

Modul 3

Class dan Object #2

A) Pokok Bahasan

1. Keyword `this`
2. Konsep Overloading
3. Constructor
4. Static dan Final Modifier

B) Tujuan

1. Mahasiswa Mampu Memahami Konsep Dasar Pemrograman Berbasis Objek

C) Dasar Teori

1. Overloading

a) Pengertian

Overloading adalah sebuah konsep menambah method dengan nama method yang sudah ada sebelumnya namun dengan parameter yang berbeda

Contoh :

```
public static double rumusLuas(double sisi) {  
    return sisi * sisi;  
}  
public static double rumusLuas(double panjang, double lebar) {  
    return panjang * lebar;  
}
```

2. Keyword `this`

a) Pengertian

Keyword `this` digunakan untuk mengakses attribute atau method dari objek yang memanggilnya

Syntax :

```
/* merujuk ke attribute milik objek tersebut */  
this.[attribute]  
/* merujuk ke method milik objek tersebut */  
this.[method]()  
/* merujuk ke constructor milik class tersebut */  
this()
```

Contoh :

```
this.nama = "Saya";  
System.out.println( this.nama );  
this.panggil();
```

3. Constructor

a) Pengertian

Constructor adalah method yang dipanggil saat membuat suatu objek

Constructor memiliki nama yang sama dengan nama class

Constructor tidak boleh memiliki nilai kembali

Apabila dalam suatu kelas tidak dideklarasikan sebuah Constructor, maka compiler secara otomatis akan membuatkan constructor default

Contoh :

```
public class Ayam {  
    public int jumlahKaki;  
  
    public Ayam(int jumlahKaki) {  
        this.jumlahKaki = jumlahKaki;  
    }  
}  
Ayam a1 = new Ayam(2);  
System.out.println( a1.jumlahKaki );
```

b) Overloading Constructor

Overloading juga dapat diterapkan pada Constructor, karena sejatinya Constructor juga adalah method

Contoh :

```
public class PersegiEmpat {  
    public double panjang;  
    public double lebar;  
  
    public PersegiEmpat(double sisi) {  
        this.panjang = sisi;  
        this.lebar = sisi;  
    }  
    public PersegiEmpat(double panjang, double lebar) {  
        this.panjang = panjang;  
        this.lebar = lebar;  
    }  
}
```

4. Non Access Modifier

a) final

Keyword `final` dipakai untuk mendeklarasikan sebuah konstanta

Attribute atau method yang dideklarasikan menggunakan keyword `final` harus memiliki nilai dan tidak nilai tersebut tidak boleh diganti hingga program berakhir

Contoh :

```
public final double PHI = 3.14;
private final boolean sayangDia = true;
```

b) static

Keyword `static` menandakan bahwa attribute atau method tersebut adalah milik class tersebut dan bukan hanya milik suatu objek

Method yang menggunakan keyword `static` tidak boleh mengakses attribute atau method lain yang tidak memakai keyword `static`

Contoh :

```
public class Pengunjung {
    public static jumlahPengunjung = 0;
    public static panggilPelayan() {
        System.out.println("Hai Pelayan!");
    }
}
```

```
Pengunjung.jumlahPengunjung = 10;
Pengunjung.panggilPelayan();
```

D) Praktik

file : Handphone.java

```
public class Handphone {
    public String merk;
    public int harga;
    // Attribut static totalProduksi
    public static int totalProduksi = 0;
    // Konstanta static CPU
    public static final String CPU = "Intel Tripple Core 1st Gen";

    // Constructor
    public Handphone(String merk, int harga) {
        this.merk = merk;
        this.harga = harga;
        Main.tulis("\tHandphone baru telah dibeli\t");
        Handphone.totalProduksi += 1;
    }
    // Overloading Constructor
```

```

public Handphone(String merk) {
    this.merk = merk;
    if (merk.equals("Nokia")) {
        this.harga = 1_000;
    } else if (merk.equals("Samsung")) {
        this.harga = 2_000;
    } else if (merk.equals("Apple")) {
        this.harga = 100_000_000;
    } else {
        this.harga = 0;
    }
    Main.tulis("\tHandphone baru telah dibeli");
    Handphone.totalProduksi += 1;
}
public Handphone() {
    // Memanggil Constructor dengan parameter String dan Integer
    this("Nokia", 1_000);
}
}

```

file : PembelianHandphone.java

```

public class PembelianHandphone {

    public Handphone jenisHP;
    public int jumlah;
    public boolean statusPembelian = false;

    // Objek Sebagai parameter
    public void beliHape(Handphone jenisHP, int jumlah, int uangAnda) {
        int totalHarga = jenisHP.harga * jumlah;

        if (totalHarga > uangAnda) {
            Main.tulis("Pembelian Gagal");
            return;
        }
        this.jenisHP = jenisHP;
        this.jumlah = jumlah;
        this.statusPembelian = true;
        int kembalian = this.hitungKembalian(totalHarga, uangAnda);

        Main.tulis("Pembelian Handphone "+jenisHP.merk+" sebanyak "+jumlah+"
Berhasil Dilakukan");
        Main.tulis("Kembalian anda "+kembalian);
    }

    // method dengan nilai kembalian berupa integer
    public int hitungKembalian(int totalHarga, int uang) {
        return uang - totalHarga;
    }

    public int totalHarga() {
        return jenisHP.harga * jumlah;
    }
}

```

```
}  
}
```

file : Main.java

```
public class Main {  
    // Overloading method bernama tulis  
    public static void tulis() {  
        System.out.println("Saya Sedang Menulis");  
    }  
    public static void tulis(String text) {  
        System.out.println(text);  
    }  
    public static void tulis(int angka) {  
        System.out.println(angka);  
    }  
    public static void tulis(boolean kondisi) {  
        System.out.println(kondisi);  
    }  
    public static void tulis(String text, int berapaKali) {  
        for(int i = 0; i < berapaKali; i++) {  
            tulis(text);  
        }  
    }  
  
    public static void main(String[] args) {  
        tulis();  
        tulis("Hello, "+" world!");  
        tulis(10 + 20);  
        tulis(true || false);  
        tulis("ayam", 3);  
  
        final int jumlahKekasih = 0;  
        // jumlahKekasih++; // Error  
        tulis(jumlahKekasih);  
  
        Handphone hpSaya;  
  
        hpSaya = new Handphone("Oppo", 50_000);  
        tulis( "Merk Handphone = "+hpSaya.merk );  
        tulis( "Merk Handphone = "+hpSaya.harga );  
  
        hpSaya = new Handphone();  
        tulis( "Merk Handphone = "+hpSaya.merk );  
        tulis( "Merk Handphone = "+hpSaya.harga );  
  
        hpSaya = new Handphone("Apple");  
        tulis( "Merk Handphone = "+hpSaya.merk );  
        tulis( "Merk Handphone = "+hpSaya.harga );  
  
        tulis( "Jumlah Handphone yang telah di produksi =
```

```
" + Handphone.totalProduksi );  
    tulis( "CPU yang dipakai pada Handphone = " + Handphone.CPU );  
  
    PembelianHandphone pembelian = new PembelianHandphone();  
    pembelian.beliHape( new Handphone("Samsung"), 3, 50_000 );  
    tulis("Total Harga = " + pembelian.totalHarga());  
    }  
}
```