**LAPORAN RESMI**

**MODUL II**

**(CONSTRUCTOR DAN KEYWORD STATIC)**

**PEMROGRAMAN BERBASIS OBJEK**

**Disetujui :.. ……. 2023**

**Asisten**

**SYAHRUL RAMADHANI**

**NIM. 22.04.411.00128**

**NAMA : CYNDIFANITA B’THARI MARSHA**

**N.R.P : 230441100109**

**DOSEN : AHMAD ZAIN NUR S.Kom M.Kom**

**ASISTEN : SYAHRUL RAMADHANI**

**TGL PRAKTIKUM : 26 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**

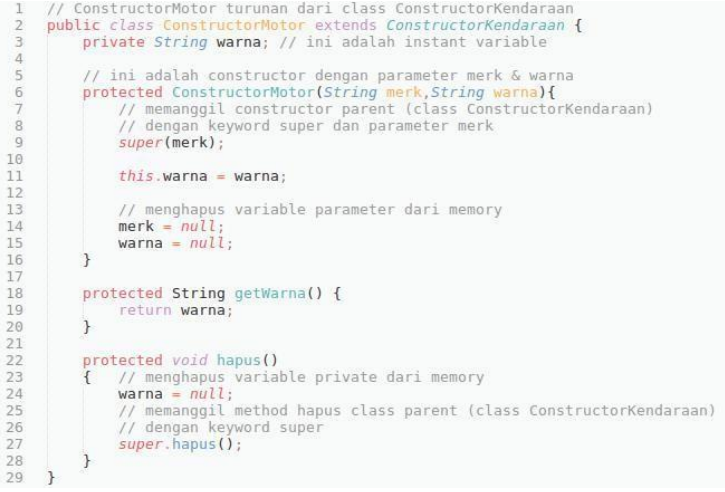
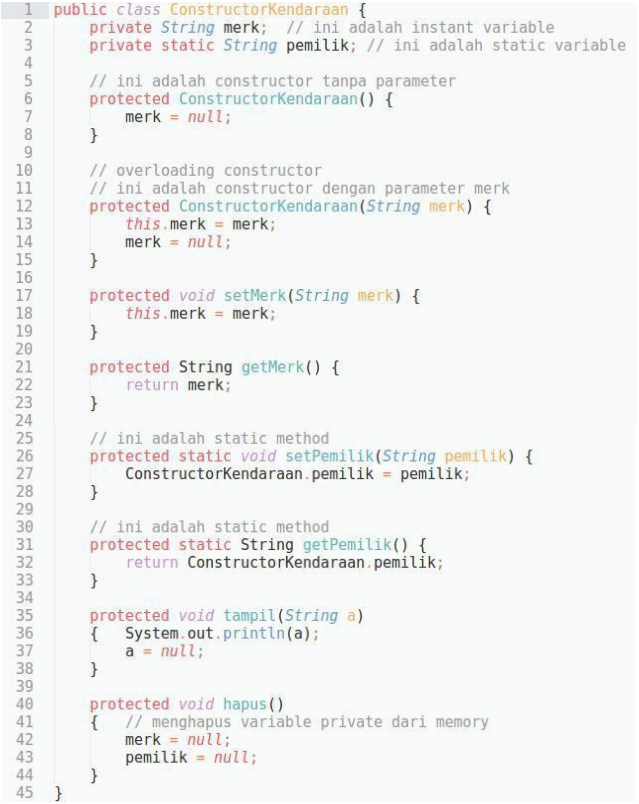
Constructor adalah method yang secara otomatis dipanggil/dijalankan pada saat new dipakai untuk menciptakan instan kelas. Atau dengan kata lain constructor adalah method yang pertama kali dijalankan pada saat sebuah objek pertama kali diciptakan. Jika dalam sebuah class tidak terdapat constructor, maka 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.

b.Tidak ada return type yang diberikan kedalam Constructor Signature.

c.Tidak ada return statement, di dalam tubuh constructor

1. Contoh ConstructorMotor.java



1. MainConstructorMotor.java

****

****



## **Keyword Static**

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.

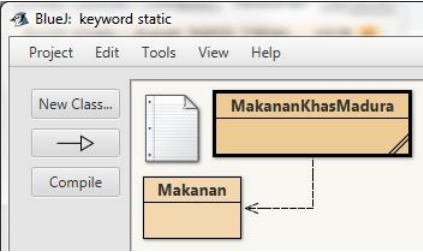
Static member adalah variable atau method yang bukan milik dari suatu object, melainkan milik dari class.

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.

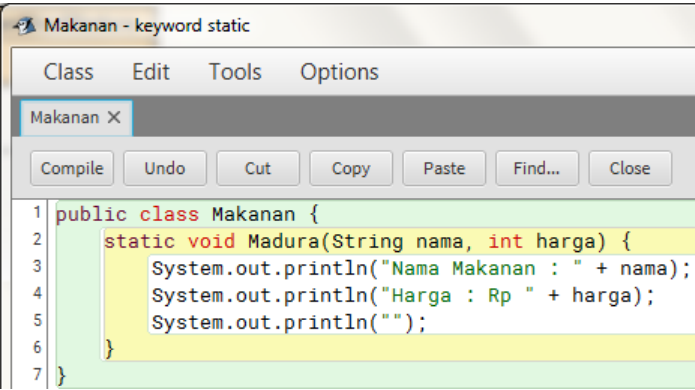
****

1. Class Makanan
2. Class MakananKhasMadura

Tampilan Class

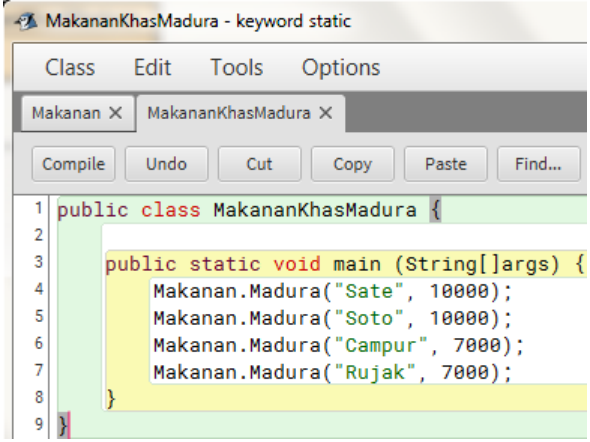
Class Makanan

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

****

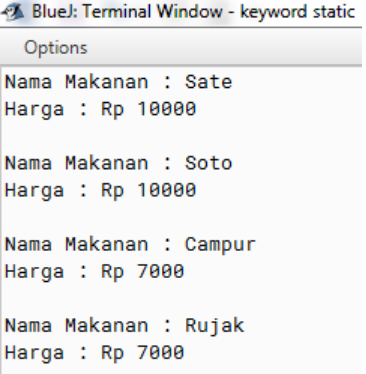
Class Makanan Khas Madura

Pada class Makanan Khas Madura, 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!

****

Penjelasan Program

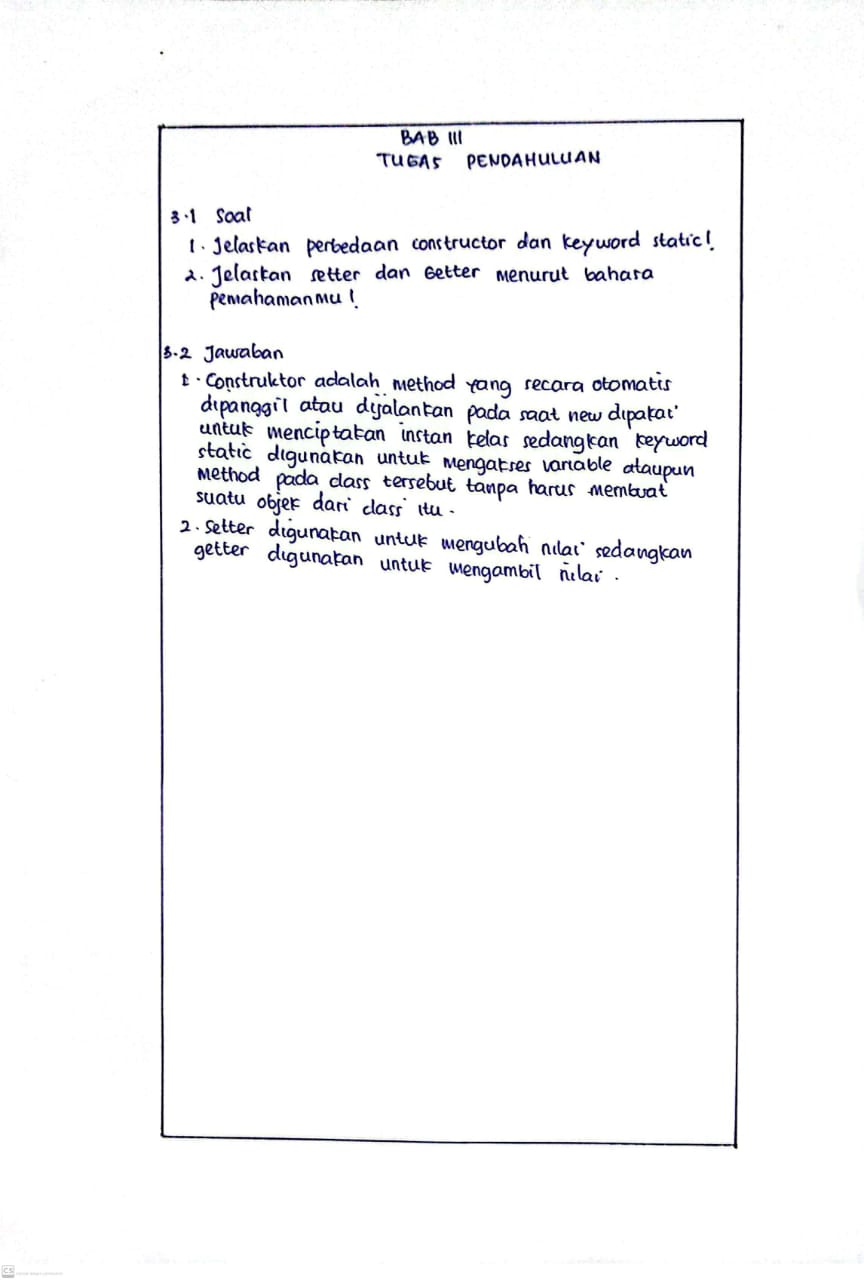
Class Makanan, Pada class Makanan, terdapat method static dengan parameter makanan dan harga.

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

Class Makanan Khas Madura

Pada class Makanan Khas Madura, 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 Source code**

**1. Program mahasiswa2**

public class MahasiswaMODUL2 {

private String Nama, NIM, Jurusan, Alamat;

//Konstruktor

public MahasiswaMODUL2(String Namaa, String NIMM, String Jurusann, String Alamatt){

this.Nama = Namaa;

this.NIM = NIMM;

this.Jurusan = Jurusann;

this.Alamat = Alamatt;

}

//Getter

public String getNama(){

return Nama;

}

public String getNIM(){

return NIM;

}

public String getJurusan(){

return Jurusan;

}

public String getAlamat(){

return Alamat;

}

//Setter

public void setNama(String Nama){

this.Nama = Nama;

}

public void setNIM(String NIM){

this.NIM = NIM;

}

public void setJurusan(String Jurusan){

this.Jurusan = Jurusan;

}

public void setAlamat(String Alamat){

this.Alamat = Alamat;

}

public static void main(String[] args) {

MahasiswaMODUL2 mhs = new MahasiswaMODUL2("Fanita", "230441100109", "Sistem Informasi", "bojonegoro");

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

System.out.println("NIM: " + mhs.getNIM());

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

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

}

}

**2.Program MahasiswaSoal2**

package MahasiswaModul2Soal2;

/\*\*

\*

\* @author HP

\*/

public class MahasiswaModul2Soal2 {

String namaa;

String niim;

String jurusanProdii;

String alamatt;

String[] ukmm; // Atribut untuk menyimpan UKM

static String universitas;

public MahasiswaModul2Soal2(String nama, String nim, String jurusanProdi, String alamat) {

namaa = nama;

niim = nim;

jurusanProdii = jurusanProdi;

alamatt = alamat;

}

// Getter dan setter untuk atribut non-static

public String getNama() {

return namaa;

}

public void setNama(String nama) {

namaa = nama;

}

public String getNim() {

return niim;

}

public void setNim(String nim) {

niim = nim;

}

public String getJurusanProdi() {

return jurusanProdii;

}

public void setJurusanProdi(String jurusanProdi) {

jurusanProdii = jurusanProdi;

}

public String getAlamat() {

return alamatt;

}

public void setAlamat(String alamat) {

alamatt = alamat;

}

// Getter dan setter untuk atribut static universitas

public static String getUniversitas() {

return universitas;

}

public static void setUniversitas(String universitas) {

MahasiswaModul2Soal2.universitas = universitas;

}

// Getter dan setter untuk atribut ukm

public String[] getUkm() {

return ukmm;

}

public void setUkm(String[] ukm) {

ukmm = ukm;

}

// Method static untuk menampilkan nilai atribut static universitas

public static void displayUniversitas() {

System.out.println("Universitas: " + universitas);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

// Pengaturan nilai atribut static universitas

MahasiswaModul2Soal2.setUniversitas("Universitas ABC");

System.out.println("Universitas: " + universitas);

// Membuat objek mahasiswa

MahasiswaModul2Soal2 mhs = new MahasiswaModul2Soal2("John Doe", "123456", "Teknik Informatika", "Jl. Contoh No. 123");

// Menambahkan UKM

String[] ukm = {"ITC", "pramuka"};

mhs.setUkm(ukm);

// Menampilkan informasi mahasiswa

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

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

System.out.println("Jurusan/Prodi: " + mhs.getJurusanProdi());

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

// Menampilkan UKM yang diikuti mahasiswa

System.out.println("UKM:");

for (String namaUKM : mhs.getUkm()) {

System.out.println("- " + namaUKM);

}

// Menampilkan nilai atribut static universitas

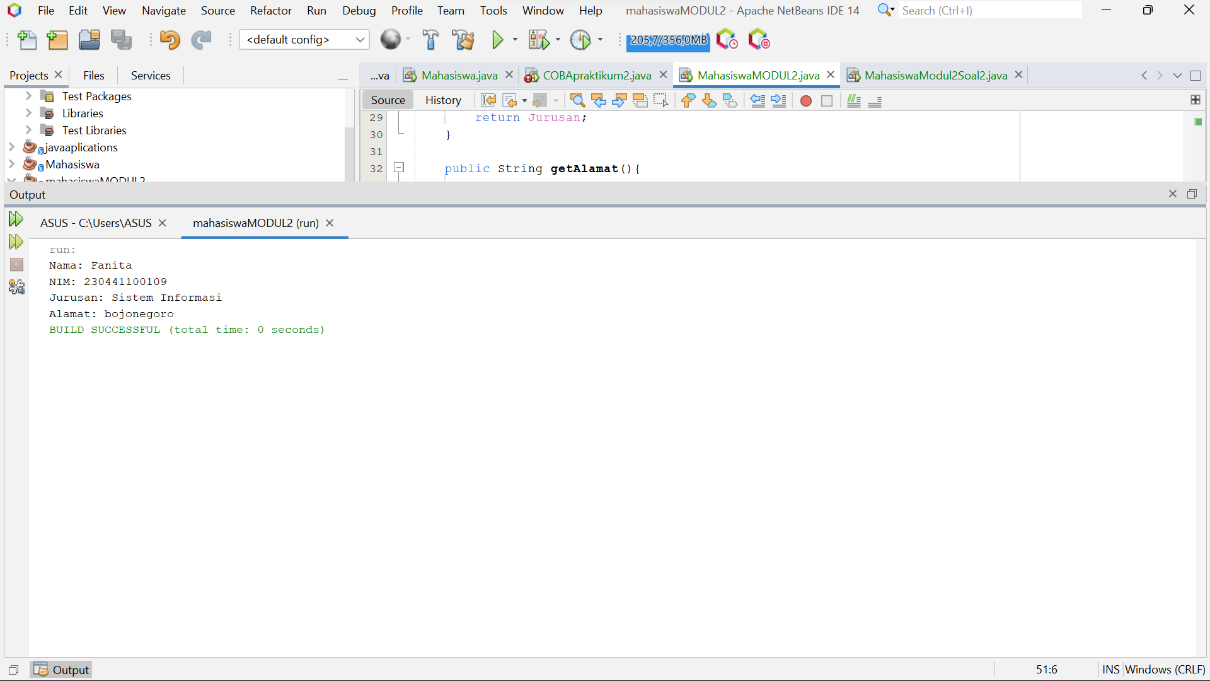
MahasiswaModul2Soal2.displayUniversitas();

}

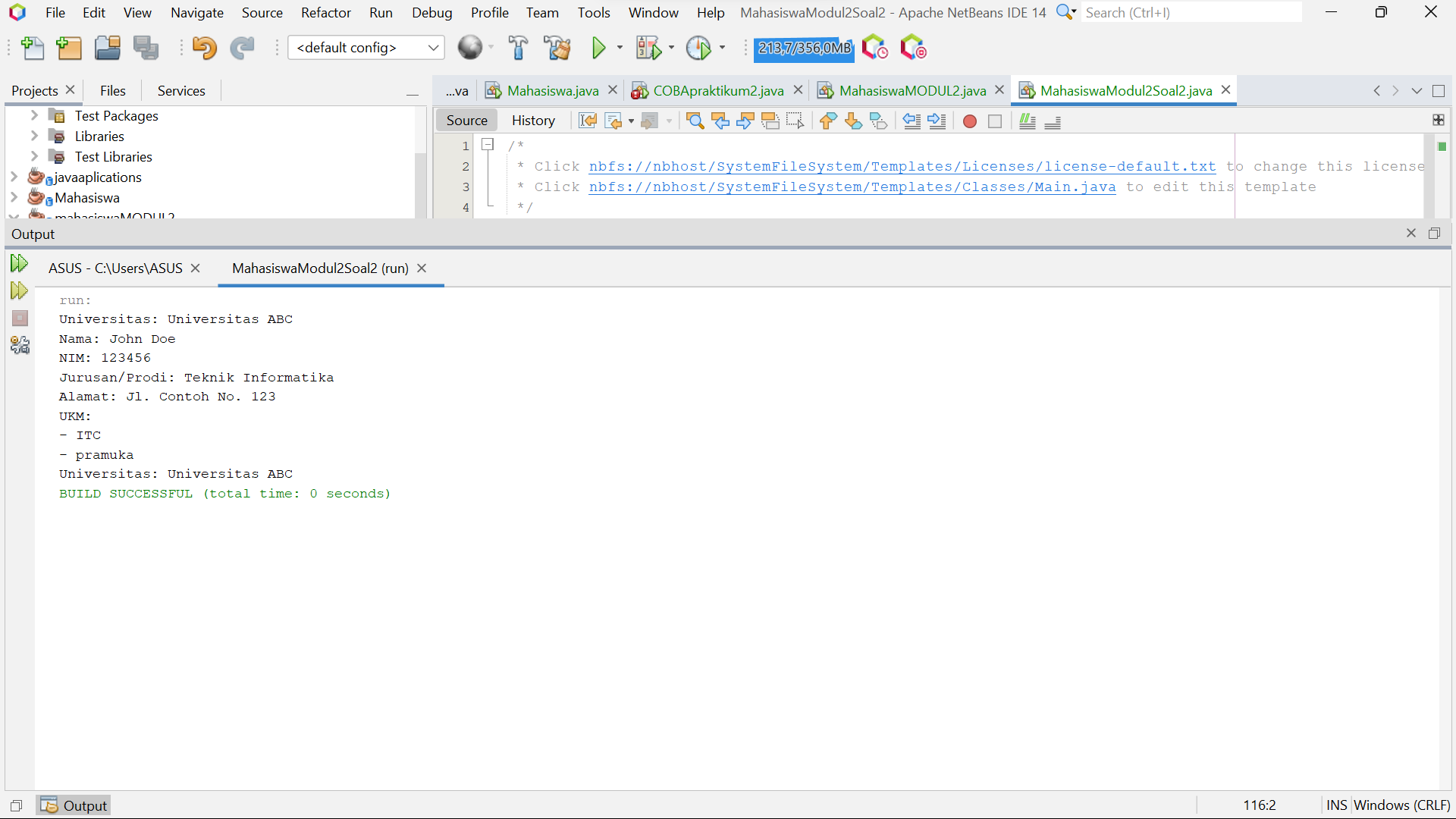
}

## **4.2 Hasil**

1. **Hasil mahasiswa2**



1. **Hasil mahasiswaSoal2**



Penjelasan :

* Melalui implementasi kelas Mahasiswa2 dalam bahasa pemrograman Java, kita dapat memperoleh pemahaman yang lebih baik tentang konsep dasar pemrograman berorientasi objek.
* Penggunaan atribut, metode, dan konstruktor dalam kelas tersebut memungkinkan kita untuk merepresentasikan dan mengelola informasi terkait mahasiswa dengan lebih terstruktur. Dengan adanya getter dan setter, kita dapat mengakses dan mengubah nilai atribut secara terkontrol, sehingga memungkinkan fleksibilitas dalam manipulasi data.
* Selain itu, penggunaan atribut static universitas memperlihatkan bagaimana nilai yang bersifat universal dapat dibagikan di antara seluruh instance dari kelas yang sama.
* Keseluruhan implementasi ini memberikan gambaran tentang cara memodelkan dan mengorganisir entitas dalam program, serta bagaimana berbagai konsep dasar pemrograman objek, seperti enkapsulasi dan penggunaan array, dapat diterapkan dalam praktek.

# **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**

1. 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.
2. 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.
3. 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.