



JOBSHEET II

OBJEK

2.1 Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Mengenal 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(): 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:

```
public class Film {
    String judul, genre, rate;
    int jumlahTiket, hargaTiket;

    void tampilFilm() {
        System.out.println("Judul: "+judul);
        System.out.println("Genre: "+genre);
        System.out.println("Rate: "+rate);
        System.out.println("Jumlah Tiket: "+jumlahTiket);
        System.out.println("Harga Tiket: "+hargaTiket);
    }

    void tambahTiket(int n) {
        jumlahTiket += n;
    }

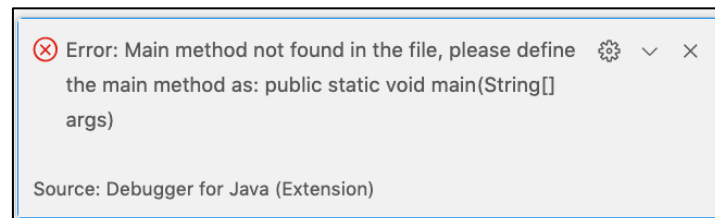
    void kurangiTiket(int n) {
        jumlahTiket -= n;
    }

    int totalRevenue() {
        return jumlahTiket*hargaTiket;
    }
}
```

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

2.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.



2.2.3 Pertanyaan

1. Sebutkan 2 karakteristik class/objek!
2. Kata kunci apakah yang digunakan untuk mendeklarasikan 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?
4. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan!
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)
6. Menurut Anda, mengapa method `tambahTiket()` dibuat dengan memiliki 1 parameter berupa bilangan int?



7. Menurut Anda, mengapa method `totalRevenue()` memiliki tipe data `int`?
8. Menurut Anda, mengapa method `tambahTiket()` memiliki tipe data `void`?

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.

```
public class FilmMain {
    public static void main(String[] args) {
        Film film1 = new Film();

        film1.judul = "Quantumania Mancing";
        film1.genre = "Action Comedy";
        film1.rate = "Remaja";
        film1.jumlahTiket = 3000;
        film1.hargaTiket = 40000;

        film1.tambahTiket(1);
        film1.kurangiTiket(3);
        film1.tampilFilm();

        int income = film1.totalRevenue();

        System.out.println("Total keuntungan tiket = "+income);
    }
}
```

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.

```
Judul: Quantumania Mancing
Genre: Action Comedy
Rate: Remaja
Jumlah Tiket: 2998
Harga Tiket: 40000
Total keuntungan tiket = 119920000
```



2.3.3 Pertanyaan

1. Pada class **FilmMain**, pada kode apa yang digunakan untuk proses instansiasi? Apa nama objek yang dihasilkan?
2. Bagaimana cara mengakses atribut dan method dari suatu objek?

2.4 Membuat Konstruktork

Waktu : 45 Menit

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

2.4.1 Langkah-langkah Percobaan

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

```
public class Film {
    String judul, genre, rate;
    int jumlahTiket, hargaTiket;

    Film() {
    }

    Film(String jd, String gr, String rt, int jt, int ht) {
        judul = jd;
        genre = gr;
        rate = rt;
        jumlahTiket = jt;
        hargaTiket = ht;
    }

    void tampilFilm() {
        System.out.println("Judul: "+judul);
        System.out.println("Genre: "+genre);
        System.out.println("Rate: "+rate);
        System.out.println("Jumlah Tiket: "+jumlahTiket);
        System.out.println("Harga Tiket: "+hargaTiket);
    }

    void tambahTiket(int n) {
        jumlahTiket += n;
    }

    void kurangiTiket(int n) {
        jumlahTiket -= n;
    }

    int totalRevenue() {
        return jumlahTiket*hargaTiket;
    }
}
```

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

```
public class FilmMain {
    public static void main(String[] args) {
        Film film1 = new Film();

        film1.judul = "Quantumania Mancing";
        film1.genre = "Action Comedy";
        film1.rate = "Remaja";
        film1.jumlahTiket = 3000;
        film1.hargaTiket = 40000;

        film1.tambahTiket(1);
        film1.kurangiTiket(3);
        film1.tampilFilm();

        int income = film1.totalRevenue();

        System.out.println("Total keuntungan tiket = "+income);

        System.out.println("=====\n");

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

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

2.4.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
Judul: Quantumania Mancing
Genre: Action Comedy
Rate: Remaja
Jumlah Tiket: 2998
Harga Tiket: 40000
Total keuntungan tiket = 119920000
=====

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?
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);
```

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

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 buku 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
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.