



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:

Barang
namaBarang: String jenisBarang: String stok: int hargaSatuan: int
tampilBarang(): void tambahStok(n: int): void kurangiStok(n: int): void hitungHargaTotal(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, buatlah class baru dengan nama **Barang**.
2. Lengkapi class **Barang** dengan atribut dan method yang telah digambarkan di dalam diagram class di atas, sebagai berikut:

```

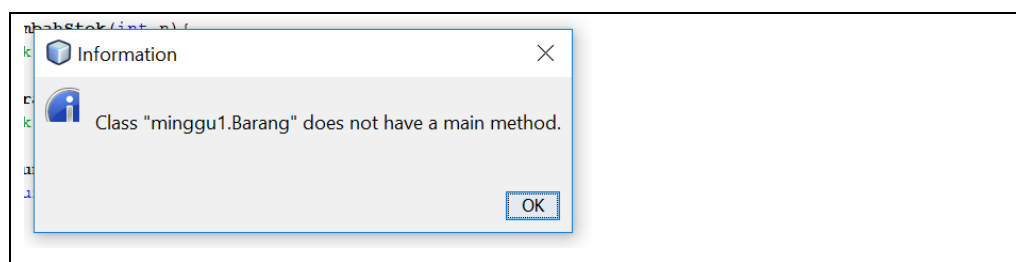
12 public class Barang {
13     String namaBarang, jenisBarang;
14     int stok, hargaSatuan;
15
16     void tampilBarang(){
17         System.out.println("Nama = "+namaBarang);
18         System.out.println("Jenis = "+jenisBarang);
19         System.out.println("Stok = "+stok);
20         System.out.println("Harga Satuan = "+hargaSatuan);
21     }
22
23     void tambahStok(int n){
24         stok=stok+n;
25     }
26
27     void kurangiStok(int n){
28         stok=stok-n;
29     }
30
31     int hitungHargaTotal(int jumlah){
32         return jumlah*hargaSatuan;
33     }
34 }
35

```

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 **Barang** 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! Dan pada baris berapa saja deklarasi method dilakukan?

5. Perhatikan method `kurangiStok()` yang ada di class `Barang`, modifikasi isi method tersebut sehingga proses pengurangan hanya dilakukan jika stok masih ada (masih lebih besar dari 0)
6. Menurut Anda, mengapa method `tambahStok()` dibuat dengan memiliki 1 parameter berupa bilangan int?
7. Menurut Anda, mengapa method `hitungHargaTotal()` memiliki tipe data int?
8. Menurut Anda, mengapa method `tambahStok()` memiliki tipe data void?

2.3 Instansiasi Objek dan Mengakses Atribut & Method

Waktu : 45 Menit

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

2.3.1 Langkah-langkah Percobaan

1. Di dalam paket `minggu2`, buatlah class baru dengan nama `BarangMain`. Dan di dalam class `BarangMain` tersebut, buatlah method `main()`.
2. Di dalam method `main()`, lakukan instansiasi, dan kemudian lanjutkan dengan mengakses atribut dan method dari objek yang telah terbentuk.

```

13 public class BarangMain {
14     public static void main(String[] args) {
15         Barang bl = new Barang();
16         bl.namaBarang = "Corsair 2 GB";
17         bl.jenisBarang = "DDR";
18         bl.hargaSatuan = 250000;
19         bl.stok = 10;
20         bl.tambahStok(1);
21         bl.kurangiStok(3);
22         bl.tampilBarang();
23         int hargaTotal = bl.hitungHargaTotal(4);
24         System.out.println("Harga 4 buah = "+hargaTotal);
25     }
26 }
27

```

3. Jalankan (Run) class `BarangMain` tersebut dan amati hasilnya.

2.3.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
run:
Nama = Corsair 2 GB
Jenis = DDR
Stok = 8
Harga Satuan = 250000
Harga 4 buah = 1000000
BUILD SUCCESSFUL (total time: 0 seconds)
```

2.3.3 Pertanyaan

1. Pada class **BarangMain**, pada baris berapakah proses instansiasi dilakukan? Dan 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 **Barang**. Tambahkan di dalam class **Barang** tersebut 2 buah konstruktork. 1 konstruktork default dan 1 konstruktork berparameter.

```
12 public class Barang {
13     String namaBarang, jenisBarang;
14     int stok, hargaSatuan;
15
16     Barang() {
17     }
18     Barang(String nm, String jn, int st, int hs){
19         namaBarang = nm;
20         jenisBarang = jn;
21         stok = st;
22         hargaSatuan = hs;
23     }
```

```

25  void tampilBarang() {
26      System.out.println("Nama = "+namaBarang);
27      System.out.println("Jenis = "+jenisBarang);
28      System.out.println("Stok = "+stok);
29      System.out.println("Harga Satuan = "+hargaSatuan);
30  }
31
32  void tambahStok(int n) {
33      stok=stok+n;
34  }
35
36  void kurangiStok(int n) {
37      stok=stok-n;
38  }
39
40  int hitungHargaTotal(int jumlah) {
41      return jumlah*hargaSatuan;
42  }
43  }

```

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

```

13  public class BarangMain {
14      public static void main(String[] args) {
15          Barang b1 = new Barang();
16          b1.namaBarang = "Corsair 2 GB";
17          b1.jenisBarang = "DDR";
18          b1.hargaSatuan = 250000;
19          b1.stok = 0;
20          //b1.tambahStok(1);
21          b1.kurangiStok(3);
22          b1.tampilBarang();
23          int hargaTotal = b1.hitungHargaTotal(4);
24          System.out.println("Harga 4 buah = "+hargaTotal);
25          Barang b2 = new Barang("Logitech", "Wireless Mouse", 25, 150000);
26          b2.tampilBarang();
27      }
28  }

```

- Jalankan kembali class **BarangMain** dan amati hasilnya.

2.4.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.



```
run:
Stok Kosong
Nama = Corsair 2 GB
Jenis = DDR
Stok = 0
Harga Satuan = 250000
Harga 4 buah = 1000000
Nama = Logitech
Jenis = Wireless Mouse
Stok = 25
Harga Satuan = 150000
BUILD SUCCESSFUL (total time: 0 seconds)
```

2.4.3 Pertanyaan

1. Perhatikan class **Barang** yang ada di Praktikum 2.4.1, pada baris berapakah deklarasi konstruktor berparameter dilakukan?
2. Perhatikan class **BarangMain** di Praktikum 2.4.1, apa sebenarnya yang dilakukan pada baris program dibawah ini?

```
Barang b2 = new Barang("Logitech", "Wireless Mouse", 150000, 25);
```

3. Coba buat objek dengan nama **b3** dengan menggunakan konstruktor berparameter dari class **Barang**.

2.5 Latihan Praktikum

Waktu : 60 Menit

1. Buat program berdasarkan diagram class berikut ini!

Barang
nama: 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

2. Buat program berdasarkan diagram class berikut ini!

PacMan
x: int y: int width: int height: int
moveLeft(): void moveRight(): void moveUp(): void moveDown(): void printPosition(): void

- Atribut `x` digunakan untuk menyimpan posisi koordinat `x` (mendatar) dari pacman, 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 pacman 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`
- Method `moveUp()` digunakan untuk mengubah posisi pacman 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`