**LAPORAN RESMI**

**MODUL II**

**(CONSTRUCTOR DAN KEYWORD STATIC)**

**PEMROGRAMAN BERBASIS OBJEK**

**Disetujui : 17 MEI 2024**

**Asisten**

**MUHAMMAD ROSYID MAULANA**

**20.04.411.00019**

**NAMA : ABAS PERMADANI**

**N.R.P : 230441100107**

**DOSEN : ACHMAD ZAIN NUR, S.KOM., M.T.**

**ASISTEN : MUHAMMAD ROSYID MAULANA**

**TGL PRAKTIKUM : 30 MARET 2023**

**LABORATORIUM BISNIS INTELIJEN SISTEM**

**PRODI SISTEM INFORMASI**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Constructor adalah metode khusus dalam sebuah kelas yang secara otomatis dipanggil saat objek dari kelas tersebut dibuat. Constructor biasanya digunakan untuk melakukan inisialisasi awal objek, seperti menetapkan nilai awal untuk atribut-atribut objek. ata kunci static digunakan untuk menyatakan bahwa suatu metode, variabel, atau blok kode merupakan milik kelas (bukan objek) dan dapat diakses langsung dari kelas tersebut tanpa perlu membuat objek dari kelas tersebut.

Kedua konsep ini sangat penting dalam pemrograman berorientasi objek, karena constructor digunakan untuk menginisialisasi objek saat pembuatannya, sementara variabel atau metode static digunakan untuk menyimpan informasi bersama atau menyediakan perilaku yang terkait dengan kelas itu sendiri, bukan dengan setiap objek individu yang dibuat dari kelas tersebut.

## **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.

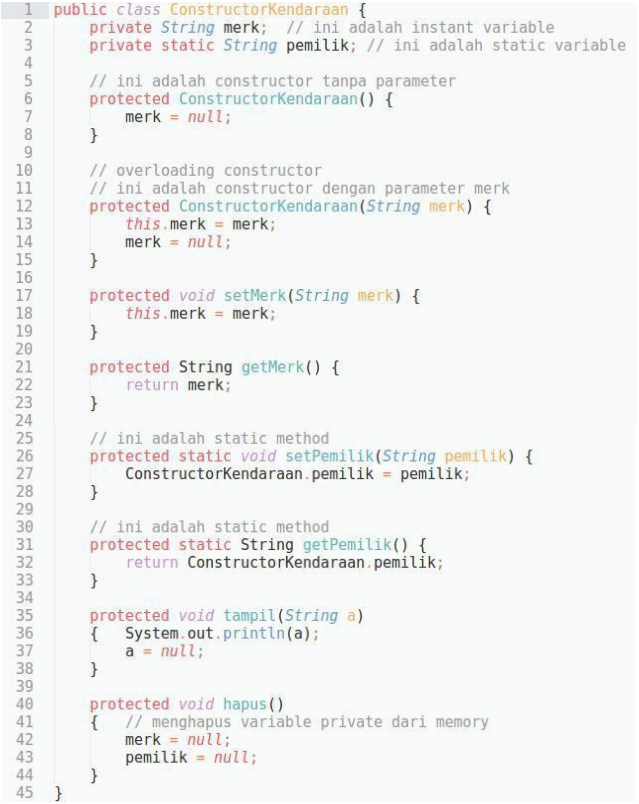
# **BAB II DASAR TEORI**

## **Constructor**

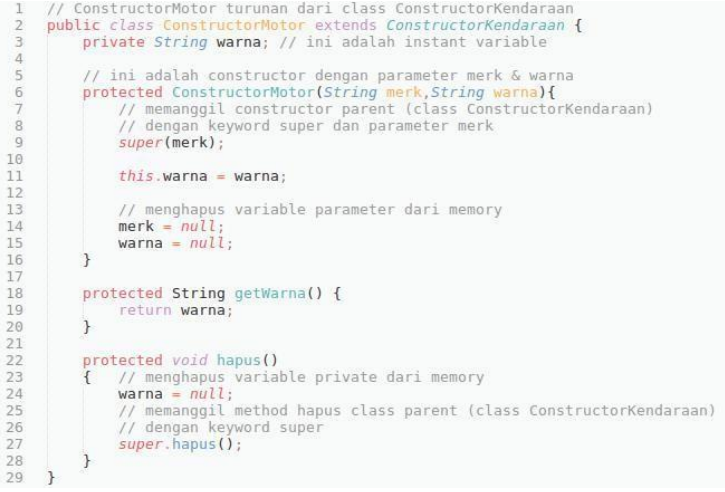
1. **Pengertian/Konsep**

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 halnnya dengan method, constructor dapat memiliki satu atau banyak parameter maupun tanpa parameter. Hal mendasar yang perlu diperhatikan, yaitu :

1. Nama Constructor sama dengan nama Class.
2. Tidak ada return type yang diberikan kedalam Constructor Signature.
3. Tidak ada return statement, di dalam tubuh constructor.
4. **Contoh Program**
5. Constructor Kendaraan.java

****

1. ConstructorMotor.java

****

1. MainConstructorMotor.java

****

****



## **Keyword Static**

1. **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.

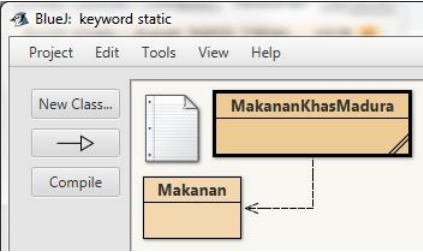
Static member adalah variable atau method yang bukan milik dari suatu object, melainkan milik dari class.

Bentuk Umum:

**static void namaMethodStatic(){**

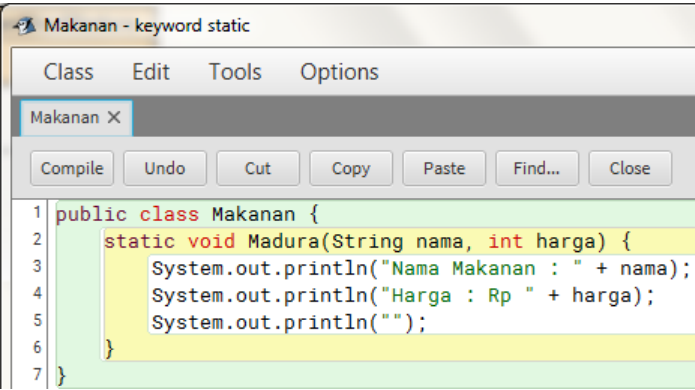
Method yang dideklarasikan sebagai static memiliki aturan sebagai berikut :

1. Hanya dapat dipanggil oleh method lain yang juga adalah static method.
2. Hanya dapat mengakses atribut static.
3. Tidak dapat menggunakan keyword this dan super, karena kedua keyword inimenunjuk ke suatu instance tertentu, bukan pada sebuah object.
4. **Contoh Program**

****

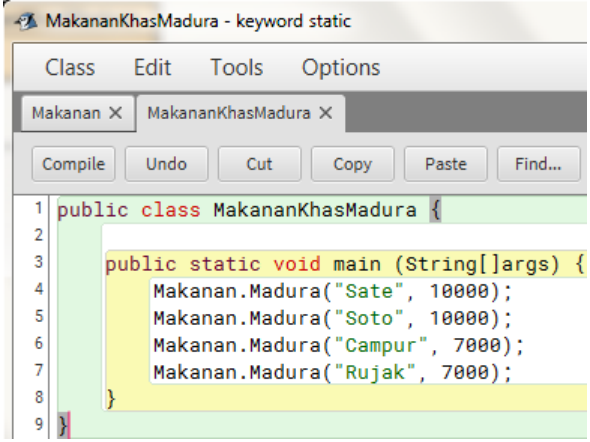
1. Class Makanan
2. Class MakananKhasMadura
   1. **Tampilan Class**
3. **Class Makanan**

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

****

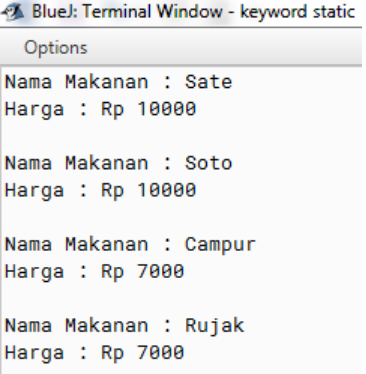
1. **Class MakananKhasMadura**

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

****

**Running Program**

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

****

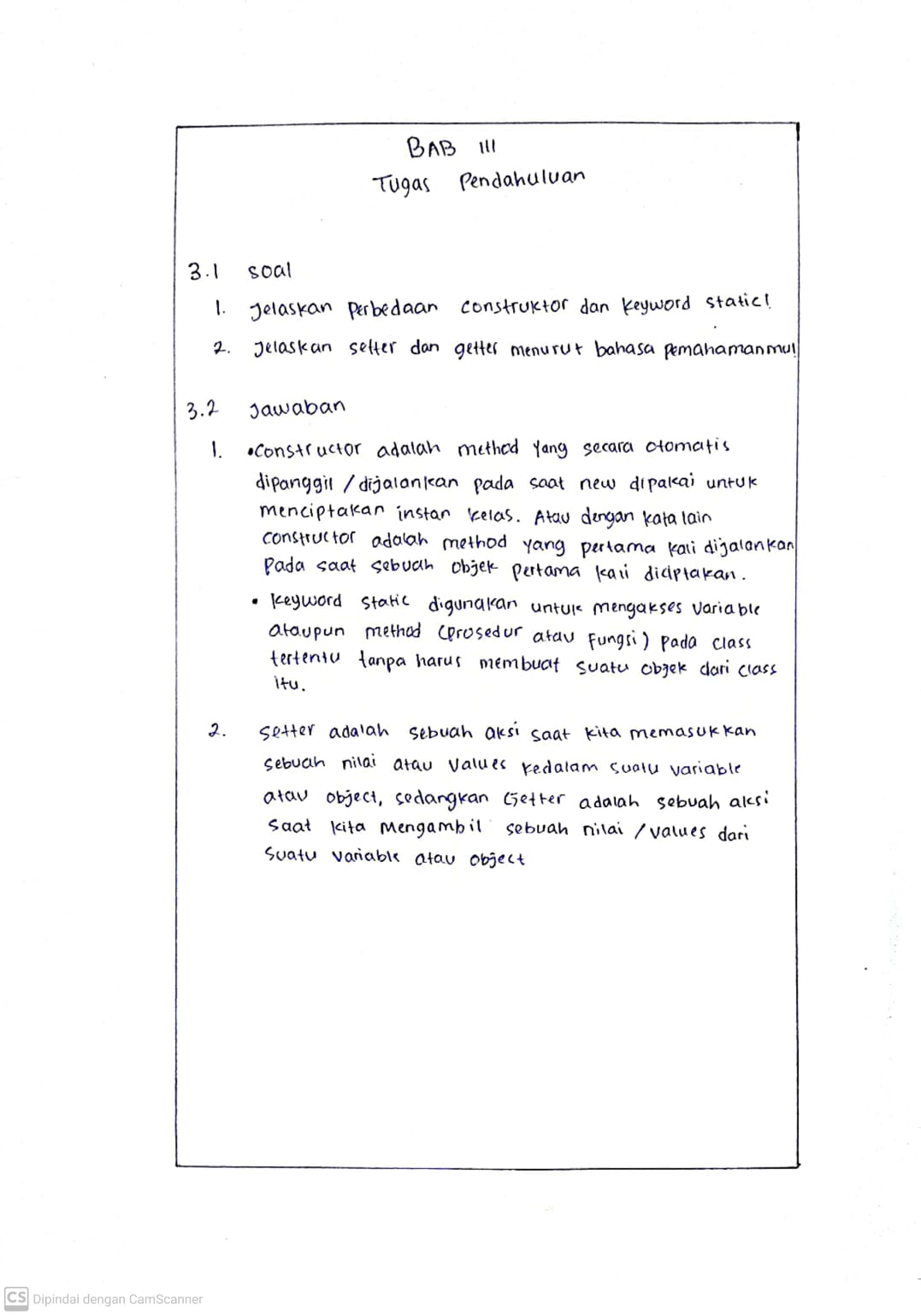
1. **Penjelasan Program**
2. **Class Makanan**

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

1. Baris 2, keyword static yang berupa method
2. Baris 3-4, mencetak nama makanan dan harga.
3. **Class MakananKhasMadura**

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

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

****

# **BAB IV IMPLEMENTASI**

## **4.1 Soal**

1. Buatlah class manusia dengan dengan attribut nama, umur dan alamat serta memiliki method berjalan dan berlari. Buatlah beberapa objek dari class tersebut.
2. Buatlah program java untuk menampilkan nama, nim, jurusan/prodi, dan alamat mahasiswa menggunakan konsep OOP. Nama, nim, jurusan/prodi, dan alamat bersifat dinamis, dengan kata lain menggunakan input dari pengguna.

## **4.2 Source code**

package universitas;

import java.util.ArrayList;

import java.util.Scanner;

/\*\*

\*

\* @author RIZQY

\*/

public class UNIVERSITAS{

private String nama;

private String nim;

private String jurusan;

private String alamat;

public UNIVERSITAS(String nama, String nim, String jurusan, String alamat) {

this.nama = nama;

this.nim = nim;

this.jurusan = jurusan;

this.alamat = alamat;

}

//setter untuk memanngil nilai

public void setNama(String nama){

this.nama = nama;

}

//getter untuk menambah nilai

public String getNama(){

return nama ;

}

//setter untuk nim

public void setNim(String nim){

this.nim = nim;

}

//getter untuk nim

public String getNim(){

return nim ;

}

//setter untuk jurusan

public void setJurusan(String jurusan){

this.jurusan = jurusan;

}

//getter untuk jurusan

public String getJurusan(){

return jurusan ;

}

//setter untuk alamat

public void setAlamat(String alamat){

this.alamat = alamat;

}

//getter untuk alamat

public String getAlamat(){

return alamat ;

}

public static void main(String[] args) {

Scanner baru = new Scanner(System.in);

ArrayList<UNIVERSITAS> daftarMahasiswa = new ArrayList<>();

String inputLagi;

do {

System.out.print("Masukkan NIM: ");

String nim = baru.next();

baru.nextLine();

System.out.print("Masukkan Nama: ");

String nama = baru.nextLine();

System.out.print("Masukkan Alamat: ");

String alamat = baru.nextLine();

System.out.println("Pilihan Jurusan:");

System.out.println("41 = TEKNIK INFORMATIKA");

System.out.println("42 = TEKNIK INDUSTRI");

System.out.println("43 = TEKNIK ELEKTRO");

System.out.println("44 = SISTEM INFORMASI");

System.out.println("48 = TEKNIK MESIN");

System.out.println("49 = TEKNIK MEKATRONIKA");

System.out.print("Masukkan Kode Jurusan: ");

String kodeJurusan = baru.next();

String jurusan;

// Menggunakan switch-case konvensional

switch (kodeJurusan) {

case "41":

jurusan = "TEKNIK INFORMATIKA";

break;

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;

default:

jurusan = "jurusan yang anda pilih tidak ada, pilih kode jurusan dengan benar";

}

UNIVERSITAS mahasiswa = new UNIVERSITAS(nim, nama, alamat, jurusan);

daftarMahasiswa.add(mahasiswa);

System.out.println("\nApakah Anda ingin memasukkan data lagi? (Y/T): ");

inputLagi = baru.next();

} while (inputLagi.equalsIgnoreCase("Y"));

System.out.println("\nDaftar Mahasiswa:");

for (UNIVERSITAS mahasiswa1 : daftarMahasiswa) {

System.out.println("NIM: " + mahasiswa1.getNim());

System.out.println("Nama: " + mahasiswa1.getNama());

System.out.println("Alamat: " + mahasiswa1.getAlamat());

System.out.println("Jurusan: " + mahasiswa1.getJurusan());

}

}

}

## **4.3 Hasil**

## **4.4 Penjelasan**

Ini adalah kelas yang merepresentasikan objek mahasiswa. Setiap objek Mahasiswa memiliki atribut seperti nim, nama, alamat, dan jurusan. sebuah constructor yang digunakan untuk membuat objek Mahasiswa Program meminta untuk memasukkan data mahasiswa seperti nim, nama, alamat, dan kode jurusan. Setiap data mahasiswa yang dimasukkan oleh pengguna akan dibuatkan objek Mahasiswa baru dengan menggunakan program ini, pengguna dapat memasukkan data mahasiswa dan melihat daftar mahasiswa yang telah dimasukkan ke dalam sistem dengan mudah.

# **BAB V PENUTUP**

## **Analisa**

Analisis konstruktor dan kata kunci static dalam pemrograman berorientasi objek (OOP) memberikan pemahaman yang mendalam tentang bagaimana kelas bekerja, bagaimana objek dibuat, dan bagaimana data dan perilaku bersama dikelola. Konstruktor bertanggung jawab untuk menginisialisasi objek dari suatu kelas. Ini dilakukan saat objek pertama kali dibuat dengan menggunakan operator new. : Metode static sering digunakan untuk menyediakan fungsi utilitas yang dapat diakses tanpa perlu membuat objek baru. Contohnya adalah metode utilitas matematika, operasi string, dan sebagainya.

Variabel static digunakan untuk menyimpan data bersama yang dibagikan oleh semua objek dari kelas tersebut. Misalnya, hitungan jumlah objek yang telah dibuat dari kelas, atau konstanta yang sama untuk semua objek.

## **Kesimpulan**

* Konstruktor dan kata kunci static adalah fitur kunci dalam OOP yang menyediakan cara untuk mengelola inisialisasi objek dan data bersama di antara objek-objek dari kelas yang sama.
* Konstruktor memungkinkan inisialisasi objek dengan nilai-nilai awal yang diperlukan, sedangkan static digunakan untuk menyediakan variabel dan metode yang bersama-sama digunakan oleh semua objek dari kelas.
* Dengan menggunakan konstruktor dan kata kunci static dengan bijaksana, pengembang dapat membangun aplikasi yang lebih efisien dan mudah dimengerti dengan memanfaatkan paradigma pemrograman berorientasi objek secara penuh.