

**LAPORAN RESMI**  
**MODUL II**  
**CONSTRUCTOR DAN KEYWORD STATIC**  
**PEMROGRAMAN BERBASIS OBJEK**



<b>NAMA</b>	<b>: ANISYAFAAH</b>
<b>N.R.P</b>	<b>: 220441100105</b>
<b>DOSEN</b>	<b>: FIRMANSYAH ADIPUTRA, ST., M.Cs.</b>
<b>ASISTEN</b>	<b>: KUKUH COKRO WIBOWO</b>
<b>TGL PRAKTIKUM</b>	<b>: 31 MARET 2023</b>

**Disetujui : 06 April 2023**  
**Asisten**

**KUKUH COKRO WIBOWO**  
**21.04.411.00102**



**LABORATORIUM BISNIS INTELIJEN SISTEM**  
**PRODI SISTEM INFORMASI**  
**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS TRUNOJOYO MADURA**

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Semakin berkembangnya teknologi di masa sekarang ini, maka akan mendapatkan sarana yang dapat mempermudah manusia dalam menyampaikan informasi kepada satu sama lain dengan tepat dan cepat. Teknologi dan kemajuannya yang pesat sangatlah membantu manusia, bahkan dengan kemajuan teknologi yang adapun bisa menciptakan peluang untuk menghasilkan pundi-pundi rupiah tanpa harus meninggalkan rumah sekalipun. Hal ini tentu akan sangat bermanfaat bagi masa yang akan datang.

Berbicara tentang teknologi, akan sangat erat kaitannya dengan software atau perangkat lunak. Software berfungsi sebagai pelengkap komputer agar dapat dioperasikan dengan baik oleh perintah-perintah tertentu, contohnya yaitu sebuah pemrograman. Pemrograman akan membutuhkan sebuah bahasa dimana didalamnya berisi instruksi atau perintah yang dapat dimengerti oleh komputer, salah satunya bahasa pemrograman Java. Berikut ini adalah contoh pembuatan program dengan menggunakan bahasa pemrograman Java dan menggunakan constructor serta keyword static.

### 1.2 Tujuan

- Mahasiswa mampu memahami konsep Constructor dalam Pemrograman Berorientasi Objek serta mampu mengimplementasikannya
- Mahasiswa mampu memahami penggunaan keyword static dalam Pemrograman Berorientasi Objek serta mampu mengimplementasikannya.

## BAB II DASAR TEORI

### 2.1 Constructor

#### Pengertian/Konsep

Constructor adalah method yang secara otomatis dipanggil / dijalankan pada saat new dipakai untuk menciptakan instan kelas. Atau dengan kata lain constructor adalah method yang pertama kali dijalankan pada saat sebuah objek pertama kali diciptakan. Jika dalam sebuah class tidak terdapat constructor, maka secara otomatis Java akan membuatkan sebuah default constructor. Sama halnya dengan method, constructor dapat memiliki satu atau banyak parameter maupun tanpa parameter. Hal mendasar yang perlu diperhatikan, yaitu :

- Nama Constructor sama dengan nama Class
- Tidak ada return type yang diberikan ke dalam Constructor Signature
- Tidak ada return statement, di dalam tubuh constructor.

#### Contoh Program

##### a. Constructor Kendaraan.java

```
public class ConstructorKendaraan {  
    private String merk; //ini adalah instant variable  
    private static String pemilik; //ini adalah static variable  
  
    //ini adalah constructor tanpa parameter  
    protected ConstructorKendaraan() {  
        merk = null;  
    }  
  
    // overloading constructor  
    // ini adalah constructor dengan parameter merk  
    protected ConstructorKendaraan(String merk) {
```

```

        this.merk = merk ;
        merk = null;
    }

    protected void setMerk (String merk) {
        this.merk = merk ;
    }

    protected String getMerk () {
        return merk ;
    }

    //ini adalah static method
    protected static void setPemilik (String pemilik) {
        ConstructorKendaraan.pemilik = pemilik;
    }

    //ini adalah static method
    protected static String getPemilik () {
        return ConstructorKendaraan.pemilik;
    }

    protected void tampil (String a)
    { System.out.println(a);
      a = null;
    }

    protected void hapus ()
    { //menghapus variable private dari memory
      merk = null;
      pemilik = null;
    }
}

```



b. ConstructorMotor.java

// ConstructorMotor turunan dari class ConstructorKendaraan

public class ConstructorMotor extends ConstructorKendaraan {

private String warna; // ini adalah instant variable

// ini adalah constructor dengan parameter merk & warna

protected ConstructorMotor (String merk, String warna) {

// memanggil constructor parent (class ConstructorKendaraan)

// dengan keyword super dan parameter merk

super(merk);

this.warna = warna;

// menghapus variable parameter dari memory

merk = null;

warna = null;

}

protected String getWarna() {

return warna;

}

protected void hapus()

{ // menghapus variable private dari memory

warna = null;

// memanggil method hapus class parent (class  
ConstructorKendaraan)

// dengan keyword super

super.hapus();

}

}

c. MainConstructorMotor.java

public class MainConstructorMotor {

public static void main (String

```
{ String pemilik = "Ahmad Afif";  
  String merk = "Honda";  
  String warna = "Merah";
```

```
// cara akses static variable dan static method dapat  
// dengan memanggil class (ConstructorKendaraan) secara  
// langsung
```

```
ConstructorKendaraan.setPemilik(pemilik);  
System.out.println("Pemilik Kendaraan = " +  
ConstructorKendaraan.getPemilik());  
System.out.println("=====");
```

```
// variable merk menjadi parameter Constructor pada saat  
// instansiasi/membuat objek baru (ob)
```

```
ConstructorKendaraan ob = new ConstructorKendaraan(merk);  
ob.tampil("Merk Kendaraan = " + ob.getMerk());
```

```
// cara akses static variable dan static method dapat  
// juga dengan objek (ob) pada method getPemilik()  
ob.tampil("Pemilik Kendaraan = " + ob.getPemilik());  
System.out.println("=====");
```

```
// instansiasi/membuat objek baru (ob2) tanpa parameter  
ConstructorKendaraan ob2 = new ConstructorKendaraan();
```

```
// bandingkan akses untuk menampikan nilai pada instant
```

```
// variable (merk) dan static variable (pemilik) melalui
```

```
// method getter setelah membuat objek baru "ob2",
```

```
// dimana sebelumnya juga sudah membuat objek "ob".
```

```
// instant variable (merk) nilainya akan hilang,
```

```
// sedangkan static variable (pemilik) nilainya tidak
```

```
// hilang/berubah
```

```
ob2.tampil("Merk Kendaraan (instant variable) = " +  
ob2.getMerk());
```

```
ob2.tampil("Pemilik Kendaraan (static variable) = " +  
ob2.getPemilik());
```

```

System.out.println("=====");

//variable merk dan warna menjadi parameter Constructor
//pada saat instansiasi / membuat objek baru (ob3)
ConstructorMotor ob3 = new ConstructorMotor (merk , warna);
ob3.tampil ("Merk Motor = " + ob3.getMerk());
ob3.tampil ("Warna Motor = " + ob3.getWarna());
ob3.tampil ("Pemilik Motor = " + ob3.getPemilik());

    pemilik = null;
    merk = null;
    warna = null;
    ob.hapus();
    ob = null;
    ob2 = null;
    ob3 = null;
}
}

```

## 2.2 Keyword Static

### Pengertian/Konsep

Dalam pemrograman java, keyword static digunakan untuk mengakses variable ataupun method (prosedur atau fungsi) pada class tertentu tanpa harus membuat suatu objek dari class itu.

Umumnya untuk mengakses member dari kelas lain kita harus membuat objek kelas, tapi dengan menggunakan keyword static kita dapat langsung menggunakan member kelas lain. Keyword static bisa digunakan untuk variable ataupun method.

Static member adalah variable atau method yang bukan milik dari suatu objek, melainkan milik dari class.

Bentuk umum:

```
static void namaMethodStatic() {
```

Method yang dideklarasikan sebagai static memiliki aturan

sebagai berikut :

- a. Hanya dapat dipanggil oleh method lain yang juga adalah static method
- b. Hanya dapat mengakses atribut static
- c. Tidak dapat menggunakan keyword this dan super, karena kedua keyword ini menunjuk ke suatu instance tertentu, bukan pada sebuah object.

#### Contoh Program

- a. Class Makanan
- b. Class MakananKhasMadura

#### Tampilan Class

- a. Class Makanan

Pada class Makanan, method di set sebagai static dengan parameter makanan dan harga.

```
public class Makanan {  
    static void Madura (String nama, int harga) {  
        System.out.println ("Nama Makanan : " + nama);  
        System.out.println (" Harga : Rp " + harga);  
        System.out.println ("");  
    }  
}
```

- b. Class MakananKhasMadura

Pada class MakananKhasMadura, method juga bersifat static tanpa membuat instance objek dari kelas Makanan.

```
public class MakananKhasMadura {  
    public static void main (String[] args) {  
        Makanan.Madura ("Sate", 10000);  
        Makanan.Madura ("Soto", 10000);  
        Makanan.Madura ("Campur", 7000);  
        Makanan.Madura ("Kusak", 7000);  
    }  
}
```



### Running Program

Pada saat program dieksekusi, maka akan menampilkan daftar makanan dan harga yang telah diset nilainya. Berikut tampilannya!

Nama Makanan : Sate

Harga : Rp 10000

Nama Makanan : Soto

Harga : Rp 10000

Nama Makanan : Campur

Harga : Rp 7000

Nama Makanan : Rujak

Harga : Rp 7000

### Penjelasan Program

#### a. Class Makanan

Pada class Makanan, terdapat method static dengan parameter makanan dan harga.

- i. Baris 2, keyword static yang berupa method
- ii. Baris 3-4, mencetak nama makanan dan harga

#### b. Class MakanankhasMadura

Pada class MakanankhasMadura, method juga bersifat static tanpa membuat instance objek dari kelas Makanan.

- i. Baris 3, method static untuk menjalankan program
- ii. Baris 4-7, nilai dari parameter yang terdapat pada class Makanan.

BAB III  
TUGAS PENDAHULUAN

3.1 Soal

1. Jelaskan perbedaan Constructor dan Keyword Static !
2. Jelaskan setter dan getter menurut bahasa pemahamanmu !

3.2 jawab

1. a) Constructor

Constructor adalah method yang pertama kali dijalankan pada saat sebuah objek pertama kali diciptakan. Constructor akan langsung memberikan nilai awal pada saat perintah new, membuat suatu objek.

b) Keyword Static

Keyword Static adalah perintah khusus yang memungkinkan sebuah method diakses langsung tanpa melalui objek, tapi cukup nama class saja.

2. a) Setter

Setter adalah method yang digunakan untuk memasukkan dan memperbarui suatu nilai atau value dari sebuah Variable.

b) Getter

Getter adalah method yang digunakan untuk mengambil nilai atau value yang telah diperbarui dalam sebuah Variable.

## BAB IV IMPLEMENTASI

### 4.1 Soal

Buatlah aplikasi Data Mahasiswa sebagai berikut

A. Masukkan Nama Universitas! Buat static variable dan static method setter dan getternya!

B. Masukkan data mahasiswa

1. NIM
2. ALAMAT
3. NAMA

C. JURUSAN :

1. 41 = TEKNIK INFORMATIKA
2. 42 = TEKNIK INDUSTRI
3. 43 = TEKNIK ELEKTRO
4. 44 = SISTEM INFORMASI
5. 48 = TEKNIK MESIN
6. 49 = TEKNIK MEKATRONIKA

D. Apakah Anda ingin memasukkan data lagi? (Y) Ya;  
(T) Tidak

Jika user memasukkan "Y", maka lanjut untuk memasukkan data mahasiswa lagi dan jika user memasukkan data "T", maka aplikasi berhenti.

Desainlah aplikasi Data Mahasiswa tersebut dengan konsep constructor dan inheritance (tentukan parent class dan child class). Setelah itu, implementasikan class-class yang telah didesain dengan membuat program sederhana yang memiliki fasilitas entry data Mahasiswa dan melihat daftar mahasiswa yang telah dimasukkan.

### 4.2 jawaban

1. a) Hasil

Selamat Datang!

Kampus : Universitas Trunojoyo Madura

Silahkan Masukkan Nama Anda :

Anisa

Silahkan Masukkan NIM Anda :

220441100105

Silahkan Masukkan Alamat Anda :

Bangkalan

===== JURUSAN =====

44

=====

Nama : Anisa

NIM : 220441100105

Alamat : Bangkalan

Jurusan : SISTEM INFORMASI

=====

Apakah Anda Ingin Mengulang Atau Tidak ? Y/T

T

BUILD SUCCESSFUL (total time : 22 seconds)

b) Source Code

```
package Project2;
```

```
import java.util.Scanner;
```

```
public class DataMahasiswa {
```

```
    String kampus;
```

```
    DataMahasiswa()
```

```
        System.out.println("Selamat Datang!");
```

```
}
```

```
void setkampus (String kampus) {
```

```
    kampus = kampus
```

```
}
```



```
String getkampus() {  
    return kampus;  
}
```

```
public static void main (String[] args) {  
    DataMahasiswa DataMhs = new DataMahasiswa ();  
    Scanner scan = new Scanner (System.in);  
    DataMhs.kampus = "Universitas Trunojoyo Madura";  
    System.out.println ("Kampus : " + DataMhs.getkampus());
```

```
    while (true) {  
        System.out.println ("Masukkan Nama Anda :");  
        String nama = scan.nextLine();  
        System.out.println ("Masukkan NIM Anda :");  
        String nim = scan.nextLine();  
        System.out.println ("Masukkan Alamat Anda :");  
        String alamat = scan.nextLine();  
        System.out.println ("===== Jurusan =====");  
        String opsi = scan.nextLine();  
        String jurusan = "";  
        switch (opsi) {  
            case "41":  
                jurusan = "TEKNIK INFORMATIKA";  
            case "42":  
                jurusan = "TEKNIK INDUSTRI";  
                break;  
            case "43":  
                jurusan = "TEKNIK ELEKTRO";  
                break;  
            case "44":  
                jurusan = "SISTEM INFORMASI";  
                break;  
            case "48":
```

```

        jurusan = "TEKNIK MESIN";
        break;
    case "49":
        jurusan = "TEKNIK MEKATRONIKA";
        break;
    case "-":
        jurusan = "Tidak Tersedia";
        break;
    default:
        break;
}

System.out.println("=====");
System.out.println("Nama : " + nama);
System.out.println("Nim : " + nim);
System.out.println("Alamat : " + alamat);
System.out.println("Jurusan : " + jurusan);
System.out.println("=====");
System.out.println("Apakah Ingin Mengulang? Y/T");
String ulang = scan.nextLine();
if ("Y".equals(ulang)) {
    break;
}
}
}
}

```

### c) Penjelasan

Pada program untuk memasukan beberapa data yang terdiri dari nama, nim, alamat, dan jurusan maka perlu menggunakan method set dan get dan juga perulangan (while) serta pengkondisian switch case untuk memilih opsi jurusan.

## BAB V

### PENUTUP

#### 5.1 Analisa

Pada hasil praktikum, praktikan menganalisa bahwa dalam membuat program menggunakan bahasa pemrograman Java dan menggunakan Constructor serta Keyword Static terdapat beberapa hal yang perlu diperhatikan, antara lain :

- a) Nama Constructor harus sama dengan nama Class
  - b) Tidak ada return type yang diberikan ke dalam Constructor Signature
  - c) Tidak ada return statement, di dalam tubuh Constructor
- Untuk Keyword Static, digunakan untuk mengakses method pada class tanpa membuat objek dari class tersebut.

#### 5.2 Kesimpulan

1. Constructor adalah method yang secara otomatis dipanggil/djalankan pada saat new dipakai untuk menciptakan instan kelas.
2. Keyword Static adalah perintah khusus yang memungkinkan sebuah method diakses langsung tanpa melalui objek, tapi cukup nama class saja.
3. Program yang telah diimplementasikan menggunakan Constructor dan Keyword Static