JOBSHEET II

**OBJEK DAN ARRAY OF OBJECT**

# 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

# Deklarasi Class, Atribut dan Method

Waktu : 45 Menit

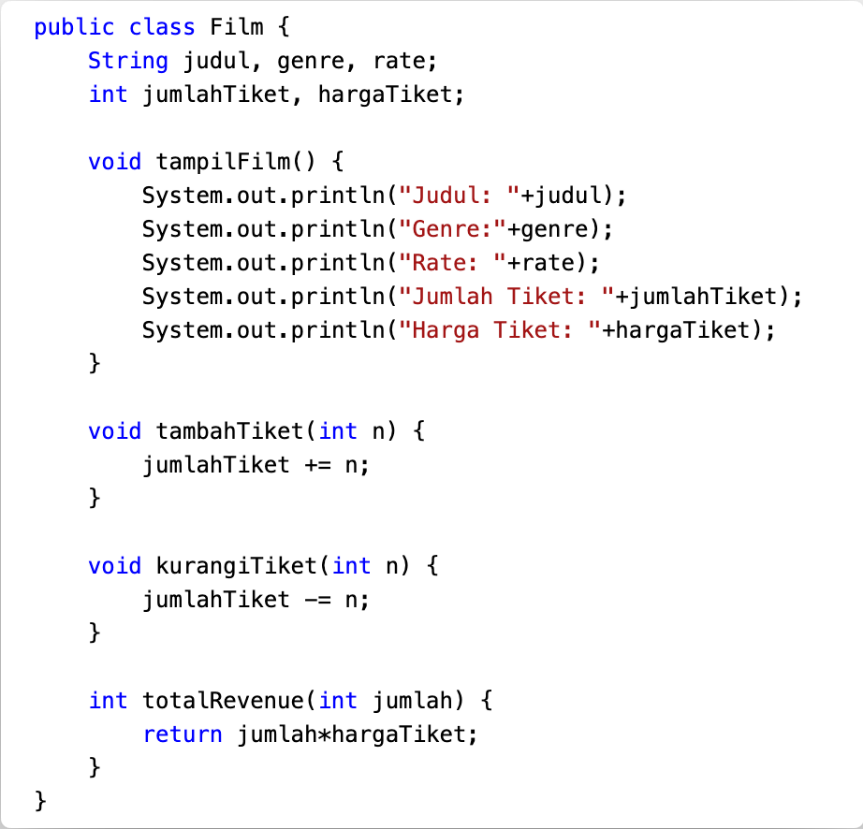
Perhatikan Diagram Class berikut ini:

|  |
| --- |
| 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.

## 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:

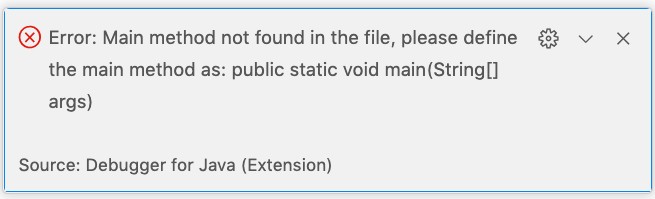


1. Coba jalankan (Run) class Barang tersebut. Apakah bisa?



## Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.



## 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?

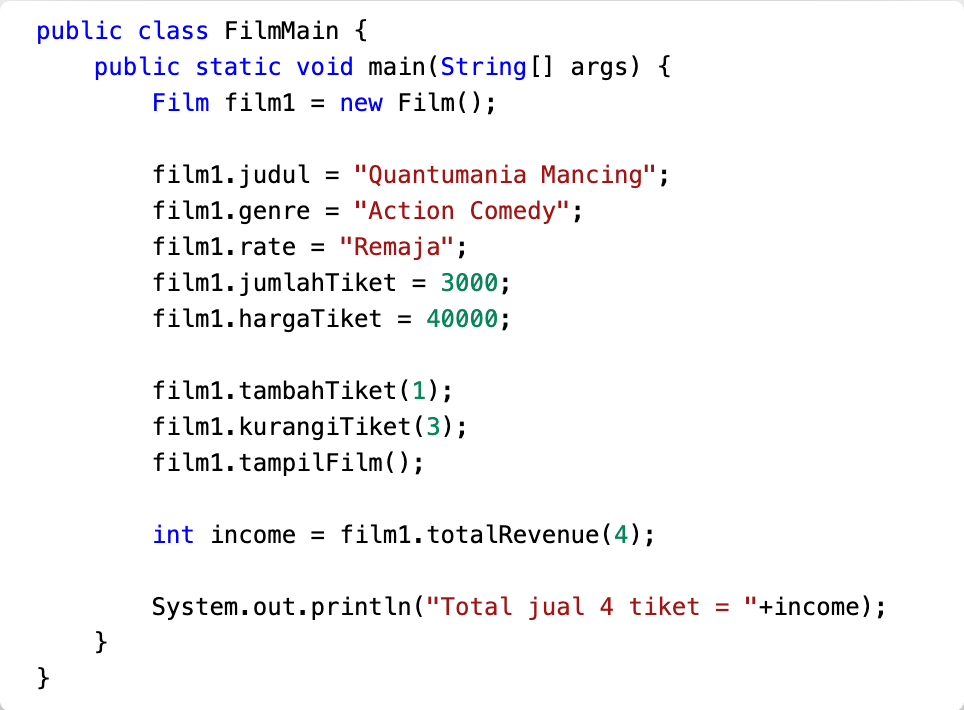
# Instansiasi Objek dan Mengakses Atribut & Method

Waktu : 45 Menit

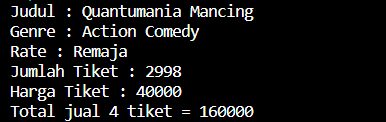
Sampai tahap ini, kita telah membuat class Film dengan sukses. Selanjutnya, apabila diinginkan untuk mulai menggunakan class Film tersebut, mengakses atribut-atribut dan method-method yang ada di dalamnya, maka selanjutnya perlu dibuat objek/instance dari class Film terlebih dahulu.

## 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.

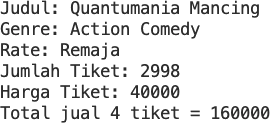


1. Jalankan (Run) class **FilmMain** tersebut dan amati hasilnya.



## Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.



## Pertanyaan

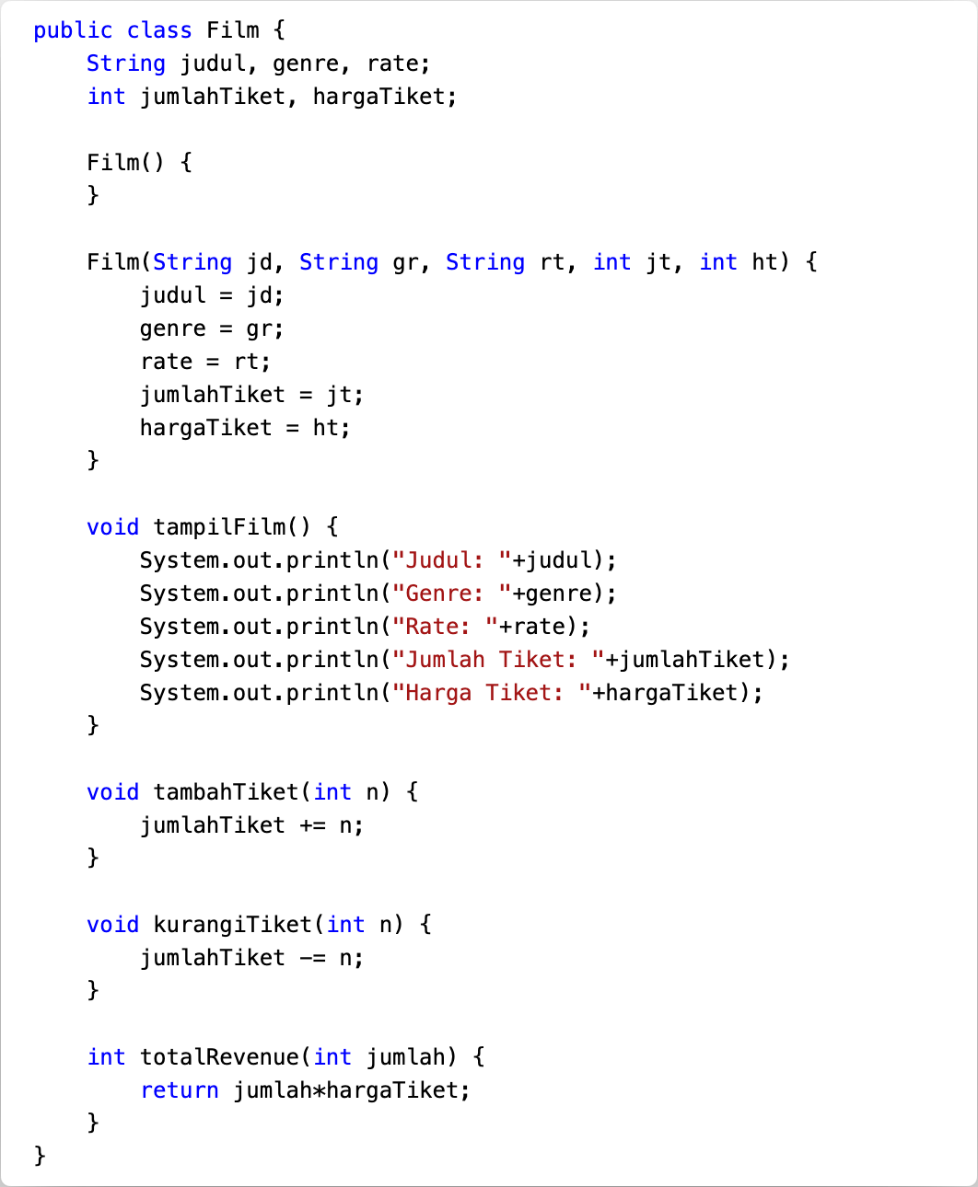
1. Pada class **FilmMain**, pada kode apa yang digunakan untuk proses instansiasi? Apa nama objek yang dihasilkan?
2. Bagaimana cara mengakses atribut dan method dari suatu objek?

# Membuat Konstruktor

Waktu : 45 Menit

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

## Langkah-langkah Percobaan

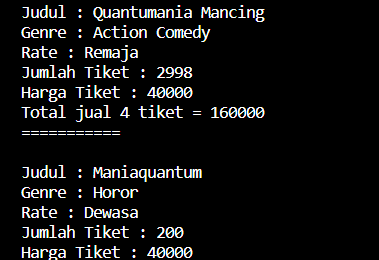
1. Perhatikan kembali class **Film**. Tambahkan di dalam class **Film** tersebut 2 buah konstruktor. 1 konstruktor default dan 1 konstruktor berparameter.

|  |
| --- |
|  |
|  |
|  |

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

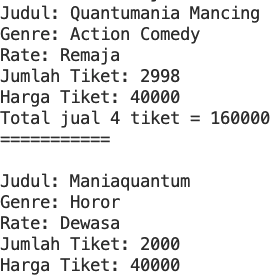


1. Jalankan kembali class **FilmMain** dan amati hasilnya.



## Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.



## 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?



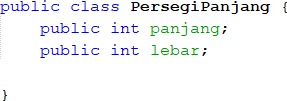
* + - 1. Coba buat objek dengan nama **film3** dengan menggunakan konstruktor berparameter dari class **Barang**.

# Membuat Array dari Object, Mengisi dan Menampilkan

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

## Langkah-langkah Percobaan

* + - 1. Buat Project baru, dengan nama “**ArrayObjects**”.
      2. Buat class **PersegiPanjang**:

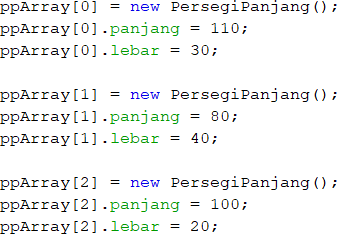


* + - 1. Pada fungsi main yaitu pada class **ArrayObjects**, buatlah array **PersegiPanjang** yang

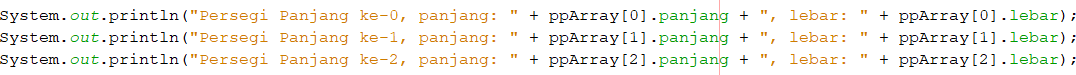
berisi 3 elemen:



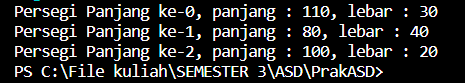
* + - 1. Kemudian isikan masing-masing atributnya:



* + - 1. Cetak ke layar semua atribut dari objek **ppArray**:

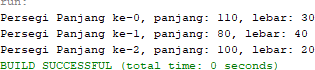


* + - 1. Jalankan dan amati hasilnya.



## Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.



## 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 konstruktur pada baris program berikut :



1. Apa yang dimaksud dengan kode berikut ini:



1. Apa yang dimaksud dengan kode berikut ini:



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

# 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 mengisikan atribut dari semua persegi panjang yang ada di ppArray.

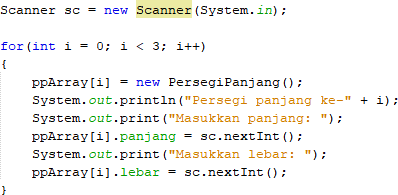
## Langkah-langkah Percobaan

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



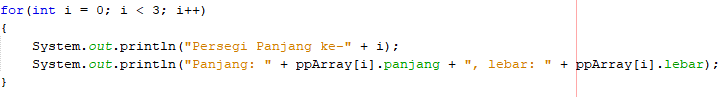
Note: Letakkan kode import dibawah kode package.

1. 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:



1. Pada praktikum 3.2 poin nomor 5, ganti kodenya dengan berikut ini, yaitu melakukan looping

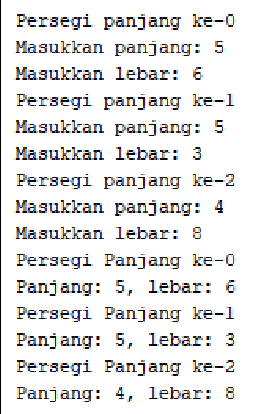
untuk mengakses isi array **ppArray** dan menampilkannya ke layar:



1. Jalankan dan amati hasilnya.

## Verifikasi Hasil Percobaan

Contoh verifikasi hasil percobaan ini.



## 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?



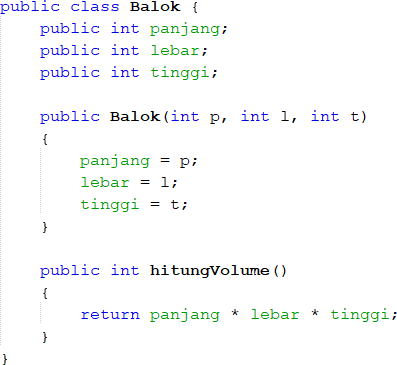
1. Modifikasi kode program pada praktikum 3.3 agar length array menjadi inputan dengan Scanner!
2. Apakah boleh Jika terjadi duplikasi instansiasi array of objek, misalkan saja instansiasi dilakukan pada **ppArray[i]** sekaligus **ppArray[0]**?Jelaskan !

# Operasi Matematika Atribut Object Array

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

## Langkah-langkah Percobaan

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



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



1. Kemudian tambahkan kode berikut ini untuk mengisi array **blArray** menggunakan

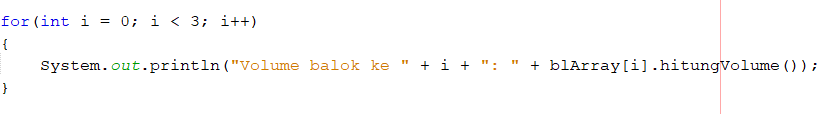
konstruktor dari class **Balok**:



1. Tampilkan semua volume balok tersebut dengan cara memanggil method **hitungVolume()**

di dalam looping seperti berikut ini:

1. Jalankan dan amati hasilnya.



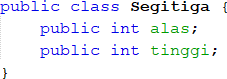
## Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.



## Pertanyaan

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



Tambahkan konstruktor pada class **Segitiga** tersebut yang berisi parameter **int a**, **int t**

yang masing-masing digunakan untuk mengisikan atribut alas dan tinggi.

* + - 1. 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*)
      2. 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()**.

# 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 |

o 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 sehinggan 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 peringatan ajuan 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.

1. Berdasarkan soal nomor 1, terapkan kasus lebih dari 1 nasabah menggunakan object of array!
2. Berdasarkan soal nomor 1 dan 2, terapkan kasus 1 nasabah dapat memiliki lebih dari 1 rekening menggunakan array biasa!