

**LAPORAN PRAKTIKUM
ALGORITMA & STRUKTUR DATA
JOBSHEET 2**



ELIS NURHIDAYATI

2241720035

D-IV TEKNIK INFORMATIKA

TEKNOLOGI INFORMASI

2022

OBJEK

2.1 Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Mengetahui objek dan class sebagai konsep mendasar pada pemrograman berorientasi objek
2. Mendeklarasikan class, atribut dan method
3. Membuat objek (instansiasi)
4. Mengakses atribut dan method dari suatu objek
5. Menerapkan konstruktor

2.2 Deklarasi Class, Atribut dan Method

Waktu : 45 Menit

Perhatikan Diagram Class berikut ini:

Film
judul: String genre: String rate: String jumlahTiket: int hargaTiket: int
tampilFilm(): void tambahTiket(n: int): void kurangiTiket(n: int): void totalRevenue(jumlah: int): int

Berdasarkan diagram class di atas, akan dibuat program class dalam Java.

2.2.1 Langkah-langkah Percobaan

1. Buat Project baru, dengan nama **StrukturData**. Buat paket dengan nama minggu2 (opsional). buatlah class baru dengan nama **Film**.
2. Lengkapi class **Film** dengan atribut dan method yang telah digambarkan di dalam diagram class di atas, sebagai berikut:

```

1  public class Film {
2      String judul, genre, rate;
3      int jumlahTiket, hargaTiket;
4
5      void tampilFilm() {
6          System.out.println("Judul: " + judul);
7          System.out.println("Genre: " + genre);
8          System.out.println("Rate: " + rate);
9          System.out.println("Jumlah Tiket: " + jumlahTiket);
10         System.out.println("Harga Tiket: " + hargaTiket);
11     }
12
13     void tambahTiket(int n) {
14         jumlahTiket += n;
15     }
16
17     void kurangiTiket(int n) {
18         jumlahTiket -= n;
19     }
20
21     int totalRevenue(int jumlah) {
22         return jumlah * hargaTiket;
23     }
24
25 }

```

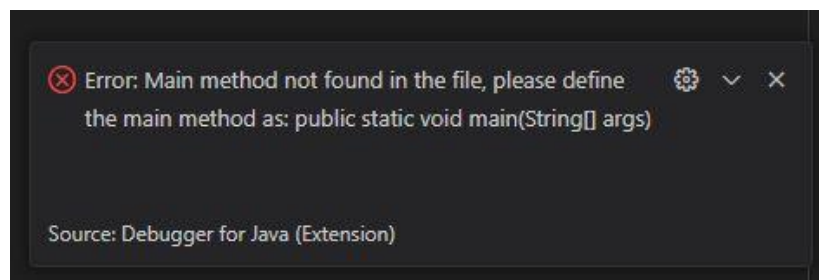
3. Coba jalankan (Run) class Barang tersebut. Apakah bisa?

Jawab : Program tidak bisa di running

2.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

Jawab : Berikut verifikasi yang muncul



2.2.3 Pertanyaan

1. Sebutkan 2 karakteristik class/objek!

Jawab : Pertama atribut yaitu data yang dimiliki oleh object atau class. Kedua method yaitu suatu proses yang bisa dilakukan oleh suatu objek.

2. Kata kunci apakah yang digunakan untuk mendeklarasikan class?

Jawab : untuk mendeklarasikan class pada program Java menggunakan kata kunci **class**

3. Perhatikan class **Film** yang ada di Praktikum di atas, ada berapa atribut yang dimiliki oleh class tersebut? Sebutkan! Dan pada baris berapa saja deklarasi atribut dilakukan?

Jawab : atribut pada class Film di atas terdapat 5 atribut, yaitu judul, genre, rate, jumlahTiket, dan hargaTiket. Deklarasi atribut tersebut berada di baris 2 dan 3.

4. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan!

Jawab : method yang dimiliki class di atas ada 4, yaitu tampilFilm():void, tambahTiket(n:int):void, kurangiTiket(n:int):void, dan totalRevenue(jumlah:int):int

5. Perhatikan method **kurangiTiket()** yang ada di class Film, modifikasi isi method tersebut sehingga proses pengurangan hanya dilakukan jika stok masih ada (masih lebih besar dari 0)

Jawab :

```
17 void kurangiTiket(int n) {  
18     if (jumlahTiket > 0) {  
19         jumlahTiket -= n;  
20     } else {  
21         System.out.println("Maaf, stok tiket sudah habis.");  
22     }  
23 }
```

6. Menurut Anda, mengapa method **tambahTiket()** dibuat dengan memiliki 1 parameter berupa bilangan int?

Jawab : Karena method **tambahTiket()** pada class Film dibuat dengan 1 parameter berupa bilangan int agar kita dapat menambahkan jumlah tiket yang sudah ada dengan jumlah tiket yang baru ditambahkan ke dalam system yang ditentukan pada method ini secara dinamis.

7. Menurut Anda, mengapa method **totalRevenue()** memiliki tipe data int?

Jawab : Karena method **totalRevenue()** pada class Film memiliki tipe data int karena method tersebut mengembalikan nilai int yang merupakan hasil kali daripada atribut jumlah tiket dengan harga tiket. Yang merupakan bilangan bulat, maka tipe data yang paling tepat untuk mewakili nilai tersebut adalah int.

8. Menurut Anda, mengapa method **tambahTiket()** memiliki tipe data void?

Jawab : Karena method **tambahTiket()** pada class Film memiliki tipe data void yang hanya digunakan untuk menambahkan jumlah tiket pada suatu film dan tidak mengembalikan nilai apapun.

2.3 Instansiasi Objek dan Mengakses Atribut & Method

Waktu : 45 Menit

Sampai tahap ini, kita telah membuat class **Film** dengan sukses. Selanjutnya, apabila diinginkan untuk mulai menggunakan class **Film** tersebut, mengakses atribut-atribut dan method-method yang ada di dalamnya, maka selanjutnya perlu dibuat objek/instance dari class **Film** terlebih dahulu.

2.3.1 Langkah-langkah Percobaan

1. Buatlah class baru dengan nama **FilmMain**. Dan di dalam class **FilmMain** tersebut, buatlah method **main()**.
2. Di dalam method **main()**, lakukan instansiasi, dan kemudian lanjutkan dengan mengakses atribut dan method dari objek yang telah terbentuk.



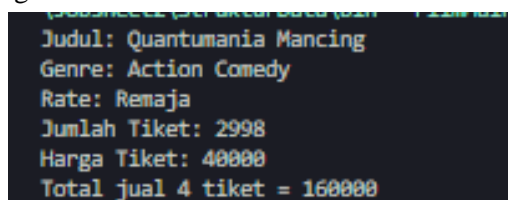
```
1 public class FilmMain {
2     public static void main(String[] args) {
3         Film film1 = new Film();
4
5         film1.judul = "Quantumania Mancing";
6         film1.genre = "Action Comedy";
7         film1.rate = "Remaja";
8         film1.jumlahTiket = 3000;
9         film1.hargaTiket = 40000;
10
11        film1.tambahTiket(1);
12        film1.kurangiTiket(3);
13        film1.tampilFilm();
14
15        int income = film1.totalRevenue(4);
16
17        System.out.println("Total jual 4 tiket = " + income);
18    }
19 }
```

3. Jalankan (Run) class **FilmMain** tersebut dan amati hasilnya.

2.3.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

Jawab : Berikut hasil running yang cocok



```
Judul: Quantumania Mancing
Genre: Action Comedy
Rate: Remaja
Jumlah Tiket: 2998
Harga Tiket: 40000
Total jual 4 tiket = 160000
```

2.3.3 Pertanyaan

1. Pada class **FilmMain**, pada kode apa yang digunakan untuk proses instansiasi? Apa nama objek yang dihasilkan?

Jawab : Pada class **FilmMain**, proses instansiasi dilakukan pada baris kode ke-3 yaitu **Film film1 = new Film()**, dan objek yang dihasilkan memiliki nama **film1**

2. Bagaimana cara mengakses atribut dan method dari suatu objek?

Jawab : Untuk mengakses atribut dan method dari suatu objek, kita harus menggunakan sintaks pemanggilan objek diikuti dengan nama atribut atau method yang ingin diakses, dan tanda titik (.) sebagai pemisah.

- 1) Mengakses atribut

```
film1.judul
```

- 2) Mengakses method

```
film1.tampilFilm();
```

2.4 Membuat Konstruktor

Waktu : 45 Menit

Di dalam percobaan ini, kita akan mempraktekkan bagaimana membuat berbagai macam konstruktor berdasarkan parameternya.

2.4.1 Langkah-langkah Percobaan

1. Perhatikan kembali class **Film**. Tambahkan di dalam class **Film** tersebut 2 buah konstruktor. 1 konstruktor default dan 1 konstruktor berparameter.

```

1 public class Film {
2     String judul, genre, rate;
3     int jumlahTiket, hargaTiket;
4
5     Film() {
6     }
7
8     Film(String jd, String gr, String rt, int jt, int ht) {
9         judul = jd;
10        genre = gr;
11        rate = rt;
12        jumlahTiket = jt;
13        hargaTiket = ht;
14    }
15
16    void tampilFilm() {
17        System.out.println("Judul: " + judul);
18        System.out.println("Genre: " + genre);
19        System.out.println("Rate: " + rate);
20        System.out.println("Jumlah Tiket: " + jumlahTiket);
21        System.out.println("Harga Tiket: " + hargaTiket);
22    }
23
24    void tambahTiket(int n) {
25        jumlahTiket += n;
26    }
27
28    void kurangiTiket(int n) {
29        if (jumlahTiket > 0) {
30            jumlahTiket -= n;
31        } else {
32            System.out.println("Maaf, stok tiket sudah habis.");
33        }
34    }
35
36    int totalRevenue(int jumlah) {
37        return jumlah * hargaTiket;
38    }

```

2. Buka kembali class **FilmMain**. Dan buat sebuah objek lagi, kali ini dengan menggunakan konstruktor berparameter.

```

1  public class FilmMain {
2      public static void main(String[] args) {
3          Film film1 = new Film();
4
5          film1.judul = "Quantumania Mancing";
6          film1.genre = "Action Comedy";
7          film1.rate = "Remaja";
8          film1.jumlahTiket = 3000;
9          film1.hargaTiket = 40000;
10
11         film1.tambahTiket(1);
12         film1.kurangiTiket(3);
13         film1.tampilFilm();
14
15         int income = film1.totalRevenue(4);
16
17         System.out.println("Total jual 4 tiket = " + income);
18         System.out.println("=====\n");
19         Film film2 = new Film("Maniaquantum", "Horor", "Dewasa", 2000, 40000);
20         film2.tampilFilm();
21     }
22 }

```

3. Jalankan kembali class **FilmMain** dan amati hasilnya.

2.4.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

Jawab : berikut hasil running program yang cocok

```

Judul: Quantumania Mancing
Genre: Action Comedy
Rate: Remaja
Jumlah Tiket: 2998
Harga Tiket: 40000
Total jual 4 tiket = 160000
=====

Judul: Maniaquantum
Genre: Horror
Rate: Dewasa
Jumlah Tiket: 2000
Harga Tiket: 40000

```

2.4.3 Pertanyaan

1. Perhatikan class **Film** yang ada di Praktikum 2.4.1, pada baris berapakah deklarasi konstruktor berparameter dilakukan?

Jawab : Deklarasi konstruktornya dilakukan pada baris 5 - 14

2. Perhatikan class **FilmMain** di Praktikum 2.4.1, apa sebenarnya yang dilakukan pada baris program dibawah ini?


```
Film film2 = new Film("Maniaquantum", "Horor", "Dewasa", 2000, 40000);
```

Jawab : Membuat objek baru dengan nama film2 dan memanggil konstruktor class Film dengan argumen sesuai gambar di atas yang akan menginisialisasi nilai-nilai awal atribut judul, genre, rate, jumlahTiket, dan hargaTiket pada objek film2.

3. Coba buat objek dengan nama **film3** dengan menggunakan konstruktor berparameter dari class **FilmMain**.

Jawab :

```
21 System.out.println("=====\n");
22 Film film3 = new Film(jd: "Alice in Wonderland", gr: "Drama Fantasi", rt: "Remaja", jt: 5000, ht: 45000);
23 film3.tampilFilm();
24
```

```
=====
Judul: Alice in Wonderland
Genre: Drama Fantasi
Rate: Remaja
Jumlah Tiket: 5000
Harga Tiket: 45000
PS C:\polinema\Jobsheet2\StrukturData>
```

2.5 Latihan Praktikum

Waktu : 60 Menit

1. Buat program berdasarkan diagram class berikut ini!

Buku
nama: String pengarang: String penerbit: String hargaSatuan: int jumlah: int
hitungHargaTotal(): int hitungDiskon(): int hitungHargaBayar(): int

- Method `hitungHargaTotal()` digunakan untuk menghitung harga total yang merupakan perkalian antara `hargaSatuan` dengan `jumlah` barang yang dibeli
- Method `hitungDiskon()` digunakan untuk menghitung diskon dengan aturan sbb:
 - Jika harga total > 100000, akan mendapat diskon 10%
 - Jika harga total mulai dari 50000 sampai 100000 akan mendapat diskon sebesar 5%
 - Jika dibawah 50000 tidak mendapat diskon
- Method `hitungHargaBayar()` digunakan untuk menghitung harga total setelah dikurangi diskon

Kode program :

class Buku :

```
1 public class Buku {
2     String nama, pengarang, penerbit;
3     int hargaSatuan, jumlah;
4
5     Buku() {
6     }
7
8     Buku(String nm, String pgr, String pnb, int hs, int jml) {
9         nama = nm;
10        pengarang = pgr;
11        penerbit = pnb;
12        hargaSatuan = hs;
13        jumlah = jml;
14    }
15
16    int hitungHargaTotal() {
17        return hargaSatuan * jumlah;
18    }
19
20    int hitungDiskon() {
21        int hargaTotal = hitungHargaTotal();
22        int diskon = 0;
23        if (hargaTotal > 100000) {
24            diskon = (int) (hargaTotal * 10 / 100);
25        } else if (hargaTotal >= 50000) {
26            diskon = (int) (hargaTotal * 5 / 100);
27        }
28        return diskon;
29    }
30
31    int hitungHargaBayar() {
32        return hitungHargaTotal() - hitungDiskon();
33    }
34
35    void tampilBuku() {
36        System.out.println("Nama Buku: " + nama);
37        System.out.println("Pengarang: " + pengarang);
38        System.out.println("Penerbit: " + penerbit);
39        System.out.println("Harga Satuan: " + hargaSatuan);
40        System.out.println("Jumlah: " + jumlah);
41        System.out.println("Harga Total: " + hitungHargaTotal());
42        System.out.println("Diskon: " + hitungDiskon());
43        System.out.println("Total Bayar: " + hitungHargaBayar());
44    }
45 }
```

class BukuMain :

```
1  public class BukuMain {
2      public static void main(String[] args) {
3          Buku buku1 = new Buku("Harry Potter", "J.K. Rowling", "Gramedia", 50000, 3);
4          buku1.tampilBuku();
5          System.out.println("=====\n");
6
7          Buku buku2 = new Buku();
8          buku2.nama = "Demian";
9          buku2.pengarang = "Hermann Hesse";
10         buku2.penerbit = "Semicolon Publisher";
11         buku2.hargaSatuan = 160000;
12         buku2.jumlah = 1;
13         buku2.tampilBuku();
14     }
15 }
```

Hasil running :

```
Nama Buku: Harry Potter
Pengarang: J.K. Rowling
Penerbit: Gramedia
Harga Satuan: 50000
Jumlah: 3
Harga Total: 150000
Diskon: 15000
Total Bayar: 135000
=====
Nama Buku: Demian
Pengarang: Hermann Hesse
Penerbit: Semicolon Publisher
Harga Satuan: 160000
Jumlah: 1
Harga Total: 160000
Diskon: 16000
Total Bayar: 144000
```

2. Buat program berdasarkan diagram class berikut ini!

Snake
x: int y: int width: int height: int
moveLeft(): void moveRight(): void moveUp(): void moveDown(): void printPosition(): void detectCollision(x: int, y: int): void

- Atribut **x** digunakan untuk menyimpan posisi koordinat x (mendatar) dari snake, sedangkan atribut **y** untuk posisi koordinat y (vertikal)
- Atribut **width** digunakan untuk menyimpan lebar dari area permainan, sedangkan **height** untuk menyimpan panjang area
- Method **moveLeft()** digunakan untuk mengubah posisi snake ke kiri (koordinat x akan berkurang 1), sedangkan **moveRight()** untuk bergerak ke kanan (koordinat x akan bertambah 1). Perlu diperhatikan bahwa koordinat x tidak boleh lebih kecil dari 0 atau lebih besar dari nilai width. Jika koordinat $x < 0$ atau $x > \text{width}$ maka panggil method **detectCollision()**
- Method **moveUp()** digunakan untuk mengubah posisi snake ke atas (koordinat y akan berkurang 1), sedangkan **moveDown()** untuk bergerak ke bawah (koordinat y akan bertambah 1). Perlu diperhatikan bahwa koordinat y tidak boleh lebih kecil dari 0 atau lebih besar dari nilai height. Jika koordinat $y < 0$ atau $y > \text{height}$ maka panggil method **detectCollision()**
- Method **detectCollision()** akan mencetak pesan “Game Over” apabila snake menyentuh ujung area permainan.

Kode program :

class Snake :

```

1  public class Snake {
2      int x, y, width, height;
3
4      Snake() {
5
6      }
7
8      Snake(int xx, int yy, int w, int h) {
9          x = xx;
10         y = yy;
11         width = w;
12         height = h;
13     }
14
15     void moveLeft() {
16         if (x > 1) {
17             x--;
18             printPosition();
19         } else {
20             detectCollision(x, y);
21         }
22     }
23
24     void moveRight() {
25         if (x < width - 1) {
26             x++;
27             printPosition();
28         } else {
29             detectCollision(x, y);
30         }
31     }
32
33     void moveUp() {
34         if (y > 1) {
35             y--;
36             printPosition();
37         } else {
38             detectCollision(x, y);
39         }
40     }
41
42     void moveDown() {
43         if (y < height - 1) {
44             y++;
45             printPosition();
46         } else {
47             detectCollision(x, y);
48         }
49     }
50
51     void detectCollision(int x, int y) {
52         System.out.println("-----GAME OVER-----");
53     }
54
55     void printPosition() {
56         for (int i = 0; i <= height; i++) {
57             for (int j = 0; j <= width; j++) {
58                 if (i == 0 || i == height || j == 0 || j == width) {
59                     System.out.print("E ");
60                 } else if (i == y && j == x) {
61                     System.out.print("O ");
62                 } else {
63                     System.out.print(" ");
64                 }
65             }
66             System.out.println();
67         }
68     }
69 }

```

```
class SnakeMain :
```



```
1  import java.util.Scanner;
2
3  public class SnakeMain {
4
5      public static void main(String[] args) {
6          Snake snke = new Snake(1, 1, 10, 10);
7          Scanner sc = new Scanner(System.in);
8
9          System.out.println("=====");
10         System.out.println("Aturan Bermain");
11         System.out.println("Tombol a untuk mengarahkan ke kiri");
12         System.out.println("Tombol d untuk mengarahkan ke kanan");
13         System.out.println("Tombol w untuk mengarahkan ke atas");
14         System.out.println("Tombol s untuk mengarahkan ke bawah");
15         System.out.println("ketik exit untuk keluar");
16         System.out.println("=====");
17
18         snke.printPosition();
19         loop: while (true) {
20             switch (sc.nextLine()) {
21                 case "a":
22                     snke.moveLeft();
23                     break;
24
25                 case "d":
26                     snke.moveRight();
27                     break;
28
29                 case "w":
30                     snke.moveUp();
31                     break;
32
33                 case "s":
34                     snke.moveDown();
35                     break;
36
37                 case "exit":
38                     snke.detectCollision(-1, -1);
39                     break loop;
40
41                 default:
42                     snke.printPosition();
43             }
44         }
45     }
46 }
```


Hasil running :

```
Tombol a untuk mengarahkan ke kiri
Tombol d untuk mengarahkan ke kanan
Tombol w untuk mengarahkan ke atas
Tombol s untuk mengarahkan ke bawah
ketik exit untuk keluar
=====
E E E E E E E E E E
E O                               E
E                               E
E                               E
E                               E
E                               E
E                               E
E                               E
E                               E
E                               E
E E E E E E E E E E
s
E E E E E E E E E E
E                               E
E O                               E
E                               E
E                               E
E                               E
E                               E
E                               E
E                               E
E E E E E E E E E E
d
E E E E E E E E E E
E                               E
E O                               E
E                               E
E                               E
E                               E
E                               E
E                               E
E                               E
E E E E E E E E E E
w
E E E E E E E E E E
E O                               E
E                               E
E                               E
E                               E
E                               E
E                               E
E                               E
E E E E E E E E E E
w
-----GAME OVER-----
```