

# JOBSHEET - 1

## CLASS DAN OBJEK

### 1. KOMPETENSI

- Mahasiswa mengenal objek dan class sebagai konsep mendasar pada pemrograman berorientasi objek
- Mahasiswa mampu mendeklarasikan class, atribut dan method
- Mahasiswa mampu membuat objek (instansiasi)
- Mahasiswa mampu mengakses atribut dan method dari suatu objek
- Mahasiswa mampu menerapkan konstruktor

### 2. ULASAN TEORI

Secara umum, paradigma dalam membuat program ada 2 macam, yaitu Pemrograman Prosedural dan Pemrograman Berorientasi Objek. Jika pada Pemrograman Prosedural, program tersusun sebagai kumpulan fungsi-fungsi atau subproses-subproses, maka dalam Pemrograman Berorientasi Objek (PBO) program tersusun atas kumpulan objek-objek yang saling berinteraksi. Objek menjadi kata kunci pada paradigma PBO. Karena pentingnya objek dalam PBO, maka memahami konteks objek menjadi sangat penting dalam menyusun program. Konsep PBO mentranslasikan kehidupan nyata ke dalam konsep pemrograman.

Dalam dunia nyata, **OBJEK** merupakan benda nyata yang bisa ditemukan di sekitar kita. Maksudnya benda nyata adalah benda tersebut sudah riil, ada dan bisa digunakan, jadi bukan masih berupa rancangan/gambaran saja. Karakteristik dari objek adalah ia memiliki properti/field/data/atribut dan juga memiliki perilaku/behaviour/method/fungsi. **ATRIBUT**/field/properti/data merupakan apa-apa yang dimiliki oleh objek, sedangkan **METHOD**/behaviour/fungsi merupakan apa-apa yang bisa dilakukan atau dikerjakan oleh objek. Setiap objek yang terbentuk, pasti berasal dari suatu rancangan/desain/blueprint/template. Artinya, tidak ada objek yang terbentuk jika sebelumnya tidak ada rancangan/desain/blueprint/template. Misalkan Gedung AF yang ada di Politeknik Negeri Malang yang ditempati saat ini, gedung ini terbentuk karena sebelumnya developer telah memiliki rancangan/blueprint Gedung AF yang dijadikan acuan dalam membuat/membangun gedung. Rancangan/template/blueprint yang akan digunakan untuk membuat objek inilah yang disebut sebagai **CLASS**. Singkat kata, objek merupakan hasil realisasi nyata atau instance dari suatu class. Proses untuk membuat objek dari suatu class disebut sebagai **INSTANSIASI**.

Karena setiap objek merupakan instance atau hasil nyata dari suatu class, maka dalam pemrograman, class merupakan bagian yang pertama kali harus dideklarasikan. Deklarasi class ditandai dengan kata kunci **class**, dengan format sebagai berikut:

```
class NamaClass {  
    //isi class  
}
```

Setelah class dideklarasikan, selanjutnya di dalam class tersebut dideklarasikan atribut dan method. Atribut menunjukkan data apa saja yang dimiliki oleh suatu class atau objek, dan method menunjukkan proses/perilaku apa yang bisa dilakukan oleh suatu class atau objek.

Format deklarasi atribut adalah sebagai berikut:

```
tipeData namaAtribut;
```

Format deklarasi method adalah sebagai berikut:

```
tipeData namaMethod (tipeData parameter) {  
    //isi method  
}
```

Keterangan:

- Tipe data method : menentukan nilai kembalian dari method. Apabila method yang dibuat tidak menghasilkan nilai kembalian, maka tipe data yang digunakan void dan tanpa return. Tetapi jika method yang dibuat menghasilkan nilai kembalian maka tipe data yang digunakan selain void, menyesuaikan dengan jenis nilai yang akan dikembalikan (di return kan).
- Parameter method : digunakan untuk melewati atau memasukan nilai dari luar method ke dalam method untuk diolah di dalam method tersebut.

Contoh:

```
class Bicycle {  
    double speed;  
    int gear;  
  
    void changeGear(int g) {  
        gear = g;  
    }  
    void speedUp(double v) {  
        speed = speed+v;  
    }  
    void applyBrakes(double v) {  
        speed = speed-v;  
    }  
    void printStates() {  
        System.out.println("Speed = "+speed);  
        System.out.println("Gear = "+gear);  
    }  
}
```

Class yang telah terbentuk, belum bisa digunakan sebelum direalisasikan dalam suatu objek nyata. Sehingga selanjutnya perlu dilakukan proses instansiasi untuk membuat objek dari class. Format instansiasi:

```
NamaClass namaObjek = new NamaClass();
```

Contoh:

```
Bicycle b = new Bicycle();
```

Setelah objek terbentuk, selanjutnya bisa diakses atribut yang dimiliki, demikian juga method yang dimiliki juga telah bisa dijalankan. Contoh untuk mengakses atribut dari suatu objek sebagai berikut:

```
b.speed = 10;  
b.gear = 1;
```

Dan contoh untuk memanggil method yang dimiliki oleh suatu objek adalah sebagai berikut:

```
b.speedUp(5);  
b.changeGear(1);  
b.printStates();
```

Untuk menggambarkan suatu rancangan class secara visual, digunakan class diagram. Class Bicycle yang telah dibuat di atas, jika digambarkan class diagram nya adalah sebagai berikut:

Bicycle
speed: double gear: int
speedUp(v: double): void applyBrakes(v: double): void changeGear(g: int): void printStates(): void

Pada saat instansiasi (proses pembentukan objek), secara lebih detil, sebenarnya yang terjadi adalah pemanggilan konstruktor. Konstruktor merupakan method istimewa yang digunakan untuk membentuk objek. Jadi sebenarnya konstruktor merupakan suatu method, akan tetapi berbeda dengan method selain konstruktor, kalau konstruktor saat deklarasi tidak memiliki tipe data, konstruktor hanya bisa dipanggil/dijalankan saat instansiasi, dan nama methodnya sama persis dengan nama class nya. Format deklarasi konstruktor:

```
NamaClass(tipeData parameter){  
  
}
```

Contoh:

```
Bicycle(){  
    //isi konstruktor  
}  
Bicycle(int g, double v){  
    //isi konstruktor  
}
```

Beberapa aturan penamaan yang perlu diperhatikan, antara lain:

- Nama class : setiap kata diawali dengan huruf kapital, dan tanpa spasi. Contoh: Sepeda, PegawaiTetap
- Nama atribut dan nama method : kata pertama diawali dengan huruf kecil, sedangkan kata selanjutnya diawali dengan huruf besar, dan tanpa spasi. Contoh: nama, jumlahBarang, tampilData()

### 3. LANGKAH PRAKTIKUM

#### PRAKTIKUM 1 – Instalasi NetBeans

NetBeans merupakan antarmuka editor program Java. Dengan menggunakan NetBeans, menulis program Java menjadi lebih mudah (fitur syntax completion dan javadoc), dan bisa meminimalisir kesalahan sintaks. Saat ini, versi terbaru yang di release adalah 8.2.

1. Pastikan di komputer sudah terinstall JDK. Jika belum, maka install JDK terlebih dahulu, dengan installer tersedia di <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Saat ini, JDK terbaru adalah versi 9, tetapi silakan download yang versi 8 saja, karena versi 9 belum di support oleh NetBeans 8.2 (versi terbaru saat ini).
2. Download installer NetBeans 8.2 di <https://netbeans.org/downloads/>
3. Jalankan installer, dan ikuti tahap-demi tahap instalasi
4. Jalankan NetBeans
5. Buat Project baru, **File > New Project**. Pada panel **Categories** pilih **Java** dan pada panel **Projects** pilih **Java Application**. Masukan nama **Project Name** (masukan nama **AlgoritmaStrukturData** sebagai nama proyek), **Project Location** dan **Project Folder**, dan **uncheck** pilihan **Create Main Class**.
6. Tekan tombol Finish. Proyek baru akan terbentuk, dan di proyek ini kita akan membuat semua program dalam satu semester ini ☺
7. Klik kanan pada **Sources Packages > New > Java Package**. Masukan **minggu1** pada nama package. Dan akan terbentuk paket baru dengan nama minggu1.
8. Klik kanan paket **minggu 1 > New > Java Class**. Masukan **HelloNetBeans** pada nama class. Pada editor class HelloNetBeans, lengkapi dengan method main sebagai berikut:

```
1 package minggu1;
2
3 public class HelloNetBeans {
4     public static void main(String[] args) {
5         System.out.println("Hello NetBeans");
6     }
7 }
```

9. Klik kanan pada class HelloNetBeans > Compile untuk mengcompile
10. Klik kanan pada class HelloNetBeans > Run File untuk menjalankan class tersebut

#### PRAKTIKUM 2 – Deklarasi Class, Atribut dan Method

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.

1. Di dalam proyek yang telah dibuat pada Praktikum 1, di dalam paket **minggu1**, 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:

```

1  package minggu1;
2
3  public class Barang {
4      String namaBarang, jenisBarang;
5      int stok, hargaSatuan;
6
7      void tampilBarang(){
8          System.out.println("Nama = "+namaBarang);
9          System.out.println("Jenis = "+jenisBarang);
10         System.out.println("Stok = "+stok);
11         System.out.println("Harga Satuan = "+hargaSatuan);
12     }
13     void tambahStok(int n){
14         stok = stok+n;
15     }
16     void kurangiStok(int n){
17         stok = stok-n;
18     }
19     int hitungHargaTotal(int jumlah){
20         return jumlah*hargaSatuan;
21     }
22 }

```

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

### PRAKTIKUM 3 – Instansiasi Objek dan Mengakses Atribut&Method

1. 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. Di dalam paket **minggu1**, buatlah class baru dengan nama **BarangMain**. Dan di dalam class **BarangMain** tersebut, buatlah method **main()**.
3. Di dalam method **main()**, lakukan instansiasi, dan kemudian lanjutkan dengan mengakses atribut dan method dari objek yang telah terbentuk.

```

1  package minggu1;
2
3  public class BarangMain {
4      public static void main(String[] args) {
5          Barang b1 = new Barang();
6          b1.namaBarang = "Corsair 2 GB";
7          b1.jenisBarang = "DDR";
8          b1.hargaSatuan = 250000;
9          b1.stok = 10;
10         b1.tambahStok(1);
11         b1.kurangiStok(3);
12         b1.tampilBarang();
13         int hargaTotal = b1.hitungHargaTotal(4);
14         System.out.println("Harga 4 buah = "+hargaTotal);
15     }
16 }

```

4. Jalankan (Run) class **BarangMain** tersebut dan amati hasilnya.

#### PRAKTIKUM 4 - Konstruktor

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

```

1  package minggu1;
2
3  public class Barang {
4      String namaBarang, jenisBarang;
5      int stok, hargaSatuan;
6
7      Barang(){
8      }
9      Barang(String nm, String jn, int st, int hs){
10         namaBarang = nm;
11         jenisBarang = jn;
12         stok = st;
13         hargaSatuan = hs;
14     }
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     void tambahStok(int n){
23         stok = stok+n;
24     }
25     void kurangiStok(int n){
26         stok = stok-n;
27     }
28     int hitungHargaTotal(int jumlah){
29         return jumlah*hargaSatuan;
30     }
31 }

```

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

```
1 package minggu1;
2
3 public class BarangMain {
4     public static void main(String[] args) {
5         Barang b1 = new Barang();
6         b1.namaBarang = "Corsair 2 GB";
7         b1.jenisBarang = "DDR";
8         b1.hargaSatuan = 250000;
9         b1.stok = 10;
10        b1.tambahStok(1);
11        b1.kurangiStok(3);
12        b1.tampilBarang();
13        int hargaTotal = b1.hitungHargaTotal(4);
14        System.out.println("Harga 4 buah = "+hargaTotal);
15
16        Barang b2 = new Barang("Logitech", "Wireless Mouse",
17                                150000, 25);
18        b2.tampilBarang();
19    }
20 }
```

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

## 11. PERTANYAAN

1. Sebutkan 2 karakteristik class/objek!
2. Kata kunci apakah yang digunakan untuk mendeklarasikan class?
3. Perhatikan **Barang** yang ada di Praktikum 2, ada berapa atribut yang dimiliki oleh class tersebut? Sebutkan!
4. Perhatikan class **Barang** yang ada di Praktikum 2, pada baris berapa saja deklarasi atribut dilakukan?
5. Perhatikan **Barang** yang ada di Praktikum 2, ada berapa method yang dimiliki oleh class tersebut? Sebutkan!
6. Perhatikan class **Barang** yang ada di Praktikum 2, pada baris berapa saja deklarasi method dilakukan?
7. 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)
8. Menurut Anda, mengapa method **tambahStok()** dibuat dengan memiliki 1 parameter berupa bilangan int?
9. Menurut Anda, mengapa method **hitungHargaTotal()** memiliki tipe data int?
10. Menurut Anda, mengapa method **tambahStok()** memiliki tipe data void?
11. Pada class **BarangMain**, pada **Praktikum 3**, pada baris berapakah proses instansiasi dilakukan? Dan apa nama objek yang dihasilkan?
12. Bagaimana cara mengakses atribut dan method dari suatu objek?
13. Perhatikan class **Barang** yang ada di Praktikum 4, pada baris berapakah deklarasi konstruktor berparameter dilakukan?
14. Perhatikan class **BarangMain** di Praktikum 4, apa sebenarnya yang dilakukan pada baris program ke 16?
15. Coba buat objek dengan nama **b3** dengan menggunakan konstruktor berparameter dari class **Barang**.

## 12.TUGAS

1. Buat program berdasarkan diagram class berikut ini!

Lingkaran
PHI: double r: double
hitungLuas(): double hitungKeliling(): double

2. Suatu toko persewaan video game salah satu yang diolah adalah peminjaman, dimana data yang dicatat ketika ada orang yang melakukan peminjaman adalah id, nama member, nama game, lama pinjam (hari) dan harga sewa per hari. Setiap peminjaman bisa menampilkan data hasil peminjaman dan harga yang harus dibayar. Buatlah class diagram dan program dari studi kasus tersebut!

3. 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

4. Buat program berdasarkan diagram class berikut ini!

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



<code>moveDown(): void</code> <code>printPosition(): void</code>
---

- 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`