

Class & Object

Pemrograman Berorientasi Object

PBO-Suprayogi.M.Kom

1

Pembahasan

- constructor
- operator new
- garbage collection and finalizers
- this
- constructor untuk inialisasi suatu nilai

PBO-Suprayogi.M.Kom

2

Membuat Class

- Class adalah template/blueprint untuk instant/object dari class.
- Code pada class berkomunikasi dengan compiler ttg bagaimana mendefinisikan, membuat, dan berinteraksi dengan object
- code berikut merupakan pembuatan class Kendaraan yang menjelaskan gambaran tentang object kendaraan

PBO-Suprayogi.M.Kom

3

Contoh Membuat Class

```

1 public class Kendaraan {
2     private String merk;
3     private int bbm;
4     //constructor
5     public Kendaraan() {
6     }
7     //mutator/setter method
8     public void setMerk(String m) {
9         merk= m;
10    }
11    //mutator/setter method
12    public void setBbm(int lbbm) {
13        bbm=lbbm;
14    }
15    //accessor/getter method
16    public String getMerk() {
17        return merk;
18    }
19    //accessor/getter method
20    public int getBbm() {
21        return bbm;
22    }
23 }

```

PBO-Suprayogi.M.Kom

4

Pembuatan Instance suatu Class

- Ketika class dibuat , instances class (objects) dapat dibuat dalam Driver Class / di dalam Class lain.
- Instance:
 - Mewariskan semua attributes dan methods yang didefinisikan dalam class.
 - Tidak tergantung dengan object lainnya.
 - Dibuat menggunakan operator new.

PBO-Suprayogi.M.Kom

5

Instansiasi suatu instance/object (penciptaan object)

```

1 public class TestKendaraan{
2     public static void main(String[] args){
3         Kendaraan mobilku = new Kendaraan();
4     }
5 }

```

PBO-Suprayogi.M.Kom

6

Constructor

- Constructors adalah methods yang membolehkan user membuat instansiasi suatu class.
- Praktek programming yang baik dimana suatu class sebaiknya memiliki constructor default.
- Constructor dengan parameter merupakan inialisasi terhadap variabel private dari class dengan nilai yang diberikan oleh user.
- Constructor tidak memiliki tipe return (void).

PBO-Suprayogi,M.Kom

7

Constructor Default

- Tidak memiliki parameter.
- Melakukan inialisasi semua variabel private dengan nilai awal.

```

1 public class Kendaraan {
2     private String merk;
3     private int bbm;
4     //constructor
5     public Kendaraan() {
6         merk="";
7         bbm=0;
8     }

```

PBO-Suprayogi,M.Kom

8

Constructor dengan Parameter

- digunakan untuk melakukan inialisasi variable-variable private dengan suatu nilai yang bukan nilai default.

```

1 public class Kendaraan {
2     private String merk;
3     private int bbm;
4     //constructor
5     public Kendaraan() {
6         merk="";
7         bbm=0;
8     }
9     public Kendaraan(String m,int bbm) {
10        merk=m;
11        bbm=bbm;
12    }

```

PBO-Suprayogi,M.Kom

9

Instansiasi Object Kendaraan

- Instansiasi object Kendaraan menggunakan constructor berparameter.

```

1 public class TestKendaraan{
2     public static void main(String[] args){
3         Kendaraan mobilku = new Kendaraan("Honda",16);
4     }
5 }

```

- menggunakan constructor default .

```

1 public class TestKendaraan{
2     public static void main(String[] args){
3         Kendaraan mobilku = new Kendaraan();
4     }
5 }

```

PBO-Suprayogi,M.Kom

10

Method

- Method adalah implementasi dari bagaimana bekerjanya sebuah class
- Berupa blok kode yang memiliki nama yang dapat dipanggil di dalam program menggunakan nama method tsb.
- Method melakukan:
 - Manipulasi data/variabel
 - Perhitungan matematika
 - Memonitor kejadian dari suatu event

PBO-Suprayogi,M.Kom

11

Pendefinisian Method

- Access Modifier (public, private, protected, default)
- Return type
- Method name
- Parameter(s)

```

public void methodName(Parameter p, ...)
{
    /*code that will execute with each call to the
    method goes here*/
}

```

```

public returnType methodName(Parameter p, ...)
{
    /*code that will execute with each call to the
    method goes here*/
}

```

PBO-Suprayogi,M.Kom

12

Access Modifier

No	Keterangan	Public	Private	Protected	Default (tidak ditulis)
1.	Akses dari dalam class	√	√	√	√
2.	Akses dari object satu folder	√	X	√	√
3.	Akses dari object beda folder	√	X	X	X
4.	Akses dari turunan satu folder	√	X	√	√
5.	Akses dari turunan beda folder	√	X	√	X

PBO-Suprayogi,M.Kom

13

Class Method

- Accessor method
 - "getter"
 - Returns the value of a specific private variable.
- Mutator method
 - "setter"
 - Changes or sets the value of a specific private variable.
- Functional method
 - Returns or performs some sort of functionality for the class.

PBO-Suprayogi,M.Kom

14

Accessor Methods

```

1 public class Kendaraan {
2     private String merk;
3     private int bbm;
4     //constructor
5     public Kendaraan () {
6     }
7     //mutator/setter method
8     public void setMerk(String m) {
9         merk= m;
10    }
11    //mutator/setter method
12    public void setBbm(int lbbm) {
13        bbm=lbbm;
14    }
15    //accessor/getter method
16    public String getMerk(){
17        return merk;
18    }
19    //accessor/getter method
20    public int getBbm(){
21        return bbm;
22    }
23 }

```

PBO-Suprayogi,M.Kom

15

Mutator Methods

```

1 public class Kendaraan {
2     private String merk;
3     private int bbm;
4     //constructor
5     public Kendaraan () {
6     }
7     //mutator/setter method
8     public void setMerk(String m) {
9         merk= m;
10    }
11    //mutator/setter method
12    public void setBbm(int lbbm) {
13        bbm=lbbm;
14    }
15    //accessor/getter method
16    public String getMerk(){
17        return merk;
18    }
19    //accessor/getter method
20    public int getBbm(){
21        return bbm;
22    }
23 }

```

PBO-Suprayogi,M.Kom

16

Functional Methods

```

1 public class Kendaraan {
2     private String merk;
3     private int bbm;
4     //constructor
5     public Kendaraan () {
6         merk="";
7         bbm=0;
8     }
9     public Kendaraan(String m,int lbbm){
10        merk=m;
11        bbm=lbbm;
12    }
13    public int compareTo(Kendaraan v1, Kendaraan v2){
14        if(v1.getBbm()== v2.getBbm())
15            return 0;
16        if(v1.getBbm()> v2.getBbm())
17            return 1;
18        return -1;
19    }
20    //accessor/getter method
21    public String getMerk(){
22        return merk;
23    }
24    //accessor/getter method
25    public int getBbm(){
26        return bbm;
27    }
28    //mutator/setter method
29    public void setMerk(String m){
30        merk= m;
31    }
32    //mutator/setter method
33    public void setBbm(int lbbm){
34        bbm=lbbm;
35    }
36 }

```

PBO-Suprayogi,M.Kom

17

menggunakan Constructor & Method dalam Driver class method main

```

1 public class TestKendaraan{
2     public static void main(String[] args){
3         Kendaraan mobilku = new Kendaraan("Honda",15);
4         Kendaraan mobilmu = new Kendaraan("Toyota",15);
5         int hasil;
6         hasil=mobilku.compareTo(mobilku,mobilmu);
7         if (hasil>0)
8             System.out.println(mobilku.getMerk()+">" +mobilmu.getMerk());
9         else if (hasil==0)
10            System.out.println(mobilku.getMerk()+"=" +mobilmu.getMerk());
11        else
12            System.out.println(mobilku.getMerk()+"<" +mobilmu.getMerk());
13    }
14 }

```

PBO-Suprayogi,M.Kom

18

this

- Within an instance method or a constructor, this is a reference to the current object.
- The reference to the object whose method or constructor is being called.
- Refer to any member of the current object by using this.
- Most commonly used when a field is shadowed by a method or constructor parameter of the same name

PBO-Suprayogi,M.Kom

19

Contoh this

Membedakan antara variabel local scope dengan class scope

```
public class Point {
    private int x;
    private int y;
    //constructor
    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
```

PBO-Suprayogi,M.Kom

20

Reference Object Representation

- When creating a new instance of an object, a reference is made to the object in memory.
- The reference points to the object.
- All attribute variables are created and initialized based on the constructor used.

```
public class Card {
    private String suit;
    private String name;
    private int points;
}
```

```
Card c = new Card();
```

```
suit = null
name = null
points = 0
```

PBO-Suprayogi,M.Kom

21

Garbage Collection

```
Kendaraan.java TestKendaraan.java
1 public class TestKendaraan {
2     public static void main(String[] args) {
3         Kendaraan mobilku = new Kendaraan("Honda", 16);
4         Kendaraan mobilmu = new Kendaraan("Toyota", 15);
5         mobilku = mobilmu;
6         System.out.println(mobilku.getMerk());
7     }
8 }
```

PBO-Suprayogi,M.Kom

22

Finalizers

- finalizer adalah code yang dipanggil oleh garbage collector ketika ditemukan tidak adanya acuan pada suatu object.
- seluruh objects mewarisi metod finalize() dari java.lang.Object.
- method tidak memiliki parameter dan dibuat tanpa mengerjakan suatu aksi ketika dipanggil.

PBO-Suprayogi,M.Kom

23

Finalizers

- Overriding pada method finalize() membuat kita dapat memodifikasi aksi sebelum garbage collection:
- Memberikan notifikasi pada user tentang adanya garbage collection
- Membersihkan resource non-Java, spt penutupan file.

PBO-Suprayogi,M.Kom

24

Contoh method Finalize

```
Protected void finalize(){
    try{
        close();//close all files
    }
    finally{
        System.out.println("Finalization has occurred");
    }
}
```

PBO-Suprayogi,M.Kom

25

Terminology

Key terms used in this lesson included:

- Method
- Accessor method
- Class
- Constructor
- Finalizers
- Garbage collection
- Initialization
- Instantiate

PBO-Suprayogi,M.Kom

26

Latihan

Buatlah aplikasi menggunakan class Product dengan variabel NmrProduct, NamaProduct, jmlStok, harga dengan method mutator dan asesor, data produk ditampung dalam bentuk array object.

PBO-Suprayogi,M.Kom

27

Latihan

```
C:\Windows\system32\cmd.exe
D:\latihan\java\JawaInventory>java Test3Product
Hallo maszabek User, silahkan tambahkan produk di
Isikan jml produk yang ingin ditambahkan
Isikan 0 apabila tdk ingin menambahkan produk>
3
Masukkan Item Produk #0
Silakan isi Nomor item
1
Isikan isi nama produk
DUD
Isikan jumlah stok
20
Isikan harga
8000
Masukkan Item Produk #1
Silakan isi Nomor item
2
Isikan isi nama produk
CD
Isikan jumlah stok
50
Isikan harga
3000
Masukkan Item Produk #2
Silakan isi Nomor item
3
Isikan isi nama produk
flasdisk
Isikan jumlah stok
5
Isikan harga
170000
-----
Daftar Inventory dari 3 products:
NoItem : 1
Nama: DUD
JmlStok: 20
Harga: 8000.0
-----
NoItem : 2
Nama: CD
JmlStok: 50
Harga: 3000.0
-----
NoItem : 3
Nama: flasdisk
JmlStok: 5
Harga: 170000.0
-----
```

PBO-Suprayogi,M.Kom

28