**LAPORAN RESMI**

**MODUL I**

**OBJECT DAN CLASS**

**PEMPROGRAMAN BERBASIS OBJEK**



**NAMA : FIRDAUS NUR HIDAYATULLAH**

**N.R.P : 230441100023**

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

**ASISTEN : DWI ANGGA VIRGI DANIANTO**

**TGL PRAKTIKUM : 23 MARET 2024**

**Disetujui : 2024**

**Asisten**

**DWI ANGGA VIRGI DANIANTO**

**20.04.411.00174**

**LABORATORIUM BISNIS INTELIJEN SISTEM**

**PRODI SISTEM INFORMASI**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

# BAB I PENDAHULUAN

## Latar Belakang

Dengan berkembangannya Teknologi Informasi, yang membutuhkan kecepatan dan ketepatan dalam berbagai aspek kehidupan. Kita dituntuk untuk bergerak cepat dalam segala hal, termasuk juga dalam bidang komunikasi informasi. Teknologi sangat berperan penting dalam kehidupan sehari – hari.

Dengan adanya kemajuan Teknologi di dunia ini akal pikiran manusia semakin mahir dalam menciptakan sesuatu yang bermanfaat bagi kehidupan mereka, salah satunya dalam pembuatan program – program yang sangat membantu mereka dalam menyelesaikan pekerjaan mereka dengan cepat, baik dan memuaskan, maka dari itu menyusunan laporan ini mengenai Object dan Class menggunakan Bahasa pemprograman Java ini juga merupakan salah satu bagian dari kemajuan zaman yang terlihat manfaatnya.

## Tujuan

* Mahasiswa mampu memahami konsep Class dan Objectdalam Pemrograman Berorientasi Objek serta mampu mengimplementasikannya.
* Mahasiswa mampu memahami atribut dan Methoddalam Pemrograman Berorientasi Objek serta mampu mengimplementasikannya.

# BAB II DASAR TEORI

## **Class dan Objek**

Class diumpamakan seperti cetakan yang berguna untuk mencetak suatu objek, classjuga merupakan grup suatu objek dengan kemiripan attribut, brhaviour dan relasi ke objek lain. Contoh dari sebuah classtersebut misalnya cetakan dari sebuah mobil sedan. Mobil sedan tentunya mempunyai konsep, dan konsep inilah dipakai untuk membentuk/mencetak hingga menjadi mobil sedan. Oleh karena itu konsep bisa diartikan sebagai class.

Object adalah instance dari class. Jika classsecara umum merepresentasikan sebuah object, sebuah instance adalah representasi nyata dari class itu sendiri. Contoh : Dari class Kendaraan, maka kita dapat membuat objek sepeda, motor, mobil, becak dll.

Pada dasarnya ada dua karakteristik yang utama pada sebuah object yaitu :

1. Setiap object memiliki attribut sebagai status yan akan disebut state.
2. Setiap object memiliki tingkah laku yang kemudian akan disebut dengan method (behaviour).

| **State** | **Behaviour** |
| --- | --- |
| Pedal | Kecepatannya menaik |
| Roda | Kecepatannya menurun |
| Jeruji | Perpindahan gigi sepeda |

Object sepeda

Dalam pengembangan perangkat lunak berorientasi object, object dalam perangkat lunak akan menyimpan state-nya dalam variabel dan menyimpan informasi tingkah laku (behaviour) dalam method – method atau fungsi – fungsi.

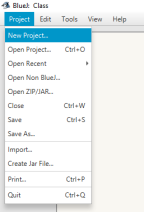
Untuk membuat object, kita menggunakan perintah new dengan sebuah nama class yang dibuat sebagai instance dari class.

Contoh :

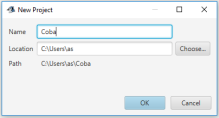
1. **Tahapan membuat suatu class.**

Untuk membuat suatu class ada beberapa tahap yang harus dilakukan seperti yang ada pada Gambar.1 dengan rincian tahapan sebagai berikut :

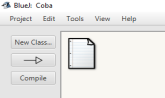
1. **Buka aplikasi BlueJ, klik project pilih “New Project”.**

****

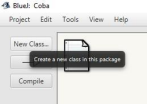
1. **Pada form new project isilah nama project yang akan kita buat.**

****

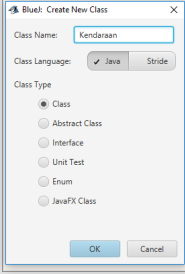
1. Tampil form setelah nama project diisi.



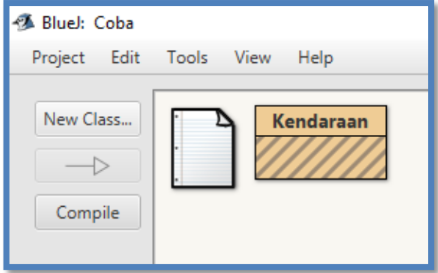
1. Klick “**New Class**”, untuk membuat class.

**

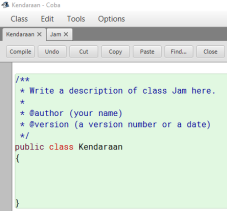
1. Akan ada form baru yang digunakan untuk mengisi nama Class sesuai dengan yang kita inginkan pada kolom “**Class Name**”, lalu tekan “Oke”.



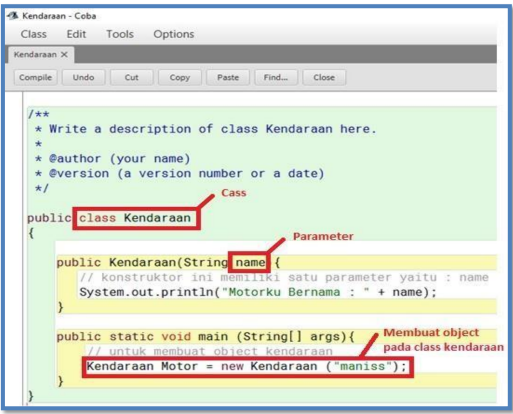
1. Tampilkan from setelah kita membuat class, dan kita bisa mengedit code di dalam class, pada contoh diatas terdapat pada class kendaraan.



1. Tampilkan class yang didalamnya bisa kita edit sesuai dengan yang kita inginkan



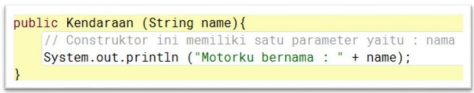
1. Contoh membuat object dari suatu class



Gambar 2. Source Code

Pada Gambar.2 merupakan contoh dari suatu class kendaraan yang didalamnya terdapat suatu object. Berdasarkan program diatas, telah dibuat object dan motor dari class kendaraan, dan kita juga bisa menggunakan construktor yang akan dijalankan secara otomatis pada saat object dibuat yakni ketika perintah “new” dijalankan.

Contoh Construktor seperti berikut :

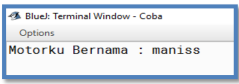


Source code diatas disebut dengan construktor yang nantinya akan dipanggil seara otomatis pada saat object dibuat, tanpa membuat method terlebih dahulu. Construktor harus menggunakan nama class untuk penamaannya,dan dalam suatu class hanya bole ada satu construktor, jika tidak maka terjadi SyntaxError.

Contoh object seperti berikut :



Kode program diatas artinya membuat suatu object motor dan class kendaraan, dengan menggunakan keyboard new. Karena menggunakan construktor maka tidak perlu menggunakan method untuk menginisialisasi object.



Gambar 3. Output dari Class Kendaraan

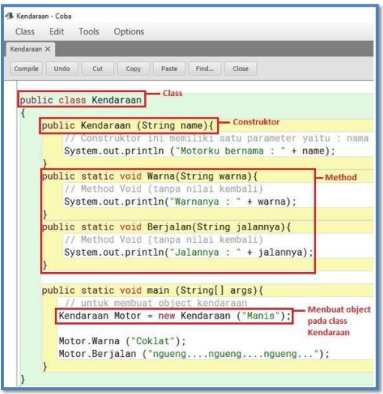
## **2.2 Method**

Method merupakan kumpulan dari fungsi yang kita buat dalam suatu class, yang kemundian fungsi – fungsi itu akan di panggil kembali., dengan method kita bisa mempersingkat coding yang kita buat. Method didefinikan pada class tetapi akan di panggil melalui object, method menentukan perilaku object yakni apa yang akan terjadi saat object itu dibuat serta berbagai operasi yang akan dapat dilakukan object.

Struktur dari method diantaranya sebagai berikut :

1. Method terdiri dari statement public, private, protected, yang menandakan hak kases method tersbut.
2. Nama method terdiri dari alfabeth saja.
3. Parameter, dalam penulisannya harus diawali dengan tipe data baru nama parameternya.
4. Isi method.

Contoh :



Gambar 4.

Penjelasan Gambar 1 :

1. Pada class kendaraan terdapat Construktor, Mentod, dan Object
2. Construktor merupakan suatu methodyang akan memberikan nilai awal pada saat suatu objectdibuat. Jadi apabila objectdiatas telah dibuat lalu dijalankan maka construktor akan bekerja secara otomatis.Biasanya nama construktor sama dengan nama classyang dibuat.
3. Pada method diatas terdapat (“public static”) yang merupakan modifier yaitu jenis akses method yang bersifat public. Pada gambar diatas ada dua method yang nantinya akan dipanggil saat object dijalankan yaitu methodWarna dan methodBerjalan.
4. Kode program dibawah artinya : membuat sebuah objectMotor dari class Kendaraan dengan menggunakan keyword new.



1. Kode program dibawah artinya : menginisialisasi object Motor menggunakan method Warna dan method Berjalan yang memiliki class kendaraan yang nantinya akan dipanggil oleh object.



## Attribut

Attribut dari sebuah kelas adalah variabel global yang dimiliki sebua kelas, Attribut dapat memiliki hak akses private, public maupun protected. Sebuah attribut yang ditanyakan sebagai private hanya dapat diakses secara langsung oleh kelas yang membungkusnya, sedangkan kelas lainnya tidak dapat mengakses attribut ini secara berlangsung.

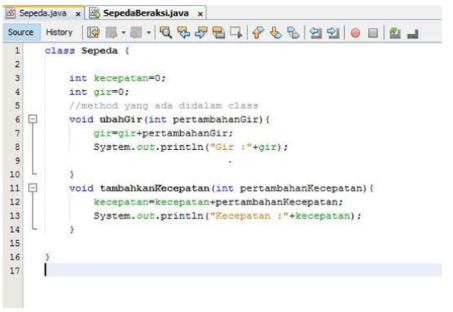
Sebuah atribut yang dinyatakan sebagai public dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya. Sebuah atribut yang dinyatakan sebagai protected tidak dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya, kecuali kelas yang mengaksesnya adalah kelas turunan dari kelas yang membungkusnya.

* **Karakteristik Atribut**

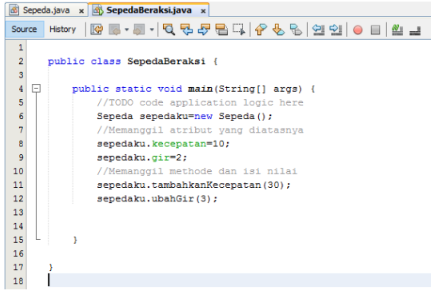
Dalam class, atribut disebut sebagai variabel. Atribut dapat membedakan antara satu object dengan object yang lain. Sebagai contoh, pada class: mahasiswa, terdapat object mahasiswa si A, dan object mahasiswa si B. Yang membedakan antara object si A dan si B adalah NPM-nya (Nomor Pokok Mahasiswa) yang merupakan atribut dari object tersebut. Pada atribut, terdapat pula dua istilahvariabel, yaitu Instance Variabel dan Class Variable.

* **Contoh penerapan atribut**

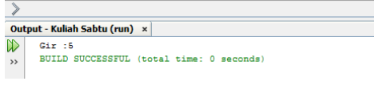
Membuat class dengan nama sepeda dan semua atribut yang dimilik oleh kelas sepeda (Gambar 1).

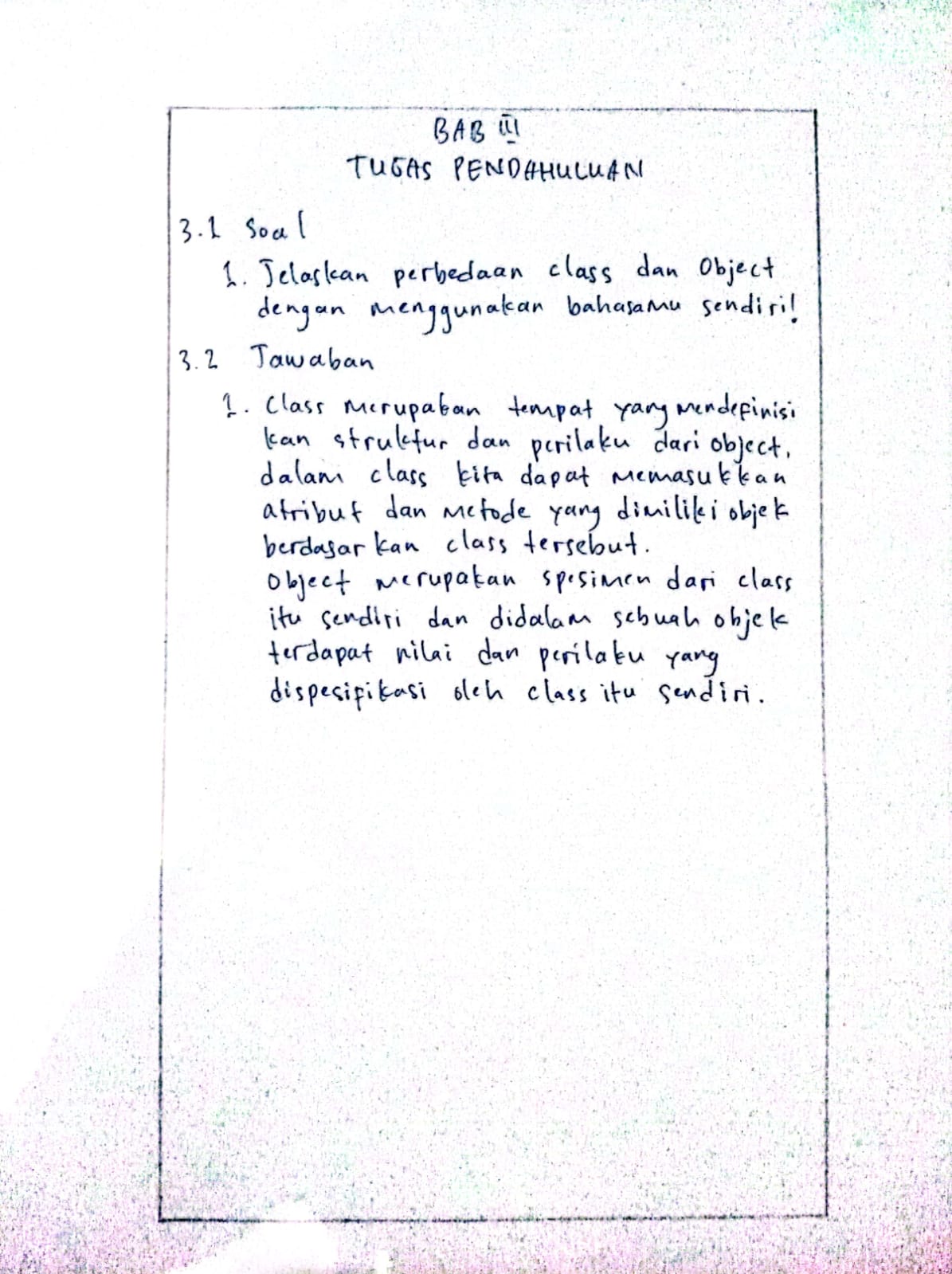


Membuat public class yang memanfaatkan / memanggil atribut dalam kelas lainnya (Gambar 2).



Hasil program setelah dijalankan (Gambar 3).





# BAB IV IMPLEMENTASI

## **Source Code**

1. **Buatlah class manusia dengan attribut nama, umur, dan alamat serta memiliki method berjalan dan berlari. Buatlah beberapa objek dari class tersebut.**

**/\***

**\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license**

**\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template**

**\*/**

**package manusia;**

**/\*\***

**\***

**\* @author ASUS**

**\*/**

**public class human {**

**String nama;**

**int umur;**

**String alamat;**

**void berjalan(){**

**System.out.println("Firdaus sedang berjalan");**

**}**

**void berlari(){**

**System.out.println("Razan sedang berlari");**

**}**

**void Allatt(){**

**System.out.println("Nama Saya adalah :"+nama);**

**System.out.println("Umur saya adalah :"+umur);**

**System.out.println("Alamat saya adalah :"+alamat);**

**}**

**}**

**/\***

**\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license**

**\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template**

**\*/**

**package manusia;**

**/\*\***

**\***

**\* @author ASUS**

**\*/**

**public class Manusia {**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String[] args) {**

**// TODO code application logic here**

**human hidup=new human();**

**hidup.nama="Firdaus";**

**hidup.umur=19;**

**hidup.alamat="Surabaya";**

**human urip=new human();**

**urip.nama="Razan";**

**urip.umur=63;**

**urip.alamat="Sumenep";**

**hidup.Allatt();**

**hidup.berjalan();**

**urip.Allatt();**

**urip.berlari();**

**}**

**}**

**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.**

**/\***

**\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license**

**\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template**

**\*/**

**package main;**

**/\*\***

**\***

**\* @author ASUS**

**\*/**

**public class Mahasiswa {**

**String Nama;**

**long NIM;**

**String Jurusan;**

**String Alamat;**

**public Mahasiswa(String inputNama, long inputNIM, String inputJurusan, String inputAlamat){**

**//menyimpan data yg sudah diinputkan agar bisa outputkan kembali**

**Nama = inputNama;**

**NIM = inputNIM;**

**Jurusan = inputJurusan;**

**Alamat = inputAlamat;**

**}**

**public void Allinfo(){**

**System.out.println("Nama :"+Nama);**

**System.out.println("NIM :"+NIM);**

**System.out.println("Jurusan :"+Jurusan);**

**System.out.println("Alamat :"+Alamat);**

**}**

**}**

**/\***

**\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license**

**\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template**

**\*/**

**package main;**

**import java.util.Scanner;**

**/\*\***

**\***

**\* @author ASUS**

**\*/**

**public class Main {**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String[] args) {**

**Scanner scanner=new Scanner(System.in);**

**System.out.println("Masukkan nama Mahasiswa :");**

**String name = scanner.nextLine();**

**System.out.println("Masukkan NIM Mahasiswa :");**

**long nim = scanner.nextLong();**

**scanner.nextLine();**

**System.out.println("Masukkan Jurusan mahasiswa :");**

**String Jurusan = scanner.nextLine();**

**System.out.println("Masukkan alamat rumah mahasiswa :");**

**String alamat = scanner.nextLine();**

**Mahasiswa mahasiswa=new Mahasiswa(name, nim, Jurusan, alamat);**

**System.out.println("\nStudent Data");**

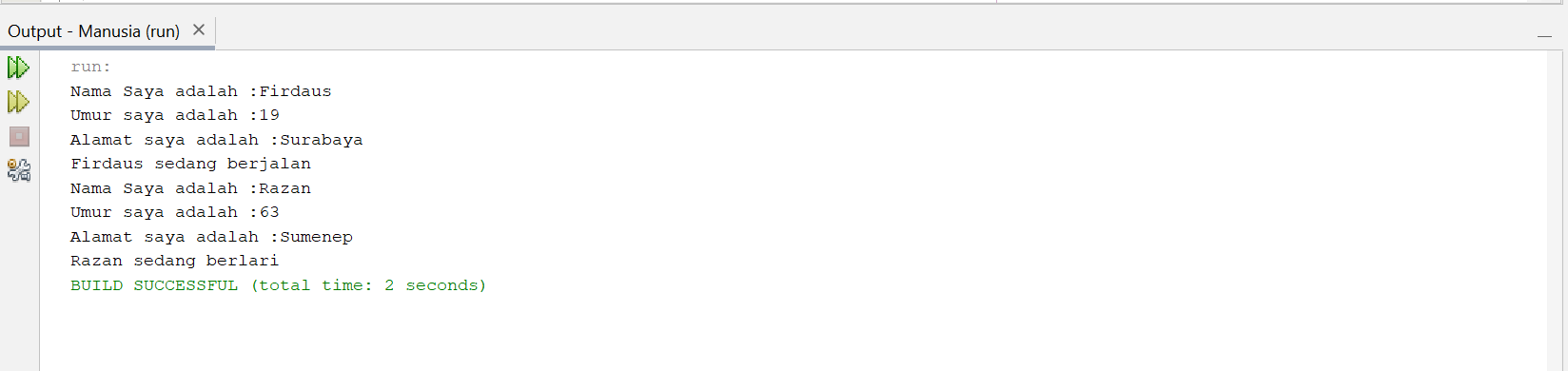
**mahasiswa.Allinfo();**

**}**

**}**

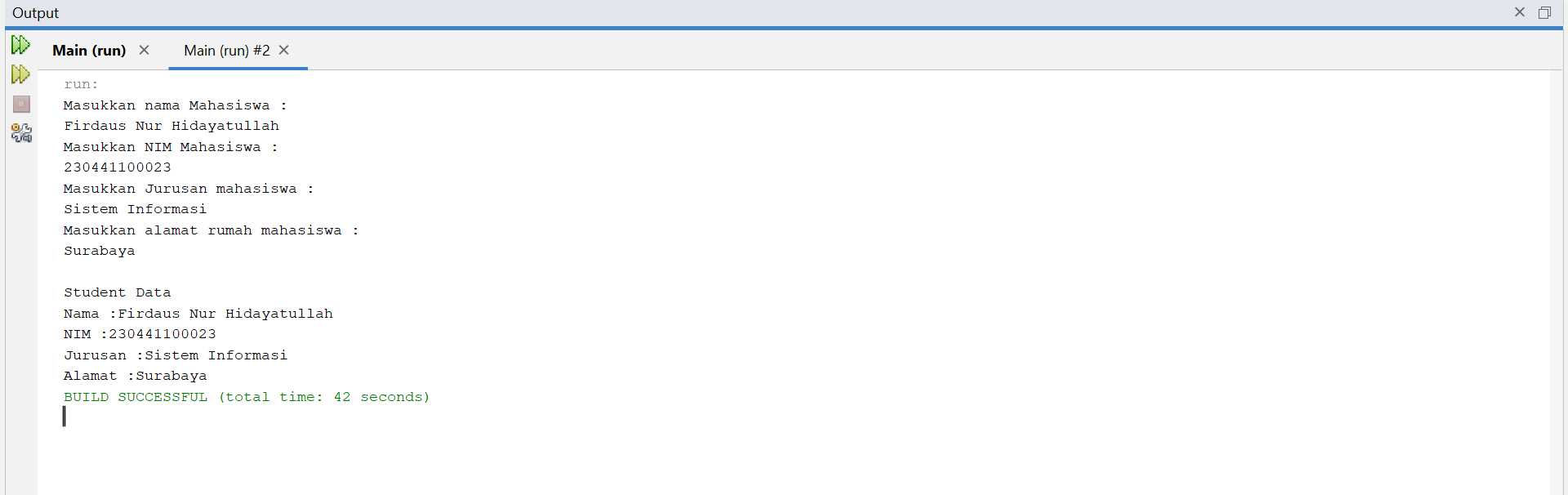
## **Hasil**

1. **Output No 1**

****

* Penjelasan

1. Package dan komentar : Program dimulai dengan deklarasi tugas1\_modul1 Komentar di awal file memberikan petunjuk kepada pengguna tentang bagaimana mengubah lisensi dan mengedit template kelas.
2. Kelas Manusia: Diikuti dengan definisi kelas Manusia. Kelas ini memiliki tiga atribut yaitu nama, umur, dan alamat. Selain itu, kelas ini memiliki dua metode yaitu berjalan() dan berlari() yang digunakan untuk menampilkan pesan bahwa objek sedang berjalan atau berlari. Metode berjalan() dan berlari() menggunakan nilai dari atribut nama objek yang memanggilnya.
3. Kelas Tugas1\_modul 1: Ini adalah kelas utama yang memiliki metode main(). Di dalam metode main(), dua objek dari kelas Manusia dibuat yaitu manusia1 dan manusia2. Atribut-atribut objek-objek tersebut diinisialisasi dengan nilai tertentu. Kemudian, nilai-nilai atribut tersebut dicetak ke layar bersama dengan pesan yang menunjukkan objek sedang berjalan atau berlari. Objek manusia1 dan manusia2 memiliki atribut yang sama karena diinisialisasi dengan nilai yang sama, yang berarti kelas Manusia dapat membuat banyak objek dengan perilaku yang sama.
4. **Output No 2**

****

* Penjelasan

1. Baris pertama mendeklarasikan `package` atau paket yang merupakan kumpulan dari kelas-kelas (classes) yang terkait. Nama paket di sini adalah ` tugas2;.
2. Baris kedua mengimpor kelas `Scanner` dari package `java.util`. Kelas `Scanner` digunakan untuk membaca input dari pengguna.
3. Ini adalah definisi public class `modul1` yang memiliki empat atribut: `nama`, `nim`, `jurusan`, dan `alamat`. Semua atribut ini adalah variabel yang akan menyimpan data tentang mahasiswa.
4. `Scanner input = new Scanner(System.in);` digunakan untuk membuat instance dari `Scanner` yang membaca input dari pengguna melalui konsol.
5. Objek `mahasiswa` dari kelas `modul1` dibuat. Program kemudian meminta pengguna untuk memasukkan nilai atribut-atribut objek tersebut: `nama`, `nim`, `jurusan`, dan `alamat`.
6. `input.nextLine()` digunakan untuk membaca `String` yang dimasukkan, sementara `input.nextLong()` untuk membaca nilai `long`. Setelah membaca `long`, `input.nextLine()` dipanggil lagi untuk membersihkan buffer (mengonsumsi newline character yang tersisa setelah memasukkan `long`).
7. Terakhir, System.out.println = DATA MAHASISWA =, nama, prodi, alamat, dan nim untuk menampilkan data yang telah dimasukkan oleh pengguna.

# BAB V PENUTUP

## Analisa

Class dan Object adalah konsep fundamental pemprograman berorientasi object (oop), yang memungkinkan pemodelan yang lebih efektif dan modular dari sistem – sistem kompleks. Kelas memungkinkan abstraksi dan reusabili tas kode, dengan memungkinkan pengelompokan variabel dan fungsi – fungsi terkait dalam sebuah unit tunggal yang dapat digunakan kembali.

Object memungkinkan penciptaan variasi dari suatu kelas dengan nilai -nilai atribut yang beda, sehingga memungkinkan program untuk mengolah banyak yang entitas berbeda dalam satu program.

## Kesimpulan

1. Kelas dan objek saling berhubungan dan merupakan bagian integral dari paradigma pemprograman berbasis objek.
2. Class digunakan untuk mendefinisikan struktur dari prilaku objek. Sementara objek digunakan untuk menciptakan instance konkret dari kelas tersebut.
3. Dengan menggunakan konsep class dan objek pengembang dapat membuat kode yang lebih terstruktur, modulor dan mudah dimengerti, serta meningkatkan fieksibilitasi dan reabilitas kode mereka.