



JOBSHEET II

OBJEK DAN ARRAY OF OBJECT

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
6. Memahami dan menjelaskan fungsi array yang berisikan variabel objek.
7. Mahasiswa mampu menangkap logika tentang permasalahan array of object dalam Java
8. Mahasiswa mampu menerapkan pembuatan array of object dalam Java

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:

```
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;
    }

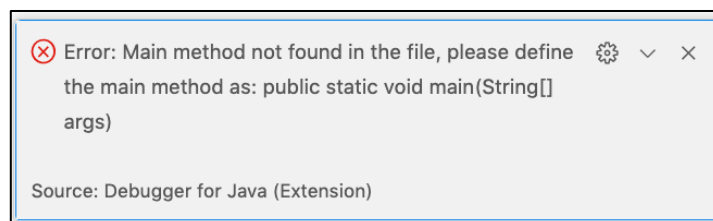
    void kurangiTiket(int n) {
        jumlahTiket -= n;
    }

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

3. Coba jalankan (Run) class Barang 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(4);

        System.out.println("Total jual 4 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 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?
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(int jumlah) {
        return jumlah*hargaTiket;
    }
}
```

2. Buka kembali class **FilmMain**. Dan buat sebuah objek lagi, kali ini dengan menggunakan konstruktor 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(4);

        System.out.println("Total jual 4 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 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?
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 **Barang**.

2.5 Membuat Array dari Object, Mengisi dan Menampilkan

Didalam praktikum ini, kita akan mempraktekkan bagaimana membuat array dari object, kemudian mengisi dan menampilkan array tersebut.

2.5.1 Langkah-langkah Percobaan

1. Buat Project baru, dengan nama "ArrayObjects".
2. Buat class **PersegiPanjang**:

```
public class PersegiPanjang {
    public int panjang;
    public int lebar;
}
```

3. Pada fungsi main yaitu pada class **ArrayObjects**, buatlah array **PersegiPanjang** yang berisi 3 elemen:

```
public static void main(String[] args) {
    PersegiPanjang[] ppArray = new PersegiPanjang[3];
}
```

4. Kemudian isikan masing-masing atributnya:

```
ppArray[0] = new PersegiPanjang();
ppArray[0].panjang = 110;
ppArray[0].lebar = 30;

ppArray[1] = new PersegiPanjang();
ppArray[1].panjang = 80;
ppArray[1].lebar = 40;

ppArray[2] = new PersegiPanjang();
ppArray[2].panjang = 100;
ppArray[2].lebar = 20;
```



5. Cetak ke layar semua atribut dari objek `ppArray`:

```
System.out.println("Persegi Panjang ke-0, panjang: " + ppArray[0].panjang + ", lebar: " + ppArray[0].lebar);
System.out.println("Persegi Panjang ke-1, panjang: " + ppArray[1].panjang + ", lebar: " + ppArray[1].lebar);
System.out.println("Persegi Panjang ke-2, panjang: " + ppArray[2].panjang + ", lebar: " + ppArray[2].lebar);
```

6. Jalankan dan amati hasilnya.

2.5.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
run:
Persegi Panjang ke-0, panjang: 110, lebar: 30
Persegi Panjang ke-1, panjang: 80, lebar: 40
Persegi Panjang ke-2, panjang: 100, lebar: 20
BUILD SUCCESSFUL (total time: 0 seconds)
```

2.5.3 Pertanyaan

1. Berdasarkan uji coba 3.2, apakah class yang akan dibuat array of object harus selalu memiliki atribut dan sekaligus method?Jelaskan!
2. Apakah class `PersegiPanjang` memiliki konstruktor?Jika tidak, kenapa dilakukan pemanggilan konstruktor pada baris program berikut :

```
ppArray[1] = new PersegiPanjang();
```

3. Apa yang dimaksud dengan kode berikut ini:

```
PersegiPanjang[] ppArray = new PersegiPanjang[3];
```

4. Apa yang dimaksud dengan kode berikut ini:

```
ppArray[1] = new PersegiPanjang();
ppArray[1].panjang = 80;
ppArray[1].lebar = 40;
```

5. Mengapa class main dan juga class `PersegiPanjang` dipisahkan pada uji coba 3.2?

2.6 Menerima Input Isian Array Menggunakan Looping

Pada praktikum ini kita akan mengubah hasil program dari praktikum 3.2 sehingga program dapat menerima input dan menggunakan looping untuk mengisi atribut dari semua persegi panjang yang ada di `ppArray`.



2.6.1 Langkah-langkah Percobaan

1. Import scanner pada class **ArrayObjects**.

```
import java.util.Scanner;
```

Note: Letakkan kode import dibawah kode package.

2. Pada praktikum 3.2 poin nomor 4, ganti kodenya dengan kode berikut ini, yaitu membuat objek **Scanner** untuk menerima input, kemudian melakukan looping untuk menerima input:

```
Scanner sc = new Scanner(System.in);

for(int i = 0; i < 3; i++)
{
    ppArray[i] = new PersegiPanjang();
    System.out.println("Persegi panjang ke-" + i);
    System.out.print("Masukkan panjang: ");
    ppArray[i].panjang = sc.nextInt();
    System.out.print("Masukkan lebar: ");
    ppArray[i].lebar = sc.nextInt();
}
```

3. Pada praktikum 3.2 poin nomor 5, ganti kodenya dengan berikut ini, yaitu melakukan looping untuk mengakses isi array **ppArray** dan menampilkannya ke layar:

```
for(int i = 0; i < 3; i++)
{
    System.out.println("Persegi Panjang ke-" + i);
    System.out.println("Panjang: " + ppArray[i].panjang + ", lebar: " + ppArray[i].lebar);
}
```

4. Jalankan dan amati hasilnya.

2.6.2 Verifikasi Hasil Percobaan

Contoh verifikasi hasil percobaan ini.

```
Persegi panjang ke-0
Masukkan panjang: 5
Masukkan lebar: 6
Persegi panjang ke-1
Masukkan panjang: 5
Masukkan lebar: 3
Persegi panjang ke-2
Masukkan panjang: 4
Masukkan lebar: 8
Persegi Panjang ke-0
Panjang: 5, lebar: 6
Persegi Panjang ke-1
Panjang: 5, lebar: 3
Persegi Panjang ke-2
Panjang: 4, lebar: 8
```

2.6.3 Pertanyaan

1. Apakah array of object dapat diimplementasikan pada array 2 Dimensi?
2. Jika jawaban soal no satu iya, berikan contohnya! Jika tidak, jelaskan!



3. Jika diketahui terdapat class **Persegi** yang memiliki atribut sisi bertipe integer, maka kode dibawah ini akan memunculkan error saat dijalankan. Mengapa?

```
Persegi[] pgArray = new Persegi[100];
pgArray[5].sisi = 20;
```

4. Modifikasi kode program pada praktikum 3.3 agar length array menjadi inputan dengan Scanner!
5. Apakah boleh Jika terjadi duplikasi instansiasi array of objek, misalkan saja instansiasi dilakukan pada **ppArray[i]** sekaligus **ppArray[0]**?Jelaskan !

2.7 Operasi Matematika Atribut Object Array

Pada praktikum ini kita akan melakukan pengoperasian matematika beberapa atribut pada masing-masing anggota array.

2.7.1 Langkah-langkah Percobaan

1. Buat package baru “**ArrayBalok**”.
2. Buat class **Balok**:

```
public class Balok {
    public int panjang;
    public int lebar;
    public int tinggi;

    public Balok(int p, int l, int t)
    {
        panjang = p;
        lebar = l;
        tinggi = t;
    }

    public int hitungVolume()
    {
        return panjang * lebar * tinggi;
    }
}
```

3. Pada fungsi **main** yaitu pada class **ArrayBalok**, buat array **Balok** yang berisi 3 elemen:

```
public static void main(String[] args) {
    Balok[] blArray = new Balok[3];
}
```

4. Kemudian tambahkan kode berikut ini untuk mengisi array **blArray** menggunakan konstruktor dari class **Balok**:

```
blArray[0] = new Balok(100, 30, 12);
blArray[1] = new Balok(120, 40, 15);
blArray[2] = new Balok(210, 50, 25);
```

5. Tampilkan semua volume balok tersebut dengan cara memanggil method **hitungVolume()** di dalam looping seperti berikut ini:



```
for(int i = 0; i < 3; i++)
{
    System.out.println("Volume balok ke " + i + ": " + blArray[i].hitungVolume());
}
```

6. Jalankan dan amati hasilnya.

2.7.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
run:
Volume balok ke 0: 36000
Volume balok ke 1: 72000
Volume balok ke 2: 262500
```

2.7.3 Pertanyaan

1. Dapatkah konstruktor berjumlah lebih dalam satu kelas? Jelaskan dengan contoh!
2. Jika diketahui terdapat class **Segitiga** seperti berikut ini:

```
public class Segitiga {
    public int alas;
    public int tinggi;
}
```

Tambahkan konstruktor pada class **Segitiga** tersebut yang berisi parameter **int a, int t** yang masing-masing digunakan untuk mengisi atribut alas dan tinggi.

3. Tambahkan method **hitungLuas()** dan **hitungKeliling()** pada class **Segitiga** tersebut. **Asumsi segitiga adalah segitiga siku-siku.** (*Hint: Anda dapat menggunakan bantuan library Math pada Java untuk mengkalkulasi sisi miring*)
4. Pada fungsi **main**, buat array **Segitiga sgArray** yang berisi 4 elemen, isikan masing-masing atributnya sebagai berikut:

sgArray ke-0	alas: 10, tinggi: 4
sgArray ke-1	alas: 20, tinggi: 10
sgArray ke-2	alas: 15, tinggi: 6
sgArray ke-3	alas: 25, tinggi: 10

Kemudian menggunakan looping, cetak luas dan keliling dengan cara memanggil method **hitungLuas()** dan **hitungKeliling()**.



2.8 Tugas Praktikum

Waktu : 60 Menit

1. Buat program berdasarkan diagram class berikut ini!

Nasabah
Id: int nama: String alamat: String noHP: int noRek: int saldo: int jmlHutang: float statusAktif: boolean
lihatSaldo(): int menabung(deposit: int): int tarikTunai(jmlTarik: int): int bukaRekening(): void tutupRekening(): void berhutang(jmlHutang: int): void

- Method lihatSaldo() digunakan untuk melihat kondisi saldo tabungan
 - Method menabung() digunakan untuk menambah saldo sesuai nominal parameter deposit yang dimasukkan
 - Method tarikTunai() digunakan untuk mengurangi saldo sejumlah parameter jmlTarik
 - Method bukaRekening() digunakan nasabah untuk pertama kali. Jika nasabah belum memiliki rekening, maka statusAktif false dan tidak boleh melakukan aktifitas lihatSaldo(), menabung(), tarikTunai(), tutupRekening(), dan berhutang().
 - Method tutupRekening() digunakan untuk menonaktifkan rekening sehingga nasabah tidak lagi dapat melakukan aktifitas lihatSaldo(), menabung(), tarikTunai(), tutupRekening(), dan berhutang().
 - Method berhutang() digunakan untuk mengajukan pinjaman dengan aturan berikut:
 - Jika jmlHutang >= saldo, akan muncul peringatanajuan peminjaman ditolak
 - Jika jmlHutang <= saldo, pinjaman disetujui dan dimunculkan simulasi skema cicilan per bulan
 - Seluruh cicilan harus dihitung dalam waktu 6 bulan
 - Simulasi skema cicilan per bulan dihitung dari jumlah hutang di rekening/6. Jangan lupa menggunakan konversi tipe data int ke float.
2. Berdasarkan soal nomor 1, terapkan kasus lebih dari 1 nasabah menggunakan object of array!
 3. Berdasarkan soal nomor 1 dan 2, terapkan kasus 1 nasabah dapat memiliki lebih dari 1 rekening menggunakan array biasa!