt();
26
27        Tiket tiket = new Tiket(judul, harga, jumlah);
28
29        Pemesanan pemesanan = new Pemesanan(pelanggan, tiket);
30        pemesanan.tampilPemesanan(pelanggan, tiket);
31
32
33
34    }
35 }
36
```

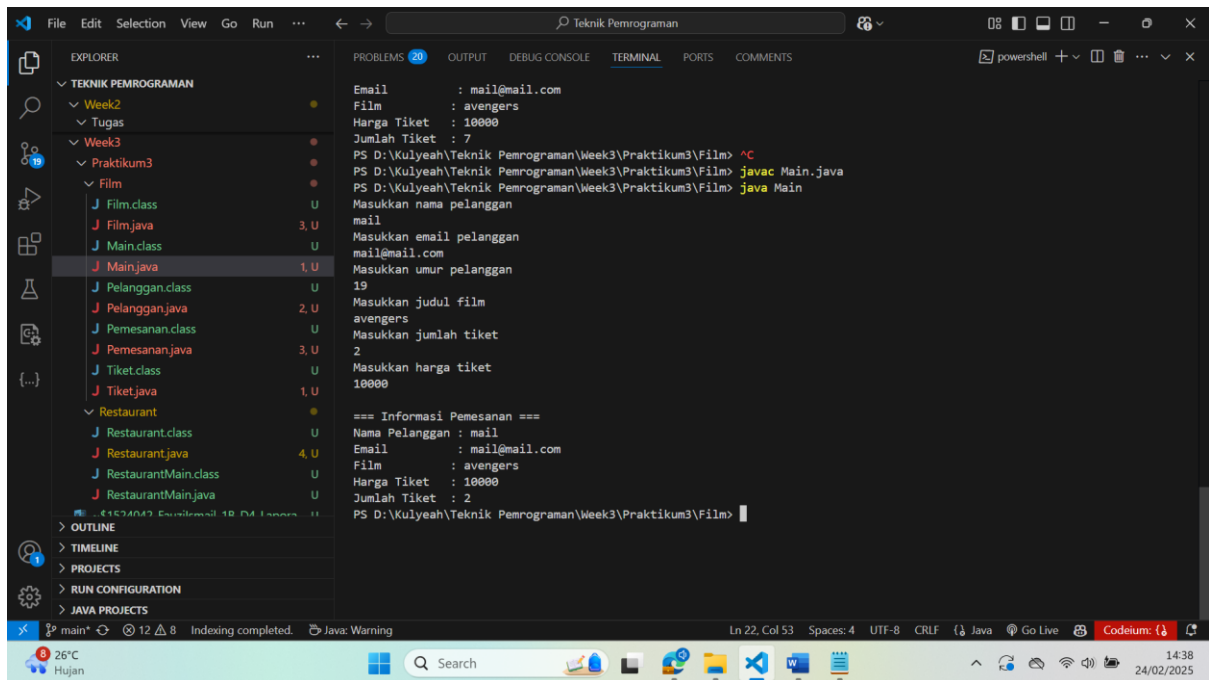
Kedua kita pengguna diminta menginput judul film, jumlah tiket, dan harga tiket. Yang kemudian data data tersebut dimasukkan kedalam Constructor Class Tiket.

Setelah itu data dari Class Tiket dan Class Pelanggan dimasukkan kedalam Class Constructor Pemesanan.

Terakhir kita tampilkan seluruh info dengan function tampilPemesanan yang ada dalam class Pemesanan.

## Uji Coba Kasus 2

Jika kita coba hasilnya seperti ini:



The screenshot shows an IDE with the following components:

- EXPLORER:** A project named 'TEKNIK PEMROGRAMAN' containing folders 'Week2', 'Tugas', and 'Week3'. Under 'Week3', there is a folder 'Praktikum3' containing files: 'Film.class', 'Film.java', 'Main.class', 'Main.java', 'Pelanggan.class', 'Pelanggan.java', 'Pemesanan.class', 'Pemesanan.java', 'Tiket.class', and 'Tiket.java'. There is also a folder 'Restaurant' containing 'Restaurant.class', 'Restaurant.java', 'RestaurantMain.class', and 'RestaurantMain.java'.
- PROBLEMS:** Shows 20 errors, mostly 'Unresolved' (U) for the class files.
- OUTPUT:** Displays the execution output of the program, showing user input and program responses.
- TERMINAL:** Shows the command prompt where the program was executed.

**Output:**

```
Email : mail@mail.com
Film : avengers
Harga Tiket : 10000
Jumlah Tiket : 7
PS D:\Kulyeah\Teknik Pemrograman\Week3\Praktikum3\Film> ^C
PS D:\Kulyeah\Teknik Pemrograman\Week3\Praktikum3\Film> javac Main.java
PS D:\Kulyeah\Teknik Pemrograman\Week3\Praktikum3\Film> java Main
Masukkan nama pelanggan
mail
Masukkan email pelanggan
mail@mail.com
Masukkan umur pelanggan
19
Masukkan judul film
avengers
Masukkan jumlah tiket
2
Masukkan harga tiket
10000

=== Informasi Pemesanan ===
Nama Pelanggan : mail
Email : mail@mail.com
Film : avengers
Harga Tiket : 10000
Jumlah Tiket : 2
PS D:\Kulyeah\Teknik Pemrograman\Week3\Praktikum3\Film>
```

Link Github

<https://github.com/mailvlous/teknikPemrograman/tree/main/Week3/Praktikum3>