

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL II : Class dan Object



Disusun Oleh :

Ahmad Saiful Huda
(19102158)

Dosen Pengampu :

Merlinda Wibowo, S.T., M. Phil.

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2021

MODUL II

Class dan Object

I. TUJUAN

- a. Mahasiswa diharapkan mampu memahami konsep class dan object
- b. Mahasiswa diharapkan mampu menerjemahkan class dan object ke bahasa pemrograman

II. TOOL

- a. NetBeans IDE 8.1 atau
- b. NetBeans IDE 12.3
- c. Java SE Development Kit 8

III. DASAR TEORI

A. PENGENALAN CLASS DAN OBJECT

- a. Class

Class merupakan cetak biru (blue print) dari objek atau dengan kata lain sebuah Class menggambarkan ciri-ciri objek secara umum. Sebagai contoh Suzuki Smash, Yamaha VegaR, Honda SupraFit, dan Kawasaki KazeR merupakan objek dari Class sepeda motor. Suzuki Smash dan objek lainnya juga memiliki kesamaan atribut (merk, tipe, berat, kapasitas bensin, tipe mesin, warna, harga) dan method untuk mengakses data pada atributnya (misal fungsi untuk menginputkan data merk, tipe, berat, dsb serta fungsi untuk mencetak data merk, tipe, berat, dsb).

Simple-nya, class adalah sebuah cetakan dan object adalah produk yang sudah jadi. Dalam pemrograman, kita menggunakan class untuk merancang sebuah object. Dalam class, ada yang namanya atribut dan method. Dalam pemrograman berbasis object, ada yang namanya visibility modifier. Modifier ini menentukan apakah attribute atau method dapat diakses oleh class lain. Dalam bahasa Java, ada 3 jenis modifier:

1. Public (+), berarti attribute/method tersebut dapat diakses secara langsung oleh class lain.

2. Protected (#), berarti attribute/method tersebut dapat diakses secara langsung oleh class lain yang masih satu package dengan class dimana attribute atau method tersebut dituliskan.
3. Private (-), berarti attribute atau method tersebut tidak dapat diakses secara langsung oleh class lain.

Contoh :

```
1 public class SepedaMotor {  
2     private String merk;  
3     private long harga;  
4  
5     public void setMerk(String merkMotor) {  
6         merk = merkMotor;  
7     }  
8  
9     public String getMerk(){  
10        return merk;  
11    }  
12  
13    public long Harga(long hargaMotor) {  
14        return harga = hargaMotor;  
15    }  
16 }
```

b. Object

Objek (Object) merupakan segala sesuatu yang ada di dunia ini, yaitu manusia, hewan, tumbuhan, rumah, kendaraan, dan lain sebagainya. Contoh-contoh objek yang telah disebutkan diatas merupakan contoh objek nyata pada kehidupan kita. Pada pemrograman berorientasi objek, kita akan belajar bagaimana membawa konsep objek dalam kehidupan nyata menjadi objek dalam dunia pemrograman. Setiap objek dalam dunia nyata pasti memiliki 2 elemen penyusunnya, yaitu keadaan (state) dan perilaku/sifat (behaviour). Sebagai contoh, sepeda memiliki keadaan yaitu warna, merk, jumlah roda, ukuran roda. Dan perilaku/sifat sepeda adalah berjalan, berhenti, belok, menambah kecepatan, mengerem. Pada saat objek diterjemahkan ke dalam konsep PBO, maka elemen penyusunnya juga terdiri atas 2 bagian, yaitu : Atribut, merupakan ciri-ciri yang melekat pada suatu objek (state). Method, merupakan fungsi-fungsi yang digunakan untuk memanipulasi nilai-nilai pada atribut atau untuk melakukan hal-hal yang dapat dilakukan suatu objek (behaviour). Objek dalam konsep PBO memiliki keadaan dan perilaku yang sama seperti halnya objek di dunia nyata, karena objek dalam konsep PBO merupakan representasi objek dari dunia nyata. Objek dalam PBO merepresentasikan keadaan melalui variabel-variabel (Atribut), sedangkan perilakunya direpresentasikan dengan method (yang merupakan

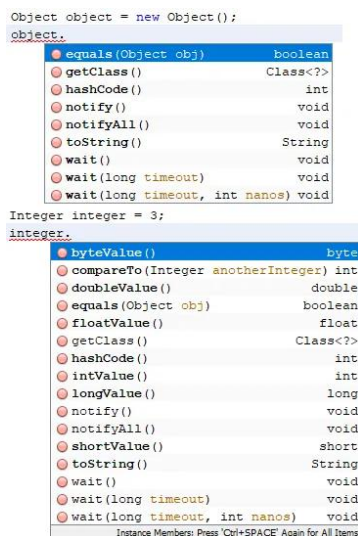
suatu fungsi yang berhubungan dengan perilaku objek tersebut maupun berhubungan dengan atribut dari objek tersebut). Objek yang memiliki kesamaan atribut dan method dapat dikelompokkan menjadi sebuah Class. Dan objek-objek yang dibuat dari suatu class itulah yang disebut dengan Instant of class.

Untuk menginstansi (membuat) objek dari class, gunakan operator new. Sintaks membuat objek dari suatu class :

```
namaClass namaObjek = new namaClass()
```

Object di java adalah bentuk jadi dari Class. Jadi class itu template atau cetakan untuk sekelompok objek yang memiliki properti yang sama. Class bisa terdiri dari:

1. Fields Template dari keadaan(property) yang bisa membedakan objek yang satu dengan yang lainnya.
2. Methods Template dari fungsi dari objek tersebut. Umumnya didalamnya juga ada mutator atau fungsi yang bisa mengubah keadaan(property) objek dan accessor untuk mengakses informasi fields pada object.
3. Constructors Sesuai dengan namanya Constructor adalah pembangun objek, umumnya memberikan nilai keadaan default dari objek.
4. Blocks kode blok seperti blok perulangan, blok percabangan dan blok lainnya.
5. Nested Class dan interface Class di dalam class, termasuk class yang didalamnya berupa fungsi abstrak (referensi type) yang disebut interface class.



IV. GUIDED

Buatlah package baru dengan format com.pbo.modul2, lalu ikuti langkah langkah di bawah:

FUNCTION Buatlah kelas baru di package guided dengan nama “DemoFunction” dan salin kode berikut:

```
6 package com.pbo.modul2;
7
8 /**
9  *
10  * @author m0n
11  */
12 public class DemoFunction {
13     public static void main(String[] args) {
14         drawTriangle(3);
15         drawSquare(5,7);
16         drawSquare(2,2);
17     }
18
19     public static void drawTriangle(int length){
20         for (int row=1; row<=length; row++){
21             for (int column=1; column <= row; column++){
22                 System.out.print("*");
23             }
24             System.out.println();
25         }
26         System.out.println();
27     }
28
29     static void drawSquare(int length, int widht){
30         for(int row=1; row<=length; row++){
31             for(int column=1; column <= widht; column++){
32                 System.out.print("*");
33             }
34             System.out.println();
35         }
36         System.out.println();
37     }
38 }
```

Hasil running:

```
Output - Run (DemoFunction) x
Changes detected - recompiling the module!
Compiling 7 source files to /home/m0n/Documents/Coding/Java/Modul2/target/classes
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Modul2 ---
*
**
***
*****
*****
*****
*****
**
**
-----
BUILD SUCCESS
```

CLASS-OBJECT

Buatlah class baru dengan nama “Mahasiswa”, dan tulis code program seperti berikut:

```
6 package com.pbo.modul2;
7
8 /**
9  *
10  * @author m0n
11  */
12 public class Mahasiswa {
13     String nama;
14     int nim;
15
16     public Mahasiswa() {}
17
18     public Mahasiswa(String nama, int nim){
19         this.nama = nama;
20         this.nim = nim;
21     }
22
23     public String getNama() {
24         return nama;
25     }
26
27     public void setNama(String nama) {
28         this.nama = nama;
29     }
30
31     public int getNim() {
32         return nim;
33     }
34
35     public void setNim(int nim) {
36         this.nim = nim;
37     }
38 }
```

Buatlah sebuah class baru dengan nama “Objek”, dan tulis kode program seperti dibawah ini.

```

6 package com.pbo.modul2;
7
8 /**
9  *
10  * @author m0n
11  */
12 public class Objek {
13     public static void main(String[] args) {
14
15         //Dekarasi objek tanpa parameter
16         Mahasiswa mahasiswaTanpa = new Mahasiswa();
17         mahasiswaTanpa.setNama("Huda");
18         mahasiswaTanpa.setNim(19102158);
19
20         //output
21         System.out.println("Mahasiswa 1");
22         System.out.println("Nama\t:" + mahasiswaTanpa.getNama());
23         System.out.println("NIM\t:" + mahasiswaTanpa.getNim());
24
25         //Deklarasi objek dengan parameter
26         Mahasiswa mahasiswa = new Mahasiswa("M0n", 19696969);
27
28         //output
29         System.out.println("Mahasiswa 2");
30         System.out.println("Nama\t:" + mahasiswa.getNama());
31         System.out.println("NIM\t:" + mahasiswa.getNim());
32
33     }
34 }

```

Hasil running:



```

Output - Run (Objek) x
Changes detected - recompiling the module!
Compiling 7 source files to /home/m0n/Documents/Coding/Java/Modul2/target/classes
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Modul2 ---
Mahasiswa 1
Nama      :Huda
NIM       :19102158
Mahasiswa 2
Nama      :M0n
NIM       :19696969
-----
BUILD SUCCESS
-----
Total time:  1.937 s
Finished at: 2021-04-17T13:48:57+07:00
-----

```

METHOD (FUNGSI DI DALAM CLASS)

Buatlah sebuah class baru dengan nama “TestPass”, dan tulis kode program seperti dibawah ini.

```

6 package com.pbo.modul2;
7
8 /**
9  *
10  * @author m0n
11  */
12 public class TestPass {
13     int nomor1, nomor2;
14
15     public TestPass(int nomor1, int nomor2) {
16         this.nomor1 = nomor1;
17         this.nomor2 = nomor2;
18     }
19
20     void calculate (int m, int n) {
21         m = n * 10;
22         n = n / 2;
23     }
24
25     void calculate (TestPass pass) {
26         pass.nomor1 = pass.nomor1 * 10;
27         pass.nomor2 = pass.nomor2 / 2;
28     }
29 }

```

Lalu buatlah sebuah class baru dengan nama “Passed”, dan tulis kode program seperti dibawah ini.

```

6 package com.pbo.modul2;
7
8 /**
9  *
10  * @author m0n
11  */
12 public class Passed {
13     public static void main(String[] args) {
14         int nomor1, nomor2;
15
16         TestPass pass = new TestPass(50, 100);
17         nomor1 = 10;
18         nomor2 = 20;
19
20         System.out.println("Nilai sebelum passed by value\t: ");
21         System.out.println("Nomor1\t: "+nomor1);
22         System.out.println("Nomor2\t: "+nomor2);
23
24         pass.calculate(nomor1, nomor2);
25         System.out.println("Nilai setelah passed by value\t: ");
26         System.out.println("Nomor1\t: "+nomor1);
27         System.out.println("Nomor2\t: "+nomor2);
28         System.out.println("");
29
30         System.out.println("Nilai sebelum passed by refrence\t: ");
31         System.out.println("Passed.Nomor1\t" + pass.nomor1);
32         System.out.println("Passed.Nomor2\t" + pass.nomor2);
33
34         pass.calculate(pass);
35         System.out.println("Nilai setelah passed by refrence\t: ");
36         System.out.println("Passed.Nomor1\t" + pass.nomor1);
37         System.out.println("Passed.Nomor2\t" + pass.nomor2);
38     }
39 }
40 }

```

Setelah melengkapinya lalu jalankan programnya, maka output dari program tersebut akan seperti ini.


```
Output - Run (Passed) X
--- maven-compiler-plugin:3.1:compile (default-compile) @ Modul2 ---
Nothing to compile - all classes are up to date

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Modul2 ---
Nilai sebelum passed by value :
Nomor1 : 10
Nomor2 : 20
Nilai setelah passed by value :
Nomor1 : 10
Nomor2 : 20

Nilai sebelum passed by reference :
Passed.Nomor1 50
Passed.Nomor2 100
Nilai setelah passed by reference :
Passed.Nomor1 500
Passed.Nomor2 50

-----
BUILD SUCCESS
```

GETTER AND SETTER

Buatlah sebuah class baru dengan nama “Manusia”, dan tulis kode program seperti dibawah ini.

```
6 package com.pbo.modul2;
7
8 /**
9  *
10  * @author m0n
11  */
12 public class Manusia {
13     private String nama;
14     private int umur;
15
16     public Manusia(){}
17
18     public Manusia(String nama) {
19         this.nama = nama;
20     }
21     public Manusia(String nama, int umur) {
22         this.nama = nama;
23         this.umur = umur;
24     }
25
26     public String getNama() {
27         return nama;
28     }
29
30     public int getUmur() {
31         return umur;
32     }
33
34     public void setNama(String nama) {
35         this.nama = nama;
36     }
37
38     public void setUmur(int umur) {
39         this.umur = umur;
40     }
41 }
42
```

Lalu buatlah sebuah class baru dengan nama “DemoManusia”, dan tulis kode program seperti dibawah ini.

```
6 package com.pbo.modul2;
7
8 /**
9  *
10  * @author m0n
11  */
12 public class DemoManusia {
13     public static void main(String[] args) {
14         Manusia arrayManusia[] = new Manusia[3];
15
16         Manusia manusia1 = new Manusia();
17         manusia1.setNama("Jake");
18         manusia1.setUmur(20);
19
20         Manusia manusia2 = new Manusia("Finn");
21         Manusia manusia3 = new Manusia("Darwin", 16);
22
23         arrayManusia[0] = manusia1;
24         arrayManusia[1] = manusia2;
25         arrayManusia[2] = manusia3;
26
27         for (Manusia x : arrayManusia){
28             System.out.println("Character ");
29             System.out.println("Nama\t: " + x.getNama());
30             System.out.println("Umur\t: " + x.getUmur());
31         }
32     }
33 }
34
```

Setelah melengkapinya lalu jalankan programnya, maka output dari program tersebut akan seperti ini:

```
Output - Run (DemoManusia) X
Changes detected - recompiling the module!
Compiling 7 source files to /home/m0n/Documents/Coding/Java/Modul2/target/classes
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Modul2 ---
Character
Nama : Jake
Umur : 20
Character
Nama : Finn
Umur : 0
Character
Nama : Darwin
Umur : 16
-----
BUILD SUCCESS
```

V. UNGUIDED

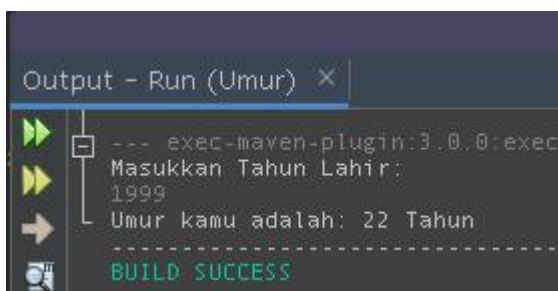
Seperti biasa, buat package baru di dalam package pertemuan2 dan beri nama unguided. Anda berada di jalan yang benar jika package anda terlihat seperti ini : `com.nama.praktikumpbo.pertemuan2.unguided`

1. Buatlah function menghitung umur berdasarkan tahun sekarang! (Buat 1 file yang berisi main dan function-nya).

Source Code:

```
6 package com.huda.praktikumpbo.pertemuan2.unguided;
7
8 import java.util.Scanner;
9
10 /**
11  *
12  * @author m0n
13  */
14 public class Umur {
15     public static void Umur() {
16         int tahun;
17         Scanner input = new Scanner(System.in);
18
19         System.out.println("Masukkan Tahun Lahir: ");
20         tahun = input.nextInt();
21
22         int umur = 2021 - tahun;
23         System.out.println("Umur kamu adalah: "+ umur+ " Tahun");
24     }
25     public static void main(String[] args) {
26         Umur();
27     }
28 }
```

Hasil running:



```
Output - Run (Umur) X
--- exec-maven-plugin:3.0.0:exec
Masukkan Tahun Lahir:
1999
Umur kamu adalah: 22 Tahun
BUILD SUCCESS
```

Pada program diatas menggunakan function *Umur* dimana terdapat variable `int` tahun untuk menginputkan tahun kemudian inisialisasi umur sama dengan 2021(tahun sekarang) dikurangi tahun (nilai input user). Class main memanggil fungsi *Umur()* untuk jalankan kemudian ditampilkan hasil outputnya.

2. Buatlah class dengan constructor tanpa parameter dan constructor dengan parameter. Jika kita membuat object dari class tersebut menggunakan constructor tanpa

parameter, beri default valuenya “no name” untuk variable nama dan “no vision” untuk variable vision! (Buat 2 file, satu untuk class dan satu untuk main-nya)

ClassObjek:

```
6      package com.huda.praktikumpbo.pertemuan2.unguided;
7
8      /**
9       *
10      * @author m0n
11      */
12      public class ClassObjek {
13          String nama = "no name";
14          String vision = "no vision";
15
16          public ClassObjek(){}
17
18          public ClassObjek(String nama){
19              this.nama = nama;
20          }
21
22          public ClassObjek(String nama, String vision) {
23              this.nama = nama;
24              this.vision = vision;
25          }
26
27          public String getNama() {
28              return nama;
29          }
30
31          public String getVision() {
32              return vision;
33          }
34
35      }
36
```

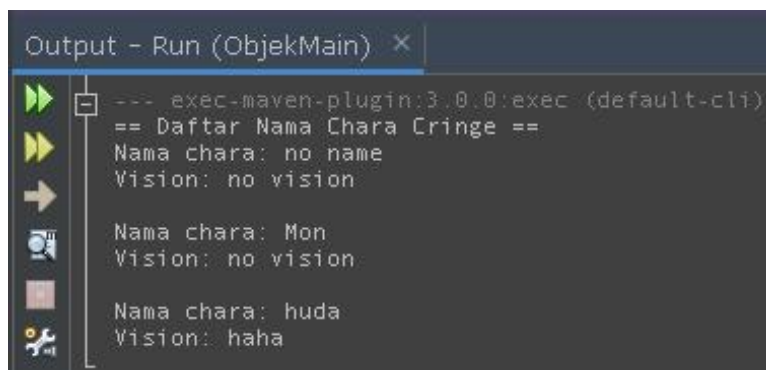
MainClass:

```

6 package com.huda.praktikumpbo.pertemuan2.unguided;
7
8 /**
9  *
10  * @author m0n
11  */
12 public class ObjekMain {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         ClassObjek arrayNama[] = new ClassObjek[3];
20
21         ClassObjek nama1 = new ClassObjek();
22         ClassObjek nama2 = new ClassObjek("Mon");
23         ClassObjek nama3 = new ClassObjek("huda", "haha");
24
25         arrayNama[0] = nama1;
26         arrayNama[1] = nama2;
27         arrayNama[2] = nama3;
28
29         System.out.println("== Daftar Nama Chara Cringe ==");
30
31         for (ClassObjek i : arrayNama){
32             System.out.println("Nama chara: " + i.getNama());
33             System.out.println("Vision: " + i.getVision());
34             System.out.println("");
35         }
36     }
37 }
38

```

Hasil running:



```

Output - Run (ObjekMain) X
--- exec-maven-plugin:3.0.0:exec (default-cli)
== Daftar Nama Chara Cringe ==
Nama chara: no name
Vision: no vision

Nama chara: Mon
Vision: no vision

Nama chara: huda
Vision: haha

```

Terdapat 2 file java dimana file class objek berisi constructor dan method yang berisi parameter. Kemudian pada main objek class tadi dipanggil untuk dieksekusi.

VI.KESIMPULAN

Class Object adalah class utama dari primitif type yang ada di java atau bisa kita katakan object class itu adalah leluhur dari class lainnya di java seperti string, Integer, Boolean, Float, dan banyak lainnya. Karena class object adalah leluhur dari class lainnya maka akan membuat class lainnya juga memiliki sifat atau fungsi yang sama yang diturunkan oleh Object class.