

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**



**Oleh :
Renna Nur Injiyani
18104019**

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM
PURWOKERTO
2020**

MODUL 1

FUNDAMENTAL OOP

Tujuan Pada Pertemuan ini :

1. Mengerti konsep dasar object oriented programming,
2. Dapat membandingkan antara pemrograman prosedural dengan pemrograman berorientasi object,
3. Dapat menerapkan object oriented programming pada program yang dibuat.

2.1 Class pada Java

2.1.1 Pengertian Class

Kelas adalah cetak biru (rancangan) dari objek. Ini berarti kita bisa membuat banyak objek dari satu macam kelas. Kelas mendefinisikan sebuah tipe dari objek. Di dalam kelas kita dapat mendeklarasikan variabel dan menciptakan object (instansiasi). Sebuah kelas mempunyai anggota (member) yang terdiri atas atribut dan method. Atribut adalah semua field identitas yang kita berikan pada suatu kelas, misal kelas manusia memiliki field atribut berupa nama, maupun umur. Method dapat kita artikan sebagai semua fungsi ataupun prosedur yang merupakan perilaku (behaviour) dari suatu kelas. Dikatakan fungsi bila method tersebut melakukan suatu proses dan mengembalikan suatu nilai (return value), dan dikatakan prosedur bila method tersebut hanya melakukan suatu proses dan tidak mengembalikan nilai (void).

2.1.2 Penggunaan Class

Pendeklarasian class sendiri ada aturannya, yaitu harus diawali dengan keyword class lalu nama classnya. Dan disarankan untuk penamaan class menggunakan huruf kapital diawal.

```
1.      class <classname> {
2.          //declaration of data member
3.          //declaration of methods
4.      }
```

Contoh pendeklarasian class pada program java.

```
1. class Mahasiswa {
2.     String nama; //data anggota
3.     int nim; //data anggota
4.     void getNamaMahasiswa() {};
```

```
5. }
```

Penjelasan kode diatas :

1. Baris satu adalah penamaan kelasnya, yaitu kelas Mahasiswa, dan diawali dengan keyword class .
2. Baris 2 inialisasi variabel, atau atribut nama dalam kelas yang bertipe data String.
3. Baris 3 inialisasi variabel, atau atribut nim dalam kelas yang bertipe data integer.
4. Baris 4, ada prosedur getNamaMahasiswa(), yang artinya jika memiliki keyword void berarti tidak memiliki nilai balikan.

2.2 Objek pada Java

2.2.1 Pengertian Objek

Object (objek) secara lugas dapat diartikan sebagai instansiasi atau hasil ciptaan dari suatu kelas yang telah dibuat sebelumnya. Dalam pengembangan program orientasi objek lebih lanjut, sebuah objek dapat dimungkinkan terdiri atas objek-objek lain. Seperti halnya objek mobil terdiri atas mesin, roda, kerangka mobil, pintu, karoseri dan lain-lain. Atau, bisa jadi sebuah objek merupakan turunan dari objek lain sehingga mewarisi sifat-sifat induknya. Misal motor dan mobil merupakan kendaraan bermotor, sehingga motor dan mobil mempunyai sifat-sifat yang dimiliki oleh kelas kendaraan bermotor dengan spesifikasi sifat-sifat tambahan sendiri. Contoh object yang lain :

Komputer, TV, mahasiswa, ponsel, ibuku, dan lainnya.

2.2.2 Penggunaan Objek

Contoh pembuatan Objek dalam java:

```
1. Manusia objMhs = new Manusia();
```

Penjelasan :

1. Membuat sebuah objek objMhs dari sebuah kelas Manusia.

2.3 Method pada Bahasa Pemrograman Objek

2.3.1 Pengertian Method

Method dikenal juga sebagai suatu function dan procedure. Dalam OOP, method digunakan untuk modularisasi program melalui pemisahan tugas dalam suatu kelas.

Pemanggilan method menspesifikasikan nama method dan menyediakan informasi (parameter) yang diperlukan untuk melaksanakan tugasnya. Deklarasi method untuk yang mengembalikan nilai (fungsi)

```
1. [modifier] Type-data namaMethod(parameter1, parameter2)
2. {
3. Deklarasi-deklarasi dan proses ;
4. return nilai-kembalian
5. }
```

Deklarasi method untuk yang tidak mengembalikan nilai (prosedur)

```
1. [modifier] void namaMethod(parameter1,parameter2)
2. {
3. Deklarasi-deklarasi dan proses ;
4. }
```

Ada dua cara melewati argumen ke method, yaitu:

1. Melewatkan secara Nilai (Pass by Value)

Digunakan untuk argumen yang mempunyai tipe data primitif (byte, short, int, long, float, double, char, dan boolean). Prosesnya adalah compiler hanya menyalin isi memori (pengalokasian suatu variabel), dan kemudian menyampaikan salinan tersebut kepada method. Isi memory ini merupakan data “sesungguhnya” yang akan dioperasikan. Karena hanya berupa salinan isi memory, maka perubahan yang terjadi pada variable akibat proses di dalam method tidak akan berpengaruh pada nilai variabel asalnya.

2. Melewatkan secara Referensi (Pass by Reference)

Digunakan pada array dan objek. Prosesnya isi memori pada variabel array dan objek merupakan penunjuk ke alamat memori yang mengandung data sesungguhnya yang akan dioperasikan. Dengan kata lain, variabel array atau objek menyimpan alamat memori bukan isi memory. Akibatnya, setiap perubahan variabel di dalam method akan mempengaruhi nilai pada variable asalnya.

2.3.2 Penggunaan Method

Contoh penggunaan method :

```
1. class Method {
2. static void SMEntertainment() {
```

```
3.      System.out.println("EXO");
4.      }
5.      public static void main(String[] args) {
6.          SMEntertainment();
7.      }
8.      }
```

Penjelasan :

1. Baris 1 membuat class method
2. Baris 2 pendeklarasian method SMEntertainment dengan tipe data void
3. Baris 3 perintah yang akan ditampilkan ketika method atau prosedur tersebut dipanggil
4. Baris 5 fungsi utama dari kelas method
5. Baris 6 memanggil method SMEntertainment() sehingga akan menampilkan output “EXO”



```
PS C:\Users\320\Documents\ATOM RENNA\ATOM P3> java Method
EXO
```

Gambar 2.1 Output Program Method

2.4 Constructor pada Bahasa Pemrograman Objek

2.4.1 Pengertian Constructor

Constructor adalah tipe khusus method yang digunakan untuk menginstansiasi atau menciptakan sebuah objek. Nama constructor adalah sama dengan nama kelasnya. Selain itu, constructor tidak bisa mengembalikan suatu nilai (not return value) bahkan void sekalipun. Defaultnya, bila kita tidak membuat constructor secara eksplisit, maka Java akan menambahkan constructor default pada program yang kita buat secara implisit. Constructor default ini tidak memiliki parameter masukan sama sekali. Namun, bila kita telah mendefinisikan minimal satu buah constructor, maka Java tidak akan menambah constructor default.

Constructor juga dimanfaatkan untuk membangun suatu objek dengan langsung mengeset atribut-atribut yang disandang pada objek yang dibuat tersebut. Oleh karena itu, constructor jenis ini haruslah memiliki parameter masukan yang akan digunakan untuk mengeset nilai atribut. Access modifier yang dipakai pada constructor selayaknya adalah

public karena constructor tersebut akan diakses di luar kelasnya (walaupun kita juga bisa memberikan access modifier pada constructor dengan private—artinya kita tidak bisa memanggil constructor tersebut di luar kelasnya). Cara memanggil constructor adalah dengan menambahkan keyword new. Keyword new dalam deklarasi ini artinya kita mengalokasikan pada memory sekian blok memori untuk menampung objek yang baru kita buat.

2.4.2 Penggunaan Constructor

Deklarasi constructor :

```
[modifier] namaclass (parameter1) {  
    Body constructor;  
}  
[modifier] namaclass (parameter1,parameter2) {  
    Body constructor;  
}  
[modifier]namaclass (parameter1,parameter2,...,paramete) {  
    Body constructor;  
}
```

Contoh penggunaan constructor pada java :

```
1. class Manusia {  
2.     private String nama;  
3.     private int umur;  
4.     public Manusia() {}  
5.     public Manusia(String a) {  
6.         nama=a;  
7.     }  
8.     public Manusia(String a, int b) {  
9.         nama=a;  
10.        umur=b;  
11.    }  
12.  
13.    public void setNama(String a) {  
14.        nama=a;  
15.    }
```

```
16. public String getNama(){
17.     return nama;
18. }
19. public void setUmur(int a){
20.     umur=a;
21. }
22. public int getUmur(){
23.     return umur;
24. }
25. }
```

Penjelasan :

1. Baris 2 deklarasi variabel nama sebagai atribut kelas yang bertipe data String.
2. Baris 3 deklarasi variabel umur sebagai atribut kelas yang bertipe data Integer.
3. Baris 4 adalah sebuah constructor yang tidak memiliki parameter.
4. Baris 5 sampai 7 adalah sebuah constructor yang memiliki satu parameter yaitu nama.
5. Baris 8 sampai 11 adalah sebuah constructor yang memiliki dua parameter yaitu nama dan umur.
6. Baris 13 sampai 15 adalah method setter untuk nama.
7. Baris 16 sampai 18 adalah method getter untuk nama.
8. Baris 19 sampai 21 adalah method setter untuk umur.
9. Baris 22 sampai 24 adalah method getter untuk umur.

Program utama constructor :

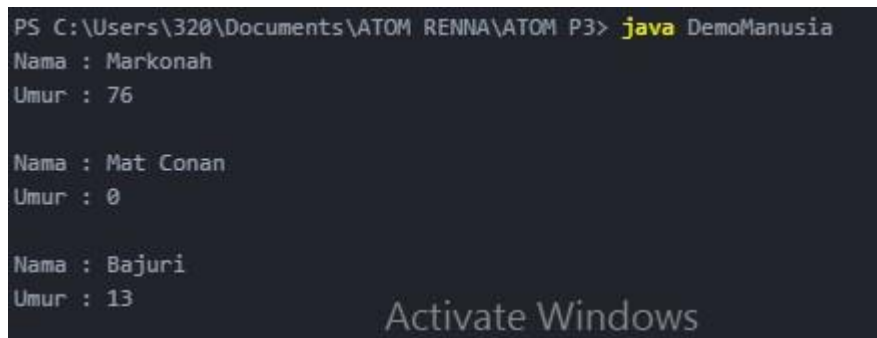
```
1. public class DemoManusia {
2.     public static void main(String[] args) {
3.         Manusia arrMns[] = new Manusia[3];
4.         Manusia objMns1 = new Manusia();
5.         objMns1.setNama("Markonah");
6.         objMns1.setUmur(76);
7.         Manusia objMns2 = new Manusia("Mat Conan");
8.         Manusia objMns3 = new Manusia("Bajuri", 13);
9.         arrMns[0] = objMns1;
```

```

10. arrMns[1] = objMns2;
11. arrMns[2] = objMns3;
12. for(int i=0; i<3; i++) {
13.     System.out.println("Nama : " + arrMns[i].getNama());
14.     System.out.println("Umur : " + arrMns[i].getUmur());
15.     System.out.println();
16. }
17. }
18. }

```

Output :



```

PS C:\Users\320\Documents\ATOM RENNA\ATOM P3> java DemoManusia
Nama : Markonah
Umur : 76

Nama : Mat Conan
Umur : 0

Nama : Bajuri
Umur : 13

```

Gambar 2.2 Output Program Construtor Penjelasan

program :

1. Baris 3 membuat array object sepanjang 3 yang akan menampung objek Manusia.
2. Baris 4 inialisasi objek objMns1 dari sebuah kelas Manusia.
3. Baris 5 adalah statement untuk mengeset nama pada objek objMns1 menggunakan method setter setName.
4. Baris 6 adalah statement untuk mengeset umur pada objek objMns1 menggunakan method setter setUmur.
5. Baris 7 untuk membuat objek objMns2 dengan constructor yang memiliki satu parameter yaitu parameter nama.
6. Baris 8 untuk membuat objek objMns3 dengan constructor yang memiliki dua parameter yaitu parameter nama dan umur.
7. Baris 9 sampai 10 mengisi array dengan masing-masing objek yang telah diinisialisasi.

8. Baris 12 sampai 18 digunakan untuk mencetak data dari objek yang telah dibuat. Menggunakan nested looping, dengan 2 perulangan.

2.5 Latihan

2.5.1 Buatlah Program menghitung konversi suhu, $C \rightarrow F$; $F \rightarrow K$; $K \rightarrow R$; $R \rightarrow C$

```
1. class Suhu {
2.     Suhu() {
3.
4.     }
5.
6.     float celciusToFahrenheit(float celcius) {
7.         return ((float)(1.8) * celcius + 32);
8.     }
9.
10.    float fahrenheitToKelvin(float fahrenheit) {
11.        return ((fahrenheit - 32) * 5) / 9 + 273;
12.    }
13.
14.    float kelvinToReamur(float kelvin) {
15.        return ((kelvin - 273) * (float)0.8);
16.    }
17.
18.    float reamurToCelcius(float reamur) {
19.        return (float)1.25 * reamur;
20.    }
21. }
22.
23. class Konversi {
24.     public static void main(String[] args) {
25.         Suhu suhu = new Suhu();
26.
27.         System.out.println("Konversi Suhu");
28.
29.         float fahrenheit = suhu.celciusToFahrenheit(100);
```

```

30. float kelvin = suhu.fahrenheitToKelvin(fahrenheit);
31. float reamur = suhu.kelvinToReamur(kelvin);
32. float celcius = suhu.reamurToCelcius(reamur);

33.
34. System.out.println("Celcius (100) to Fahrenheit = " +
    fahrenheit);
35. System.out.println("Fahrenheit (" + fahrenheit + ") to
    Kelvin = " + kelvin);
36. System.out.println("Kelvin (" + kelvin + ") to Reamur = "
    + reamur);
37. System.out.println("Reamur (" + reamur + ") to Celcius = "
    + celcius);
38. }
39. }

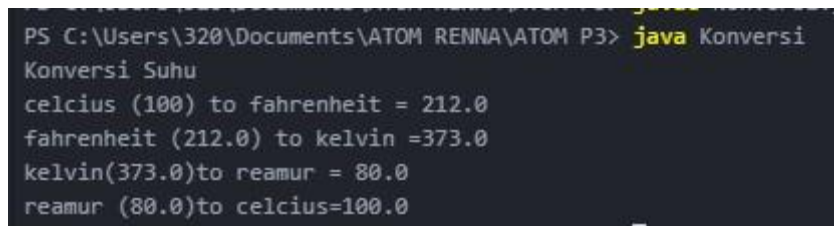
```

Penjelasan program :

1. Baris 1 membuat kelas suhu
2. Baris 2 ada konstruktor yang tidak memiliki parameter apapun.
3. Baris 6 sampai 8 adalah fungsi celciusToFahrenheit dengan satu parameter yang bertipe data float. Kembalian dari fungsi tersebut adalah berupa nilai yang bertipe data float juga. Isi dari fungsi tersebut adalah rumus untuk mengkonversi suhu dari celcius ke fahrenheit.
4. Baris 10 sampai 12 adalah fungsi fahrenheitToKelvin dengan satu parameter yang bertipe data float. Kembalian dari fungsi tersebut adalah berupa nilai yang bertipe data float juga. Isi dari fungsi tersebut adalah rumus untuk mengkonversi suhu dari fahrenheit ke kelvin.
5. Baris 14 sampai 16 adalah fungsi kelvinToReamur dengan satu parameter yang bertipe data float. Kembalian dari fungsi tersebut adalah berupa nilai yang bertipe data float juga. Isi dari fungsi tersebut adalah rumus untuk mengkonversi suhu dari kelvin ke reamur.
6. Baris 18 sampai 20 adalah fungsi reamurToCelcius dengan satu parameter yang bertipe data float. Kembalian dari fungsi tersebut adalah berupa nilai yang bertipe data float juga. Isi dari fungsi tersebut adalah rumus untuk mengkonversi suhu dari reamur ke celcius.

7. Baris 23 sampai 39 adalah program utama.
8. Baris 25, deklarasi objek suhu dari kelas Suhu dengan tidak memiliki parameter.
9. Baris 27, adalah mencetak tulisan “Konversi Suhu”.
10. Baris 29, deklarasi variabel fahrenheit yang bertipe data float yang akan langsung diisi oleh method celciusToFahrenheit.
11. Baris 30, deklarasi variabel kelvin yang bertipe data float yang akan langsung diisi oleh method fahrenheitToKelvin.
12. Baris 31, deklarasi variabel reamur yang bertipe data float yang akan langsung diisi oleh method kelvinToReamur.
13. Baris 32, deklarasi variabel celcius yang bertipe data float yang akan langsung diisi oleh method reamurToCelcius.
14. Baris 34 sampai 37 adalah kode untuk mencetak hasil dari konversi tadi.

Output :



```
PS C:\Users\320\Documents\ATOM RENNA\ATOM P3> java Konversi
Konversi Suhu
celcius (100) to fahrenheit = 212.0
fahrenheit (212.0) to kelvin =373.0
kelvin(373.0)to reamur = 80.0
reamur (80.0)to celcius=100.0
```

Gambar 2.3 Output latihan 1

2.5.2 Buatlah Program Pendaftaran Mahasiswa Baru

1.	class Mahasiswa {
2.	private String nama;
3.	private int umur;
4.	public Mahasiswa(String a, int b){
5.	nama = a;
6.	umur = b;
7.	}
8.	public String getNama(){
9.	return nama;
10.	}
11.	public void setUmur(int a){
12.	umur = a;

```

13.         }
14.         public int getUmur(){
15.             return umur;
16.         }
17.     }

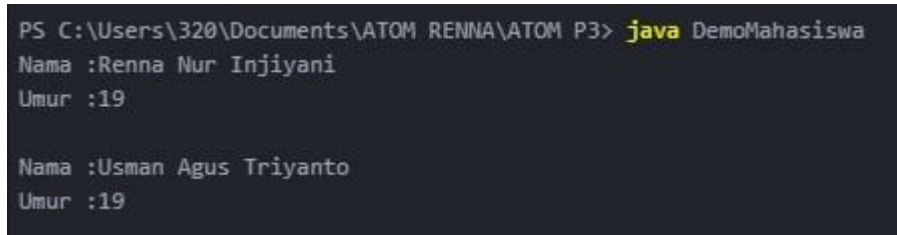
18.     class DemoMahasiswa {
19.         public static void main(String[] args){
20.             Mahasiswa arrMhs[] = new Mahasiswa[2];
21.             Mahasiswa Mhs1 = new Mahasiswa ("Renna Nur
                Injiyani", 19);
22.             Mahasiswa Mhs2 = new Mahasiswa ("Usman Agus
                Triyanto", 19);
23.             arrMhs[0] = Mhs1;
24.             arrMhs[1] = Mhs2;
25.             for(int i=0; i<3; i++){
26.                 System.out.println("Nama :"+
                    arrMhs[i].getNama());
27.                 System.out.println("Umur :"+
                    arrMhs[i].getUmur());
28.                 System.out.println();
29.             }
30.         }
31.     }

```

Penjelasan program :

1. Baris 1 sampai 17 adalah class Mahasiswa
2. Baris 2 pendeklarasian pemberian nilai pada variabel nama dengan tipe data string
3. Baris 3 pendeklarasian pendeklarasian nilai pada variabel umur dengan tipe data integer
4. Baris 18 membuat class DemoMahasiswa
5. Baris 19 fungsi main atau fungsi utama dari kelas DemoMahasiswa
6. Baris 20 sampai baris 22 menampilkan teks inputan

7. Baris 23 sampai baris 27 menampilkan hasil inputan
8. Baris 28 memanggil method Output :



```
PS C:\Users\320\Documents\ATOM RENNA\ATOM P3> java DemoMahasiswa
Nama :Renna Nur Injiyani
Umur :19

Nama :Usman Agus Triyanto
Umur :19
```

Gambar 2.4 Output latihan 2

2.5.3 Buatlah Program Pendaftaran Karyawan Baru

```
1.      Class Karyawan{
2.          Int nik = 53356;
3.          String nama = "Renna Nur Injiyani";
4.          String jabatan = "CEO";
5.          String alamat = "Purbalingga";
6.          Void output(){
7.              System.out.println("nik : " + nik );
8.              System.out.println("nama: " + nama);
9.              System.out.println("jabatan : " + jabatan );
10.             System.out.println("alamat : " + alamat );
11.         }
12.     }
13.     Public class Karyawan{
14.         Public static void main (string[] args ){
15.             Karyawan printKaryawan = new Karyawan ();
16.             PrintKaryawan.output ();
17.         }
18.     }
```

Penjelasan program :

1. Baris 1 membuat class karyawan
2. Baris 2 pendeklarasian dan pemberian nilai pada variabel nik dengan tipe data integer

3. Baris 3 sampai 5 pendeklarasian dan pemberian nilai pada variabel nama, jabatan, alamat dengan tipe data string
4. Baris 6 sampai baris 11 membuat method atau prosedur output untuk menampilkan data.
5. Baris 13 membuat class karyawan
6. Baris 14 fungsi main atau fungsi utama dari class karyawan
7. Baris 15 membua objek print karyawan dari class karyawan
8. Baris 16 memanggil method output() yang ada pada objek printKaryawan

Output :

```
PS C:\Users\320\Documents\ATOM RENNA\ATOM P3> java NomorTiga
nik : 53356
nama : Renna Nur Injiyani
jabatan : CEO
alamat : Purbalingga
```

Gambar 2.5 Output latihan 3