LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN

“*Method dan Object*”



OLEH :

EZZA ADDINI

2311532001

DOSEN PENGAMPU : Dr. WAHYUDI, M. T.

ASISTEN:

HUMAYRA FAHRERI

DEPERTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS ANDALAS

PADANG 2023

1. PENDAHULUAN
2. Method

Istilah "method" dalam pemrograman Java merujuk pada blok kode yang dapat dieksekusi secara mandiri dan memiliki tugas atau fungsi tertentu. Method ini digunakan untuk mengorganisasi dan membagi logika program menjadi komponen yang lebih kecil, sehingga kode dapat lebih mudah dipahami dan dikembangkan. Dalam Java, terdapat beberapa modifier yang dapat digunakan untuk mengatur aksesibilitas, karakteristik, dan perilaku dari sebuah method, yakni public, private, protected, tingkat akses default atau tidak menggunakan kata kunci apa pun, abstract, serta static.

1. Public

Method dengan akses modifier public dapat diakses dari mana saja, baik dari class yang sama, package yang berbeda, maupun package yang sama.

1. Private

Method dengan akses modifier private hanya dapat diakses dari dalam class yang sama. Mereka tidak dapat diakses dari class lain, bahkan jika class tersebut berada dalam package yang sama. Penggunaan private seringkali digunakan untuk menyembunyikan detail implementasi internal dari class dan mencegah akses langsung dari luar class tersebut.

1. Protected

Method dengan modifier protected dapat diakses dari class yang sama, package yang sama, dan subclass (bahkan jika subclass berada di package yang berbeda).

1. Default

Jika tidak ada modifier yang ditentukan, method akan memiliki aksesibilitas pada level package (tingkat default). Method ini dapat diakses dari class dalam package yang sama, namun tidak dapat diakses dari package yang berbeda.

1. Abstract

Method dengan modifier abstract digunakan dalam konteks abstract class atau interface. Method ini tidak memiliki implementasi dan harus diimplementasikan oleh subclass (pada abstract class) atau implementasi class (pada interface).

1. Static

Method dengan modifier static adalah method yang terkait dengan class, bukan dengan instance objek. Mereka dapat dipanggil langsung dari class tanpa membuat objek dari class tersebut.

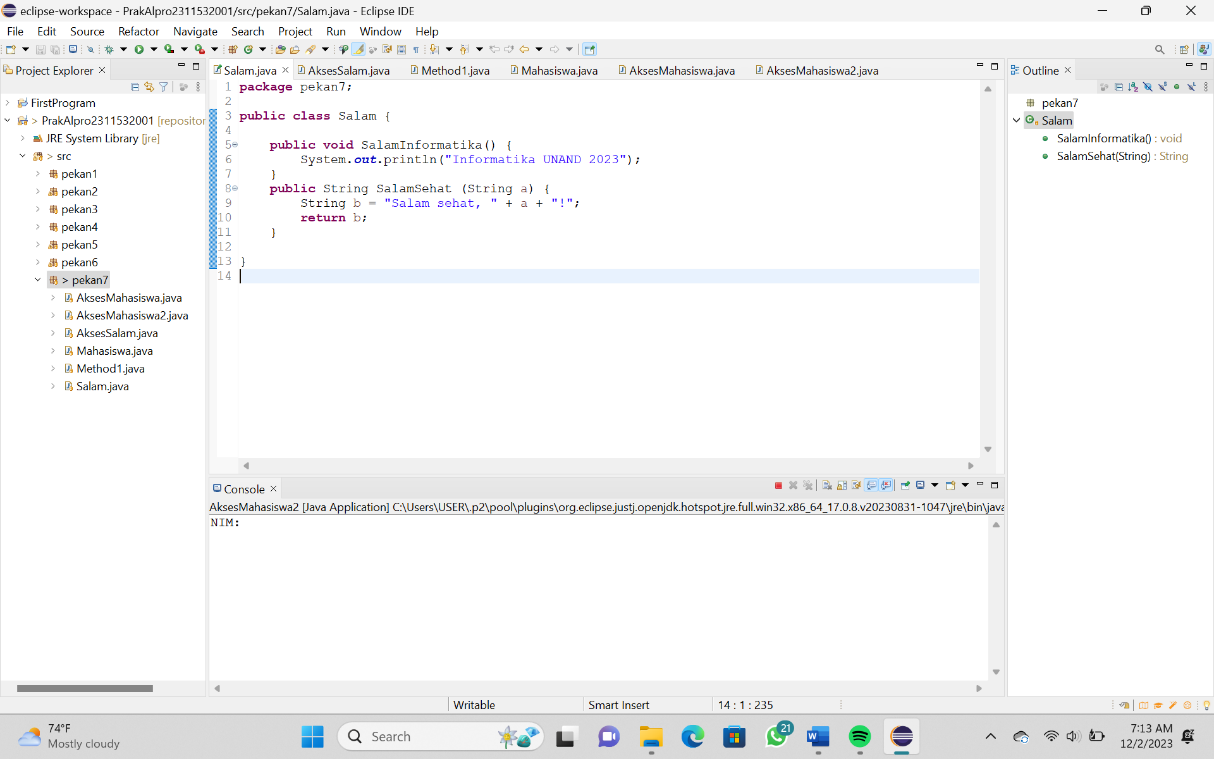
1. Object

Sebuah “object” adalah instance (instansi) dari suatu class dalam bahasa pemrograman Java, yang mana class sendiri adalah blueprint atau template yang mendefinisikan struktur dan perilaku objek. Sebuah object adalah realisasi konkret dari class, yang memiliki kemampuan untuk berinteraksi dengan objek lain atau melakukan tugas tertentu.

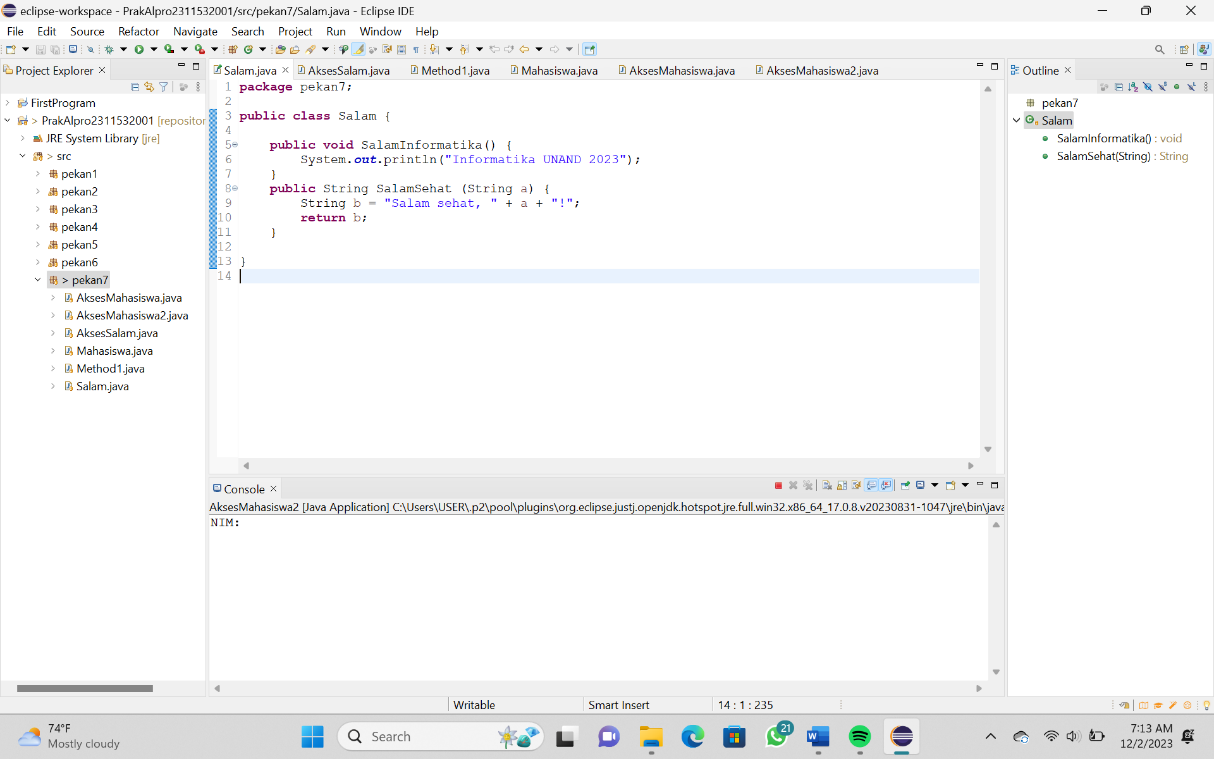
1. TUJUAN

Tujuan dari praktikum ini adalah:

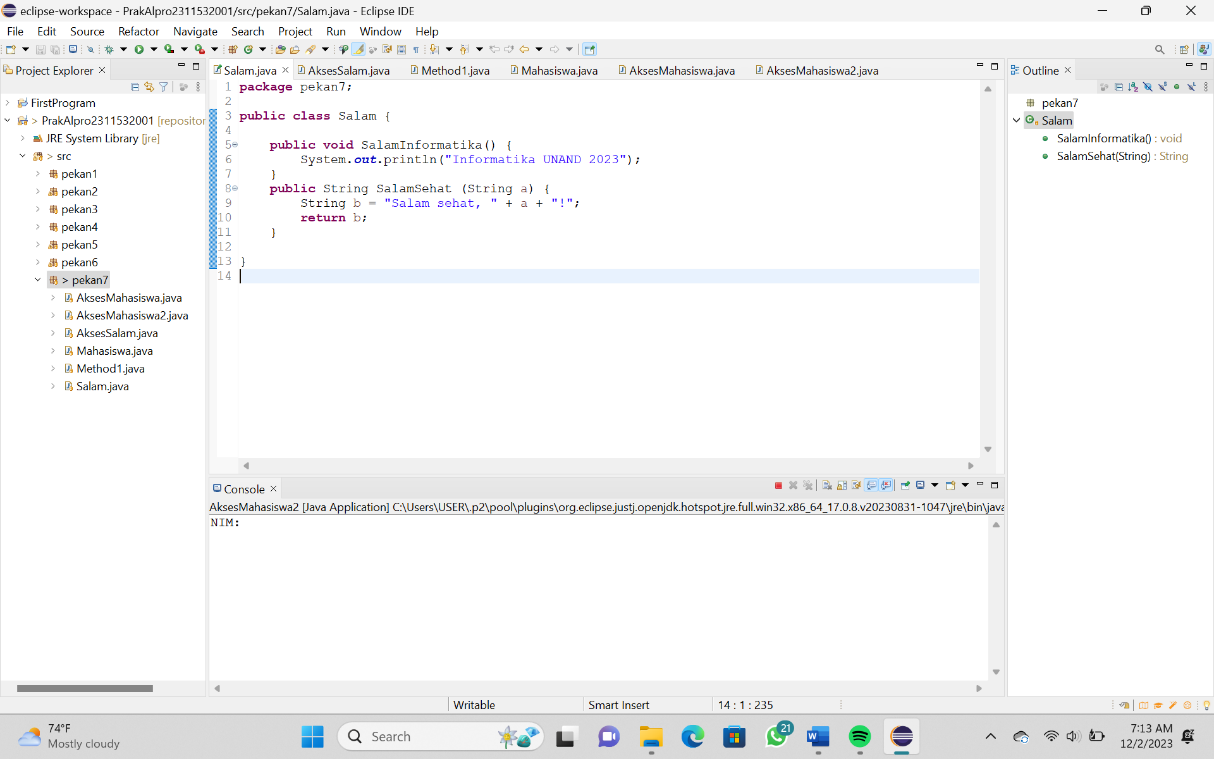
1. Memahami dan mengaplikasikan method dalam Bahasa Java.
2. Memahami dan mengaplikasikan object dalam Bahasa Java.
3. LANGKAH KERJA
4. Salam dan Akses Salam
5. Buat package dan class baru, namakan sesuai dengan yang diperintahkan atau yang diinginkan.



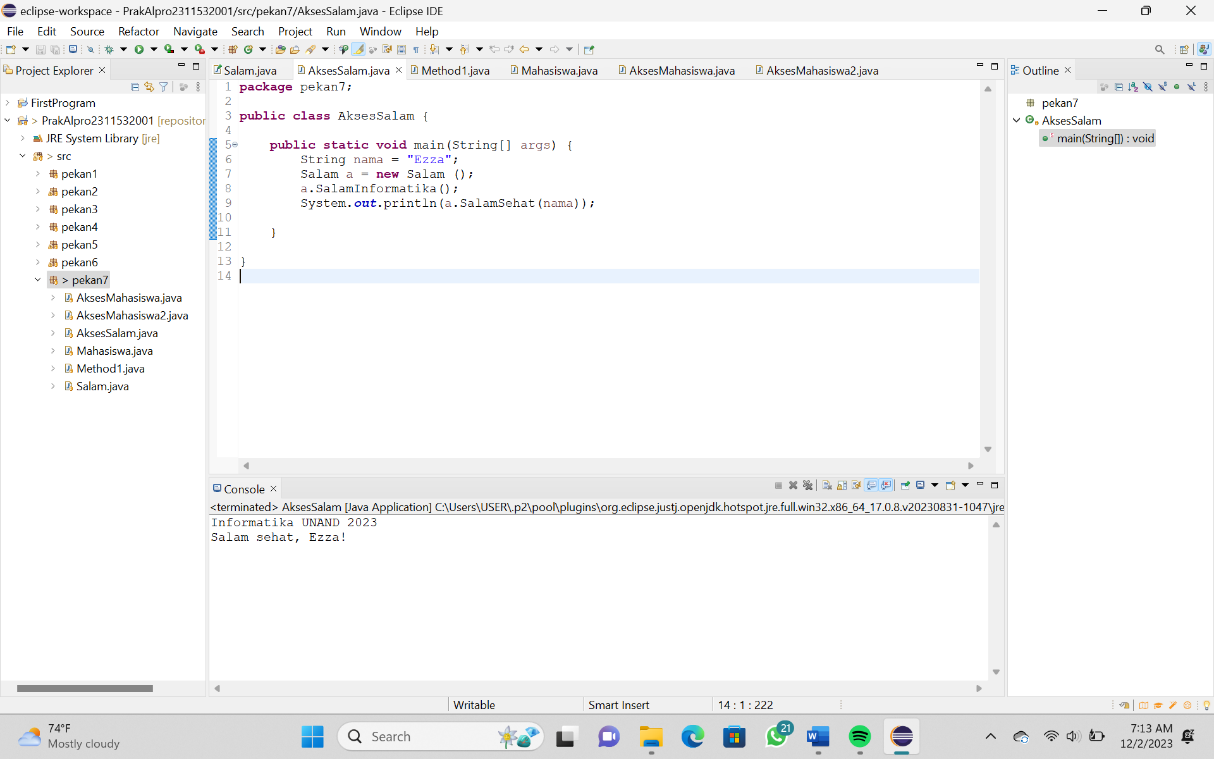
1. Buat method public seperti berikut ini, lalu buatkan perintah cetak output.



1. Buat method public return seperti berikut ini. Jangan lupa untuk mendeklarasikan variabel yang akan dipanggil, lalu berikan syntax return.



1. Buat class baru di dalam package yang sama untuk menjalankan program Salam.



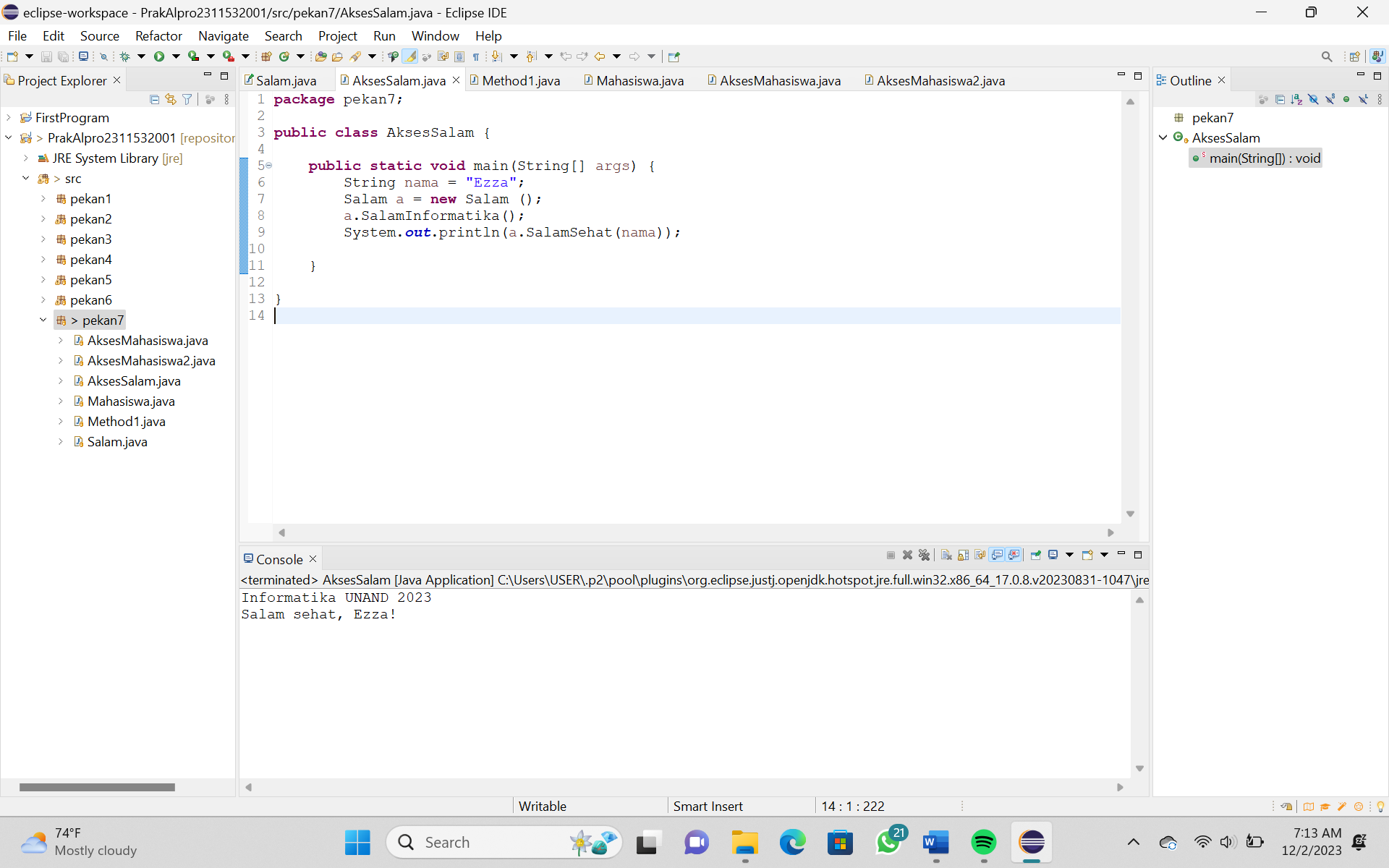
1. Gunakan method public static void main (String[] args).

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

1. Inisialisasikan nilai variabel yang akan digunakan seperti berikut.

String nama = "Ezza";

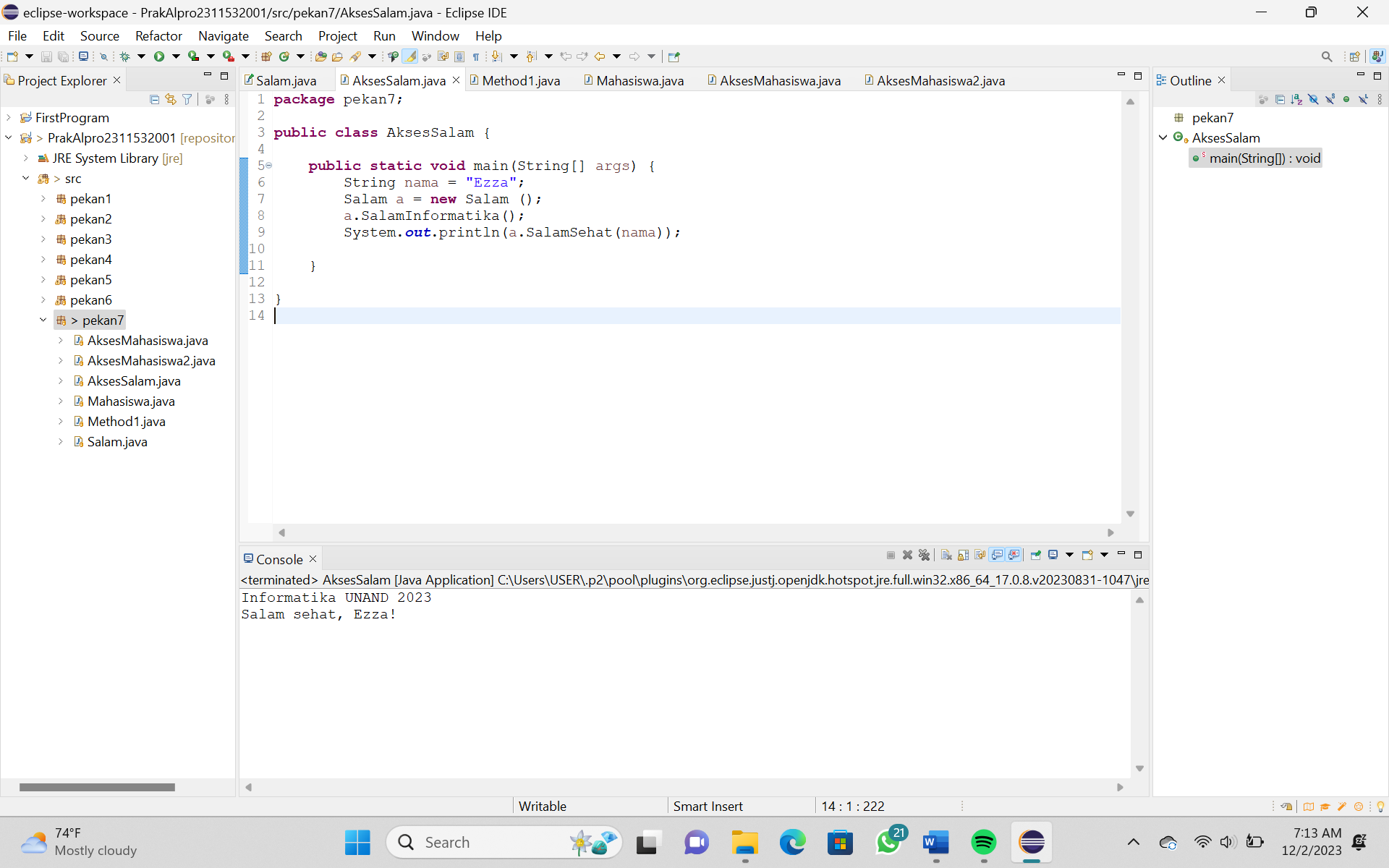
1. Tambahkan object Salam untuk memanggil blok kode Salam yang menggunakan method public di class sebelumnya.



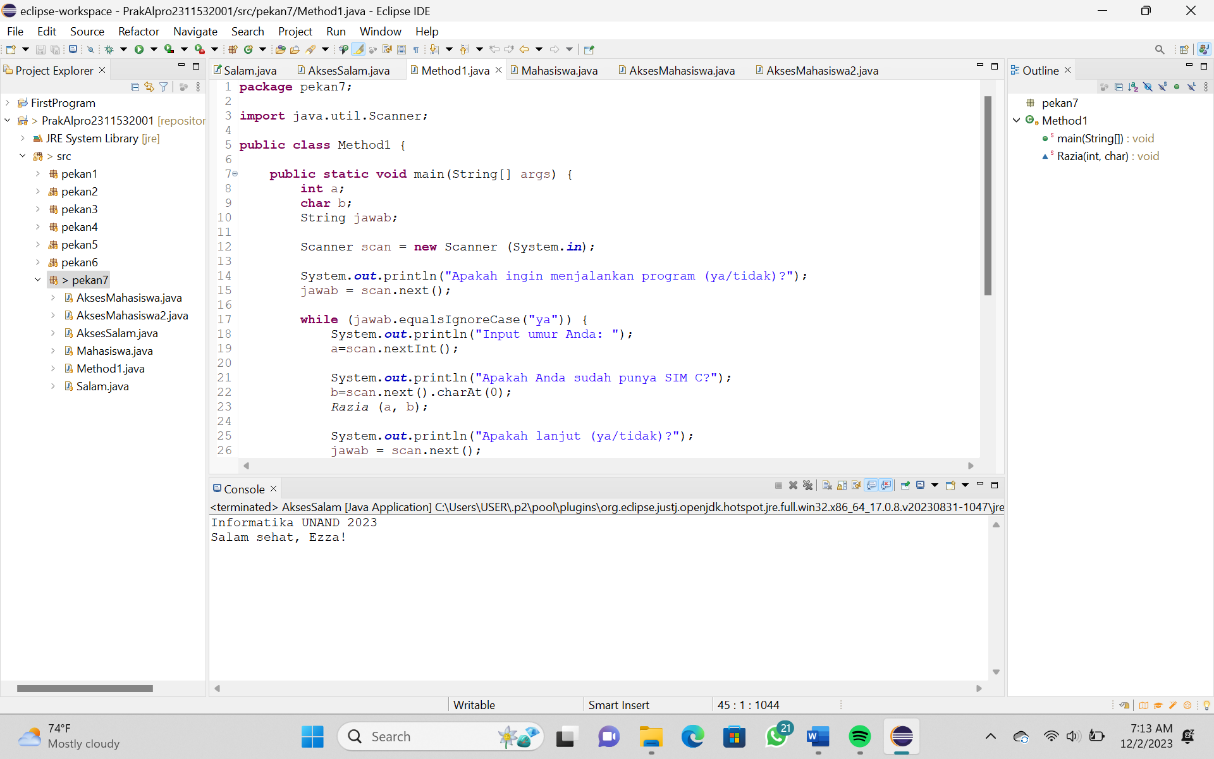
1. Buatkan perintah cetak output untuk blok kode Salam yang menggunakan method public return di class sebelumnya.

System.***out***.println(a.SalamSehat(nama));

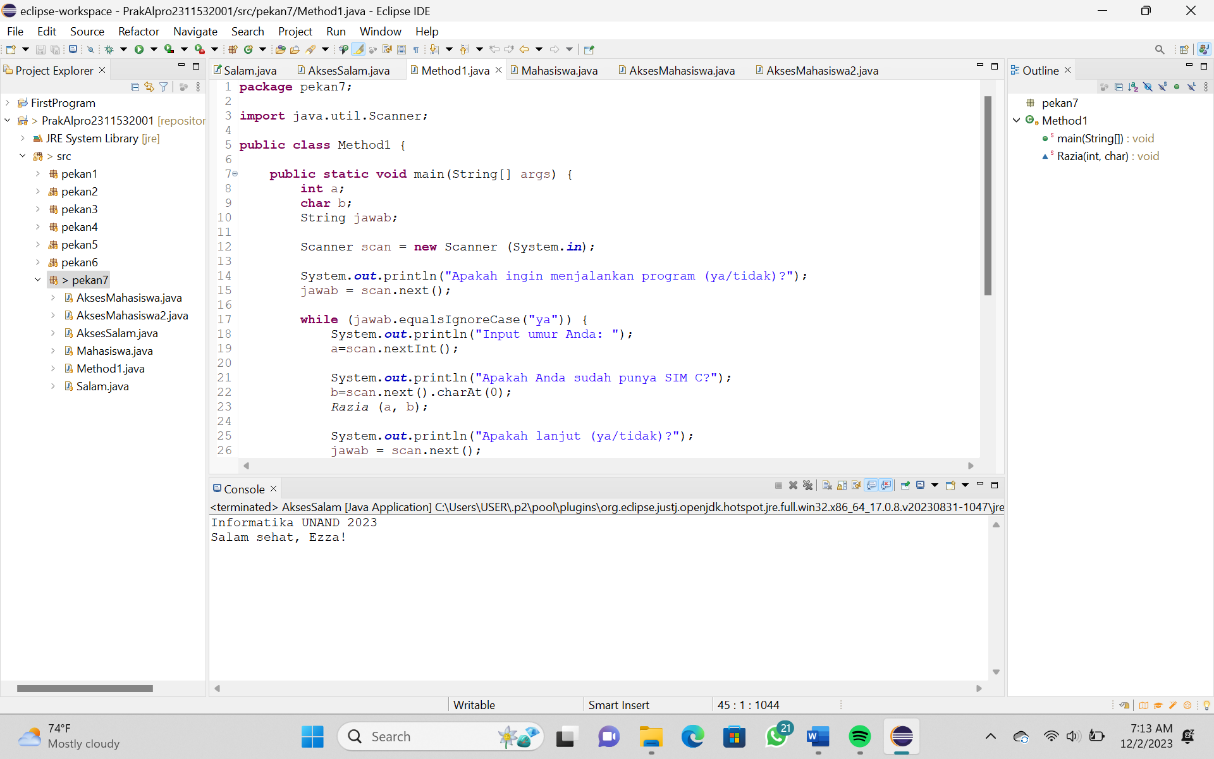
1. Jalankan program.



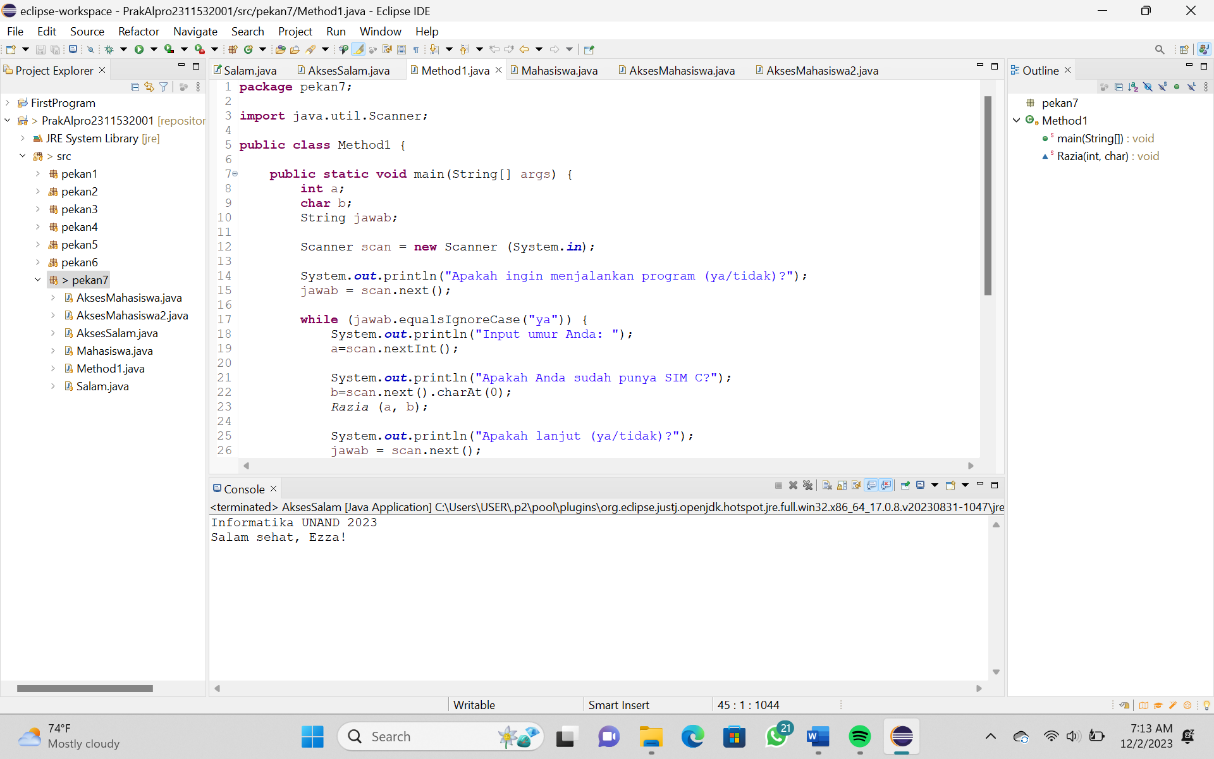
1. Method 1
2. Buat class baru dan namakan sesuai perintah atau yang diinginkan.



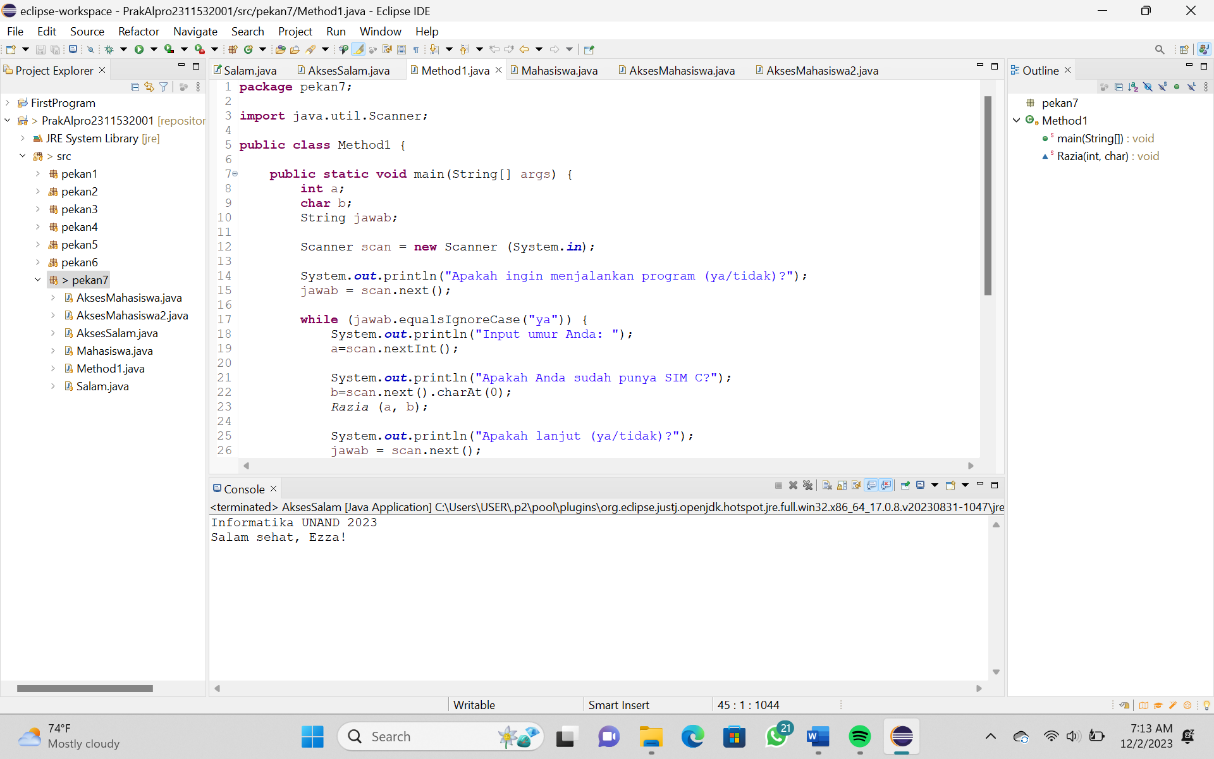
1. Buat object scanner seperti berikut.
2. Deklarasikan variabel.

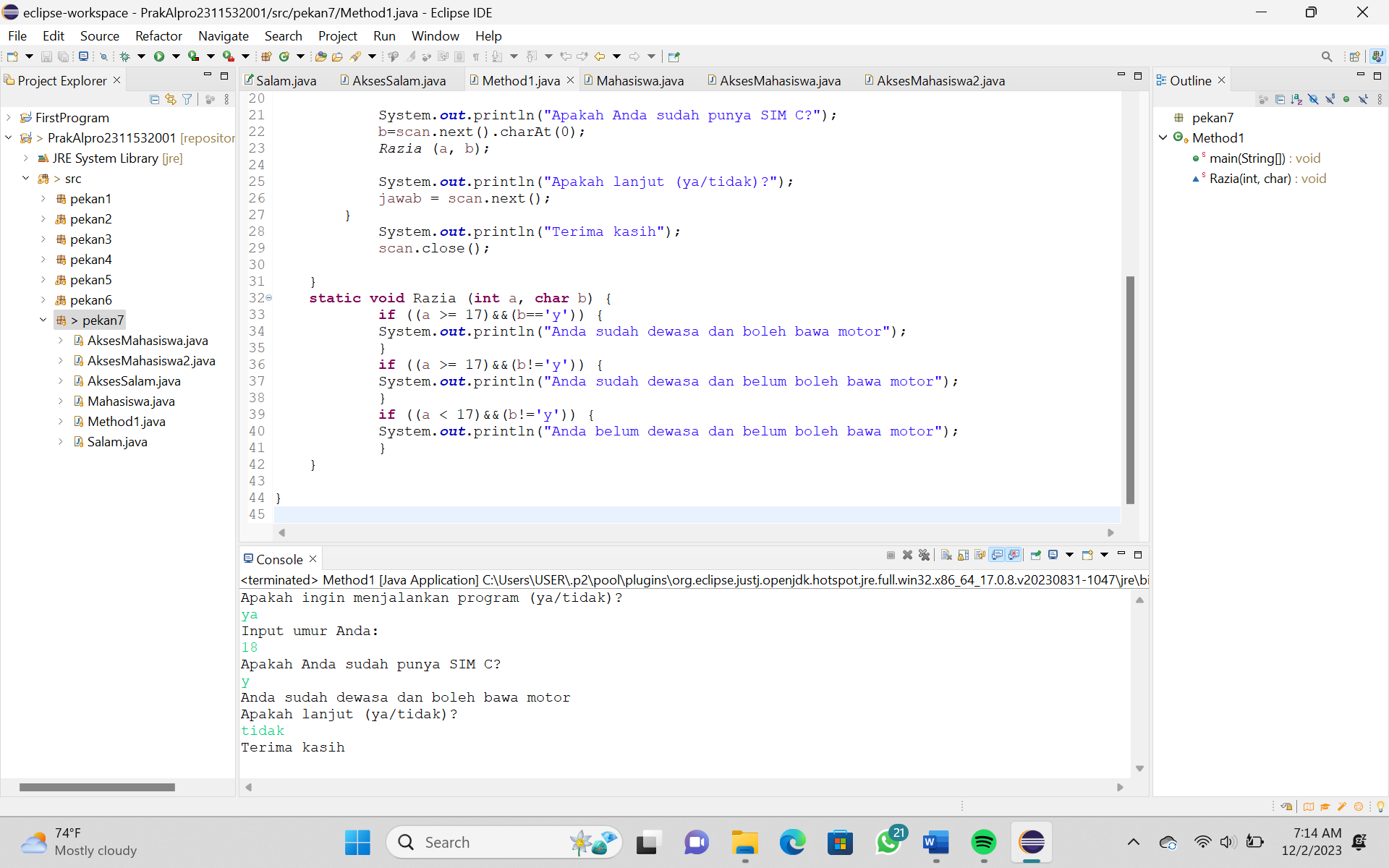


1. Buatkan perintah cetak output pertama.



1. Buatkan syntax while.

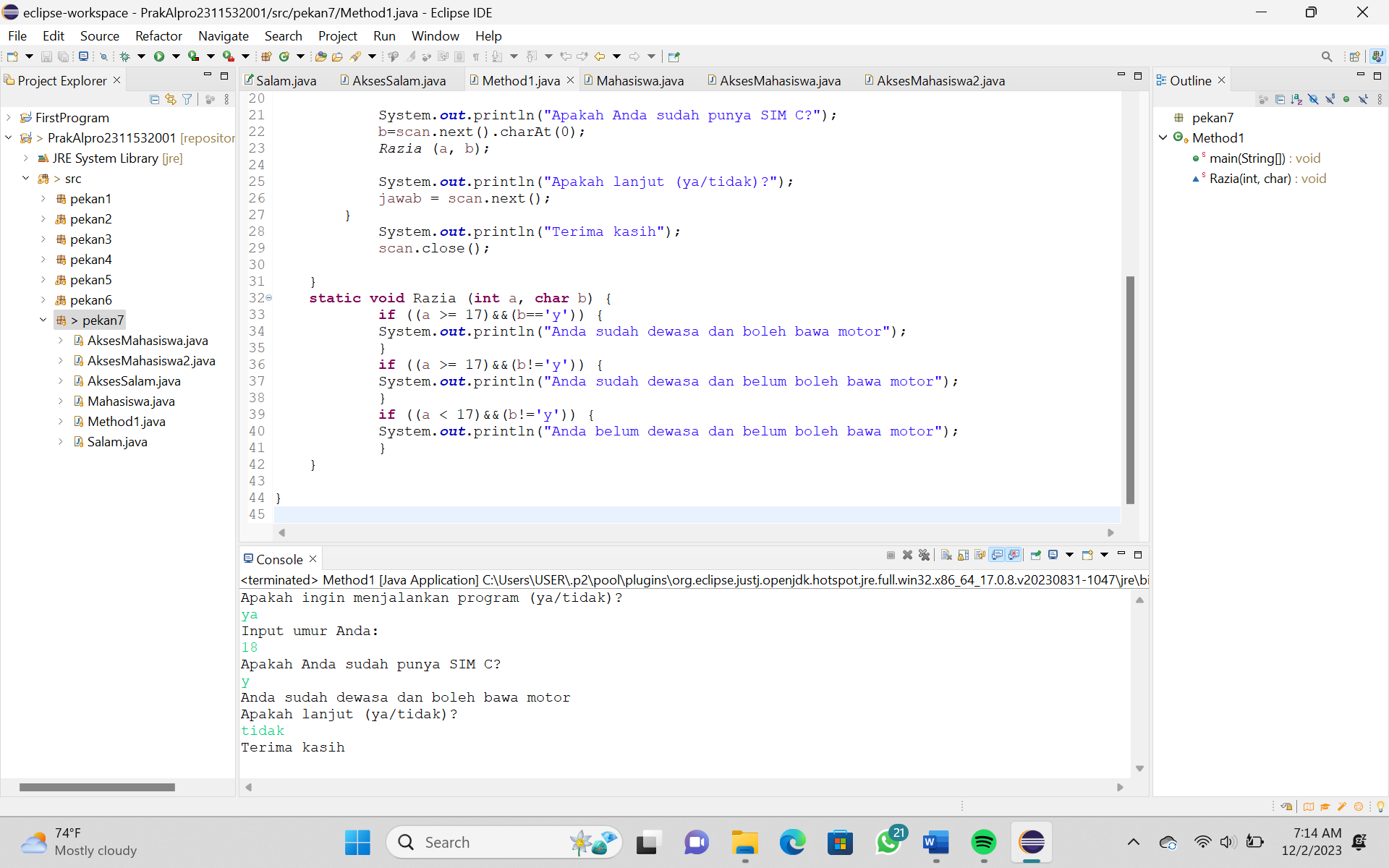




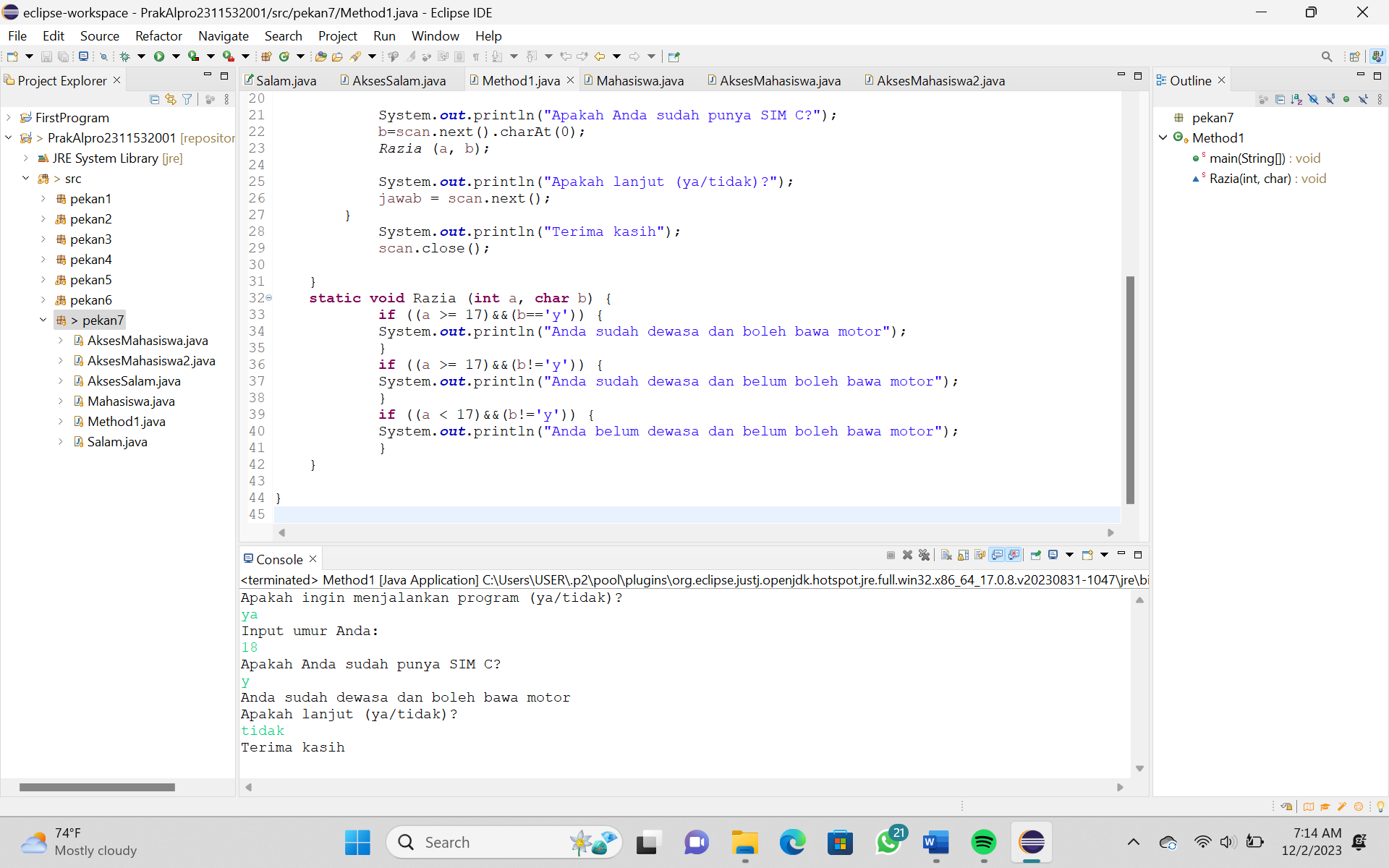
1. Selanjutnya, buatkan method static.

**static** **void** Razia (**int** a, **char** b) {

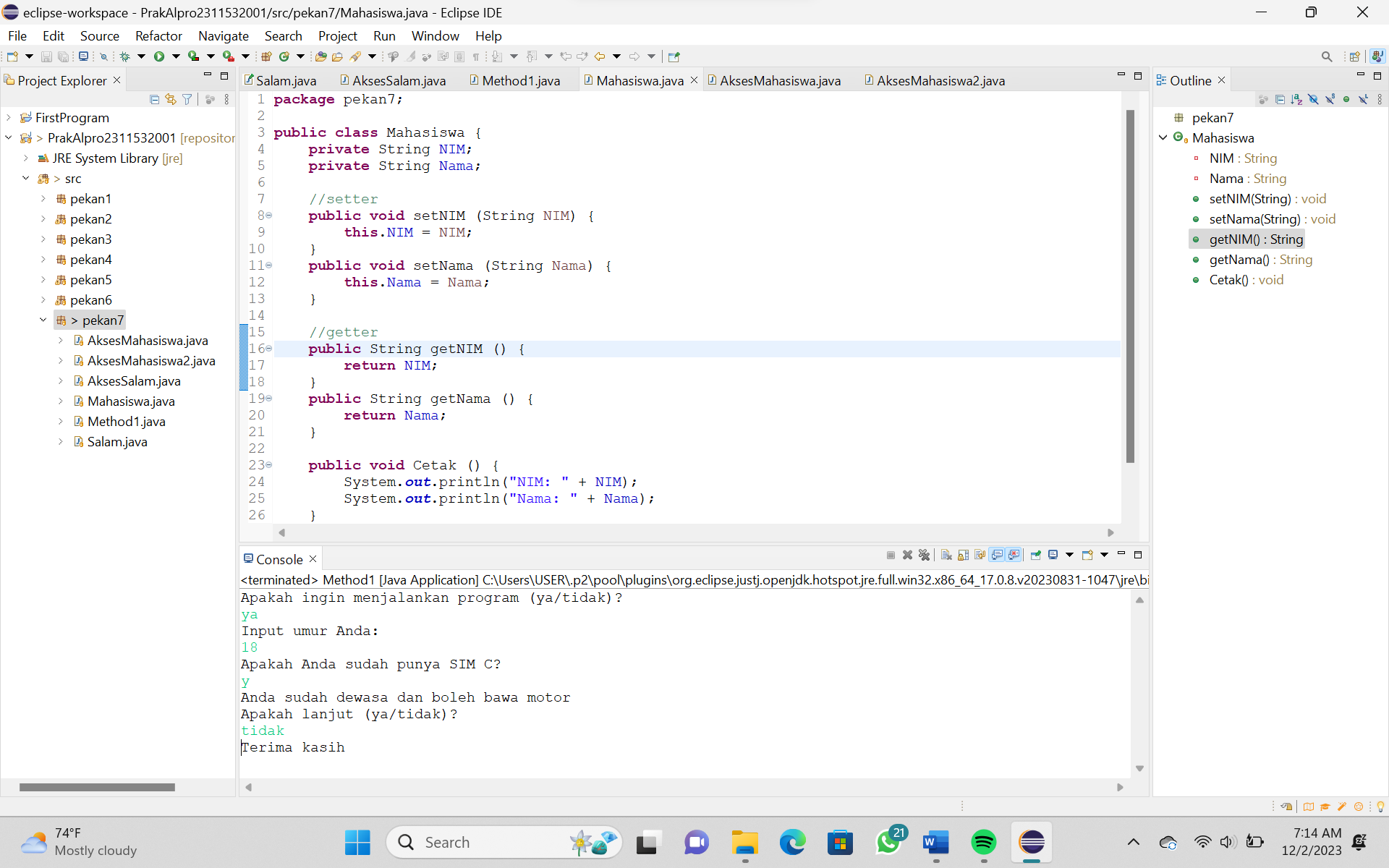
1. Beri batasan aturan menggunakan syntax if.



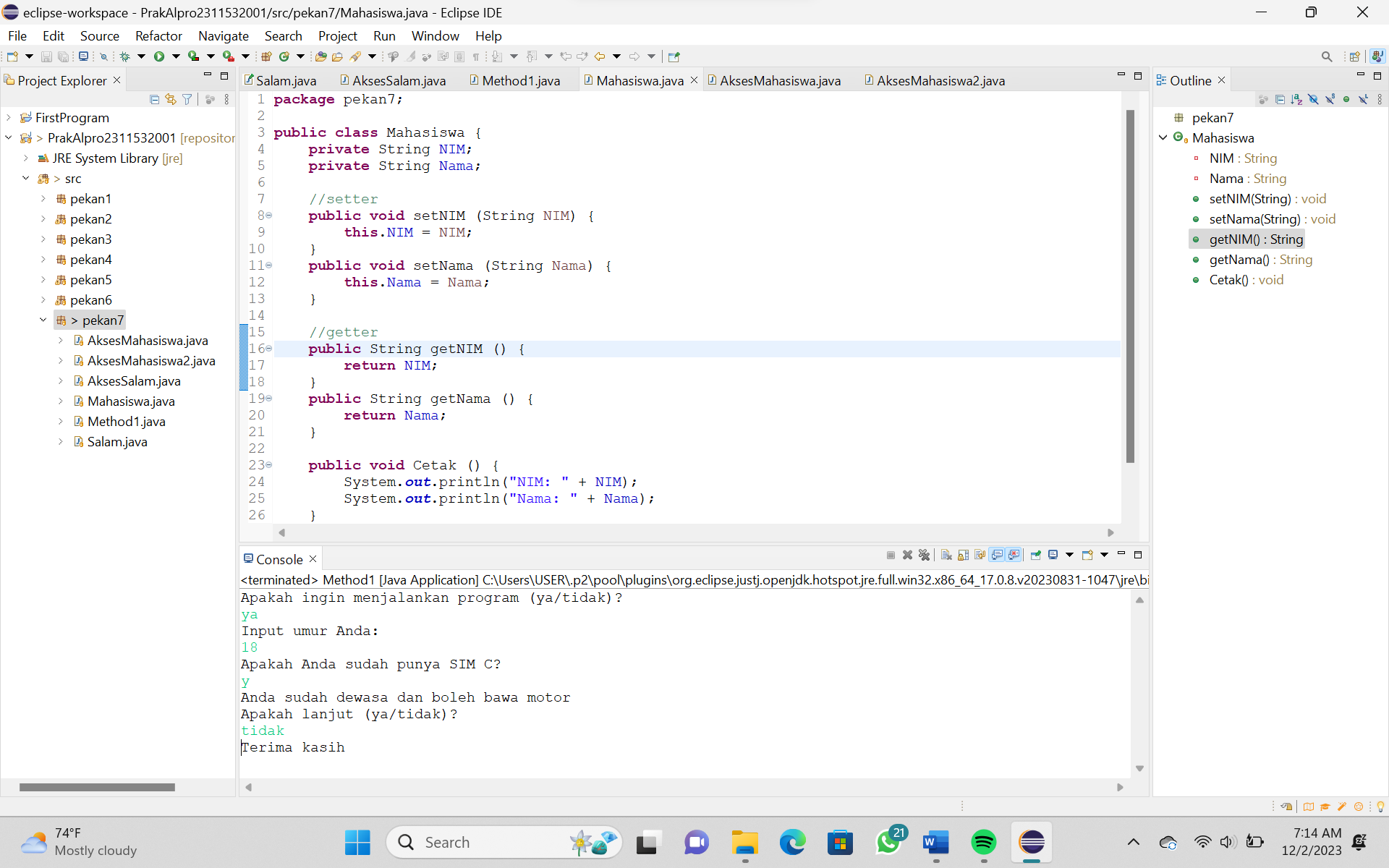
1. Jalankan program.



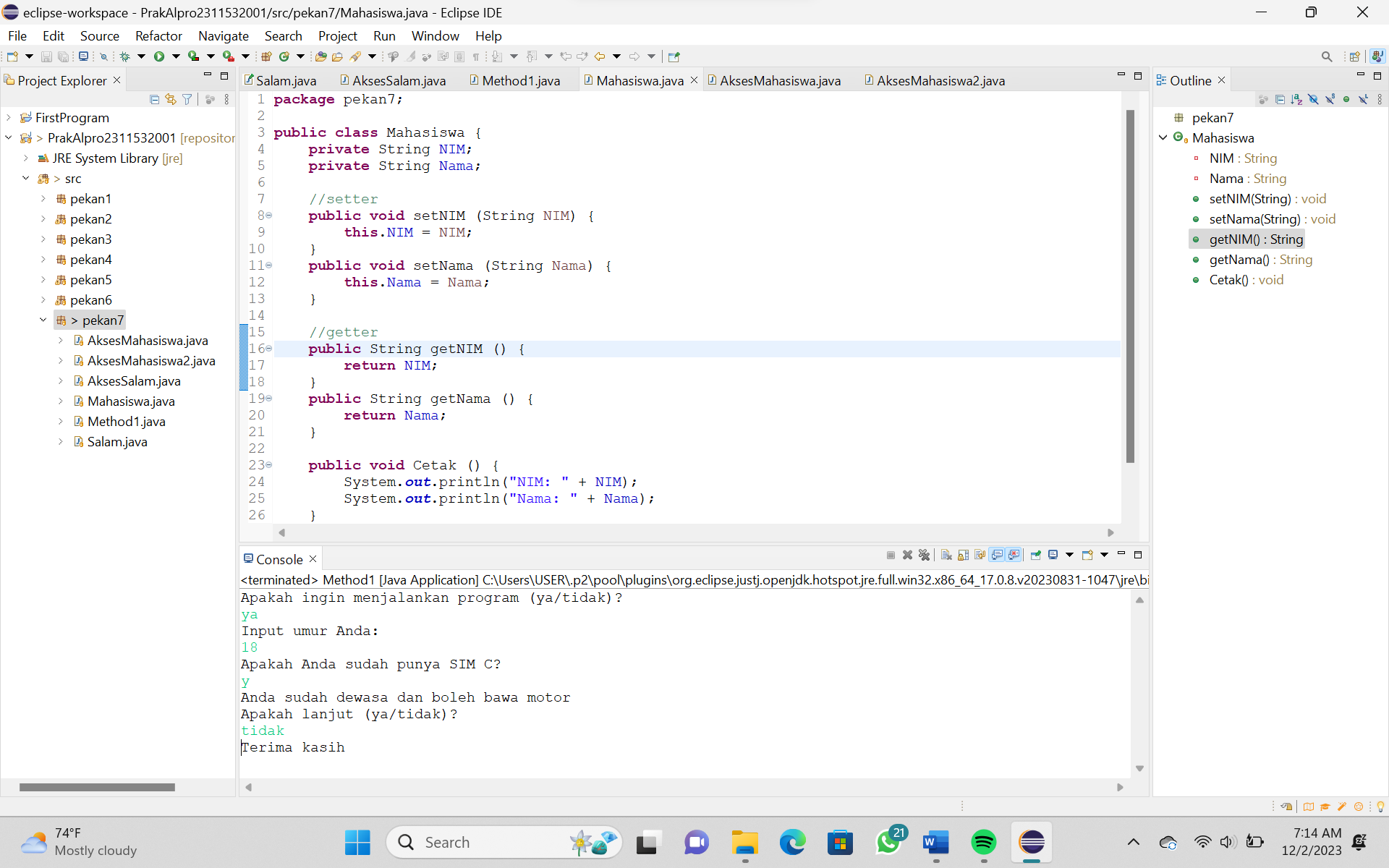
1. Mahasiswa dan Akses Mahasiswa (Menggunakan Setter dan Getter)
   1. Buat class baru dan namakan sesuai perintah atau yang diinginkan.



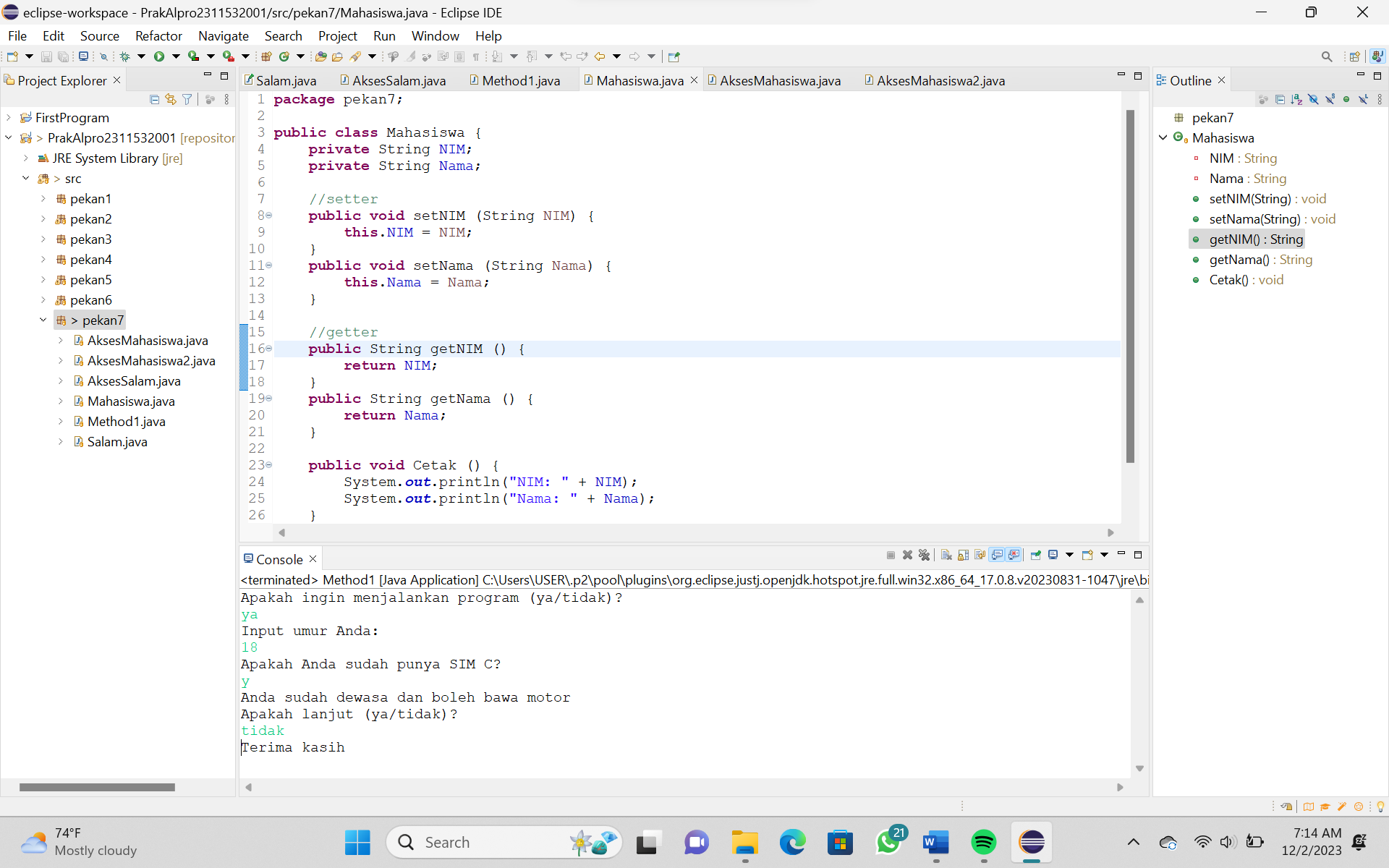
* 1. Buat method private untuk mendeklarasikan variabel.



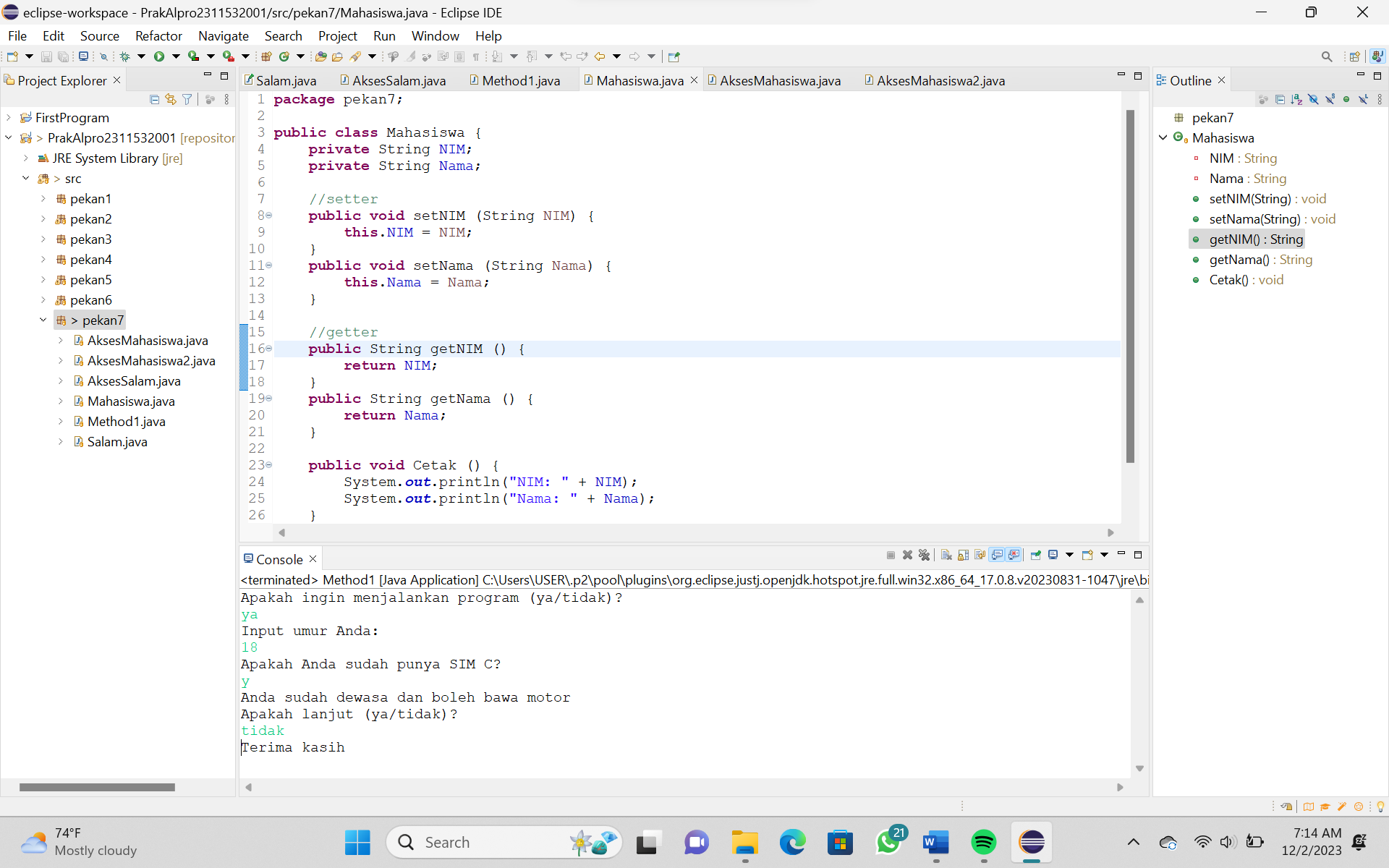
* 1. Selanjutnya, membuat setter. Deklarasikan method public void set variabel seperti berikut.



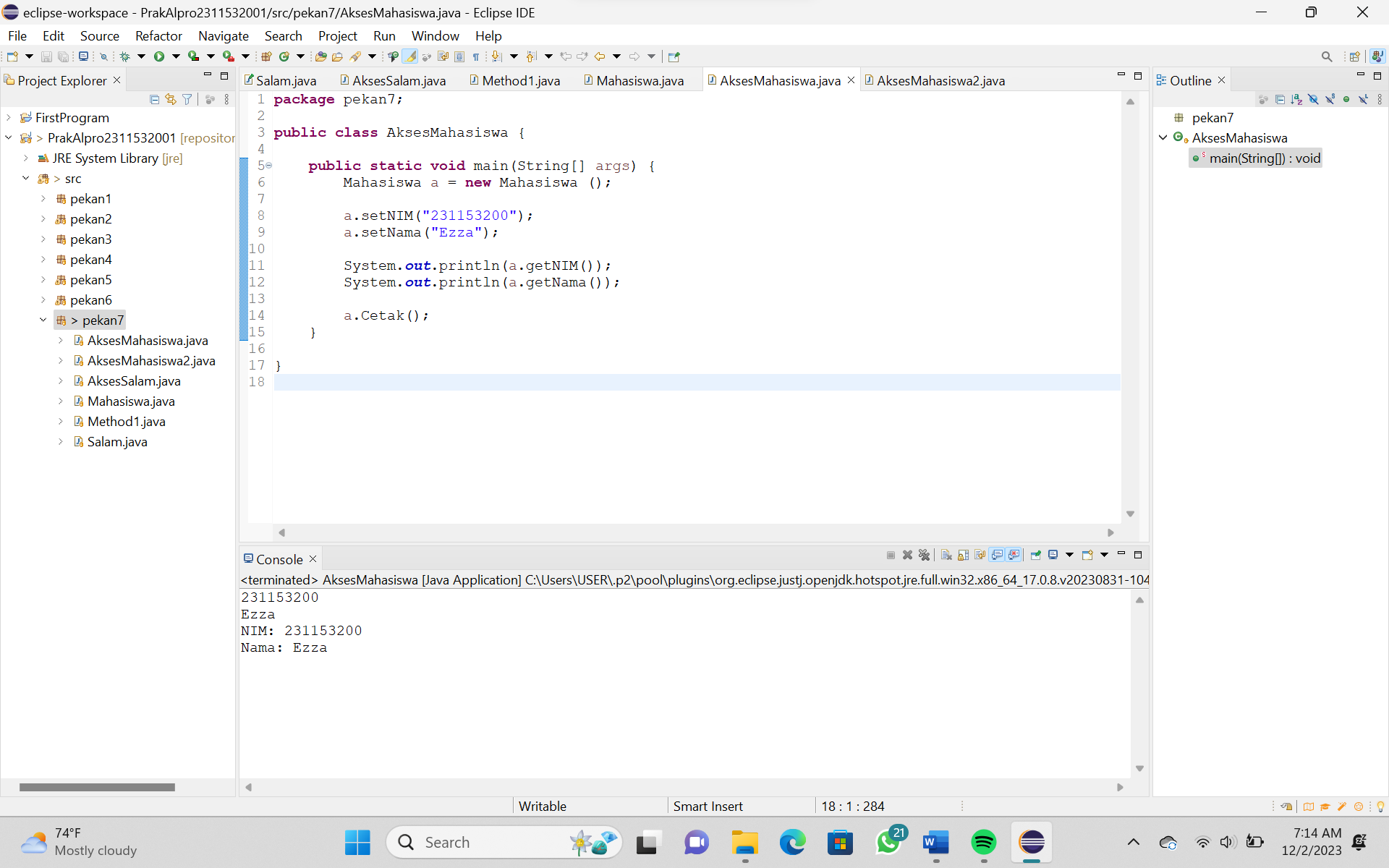
* 1. Kemudian, buatkan getter. Deklarasikan method public return variabel seperti berikut. Jangan lupa untuk mendeklarasikan variabelnya.



* 1. Buat method public void untuk mencetak output.



* 1. Selanjutnya, buat class baru di dalam package yang sama untuk mengakses program Mahasiswa tadi.



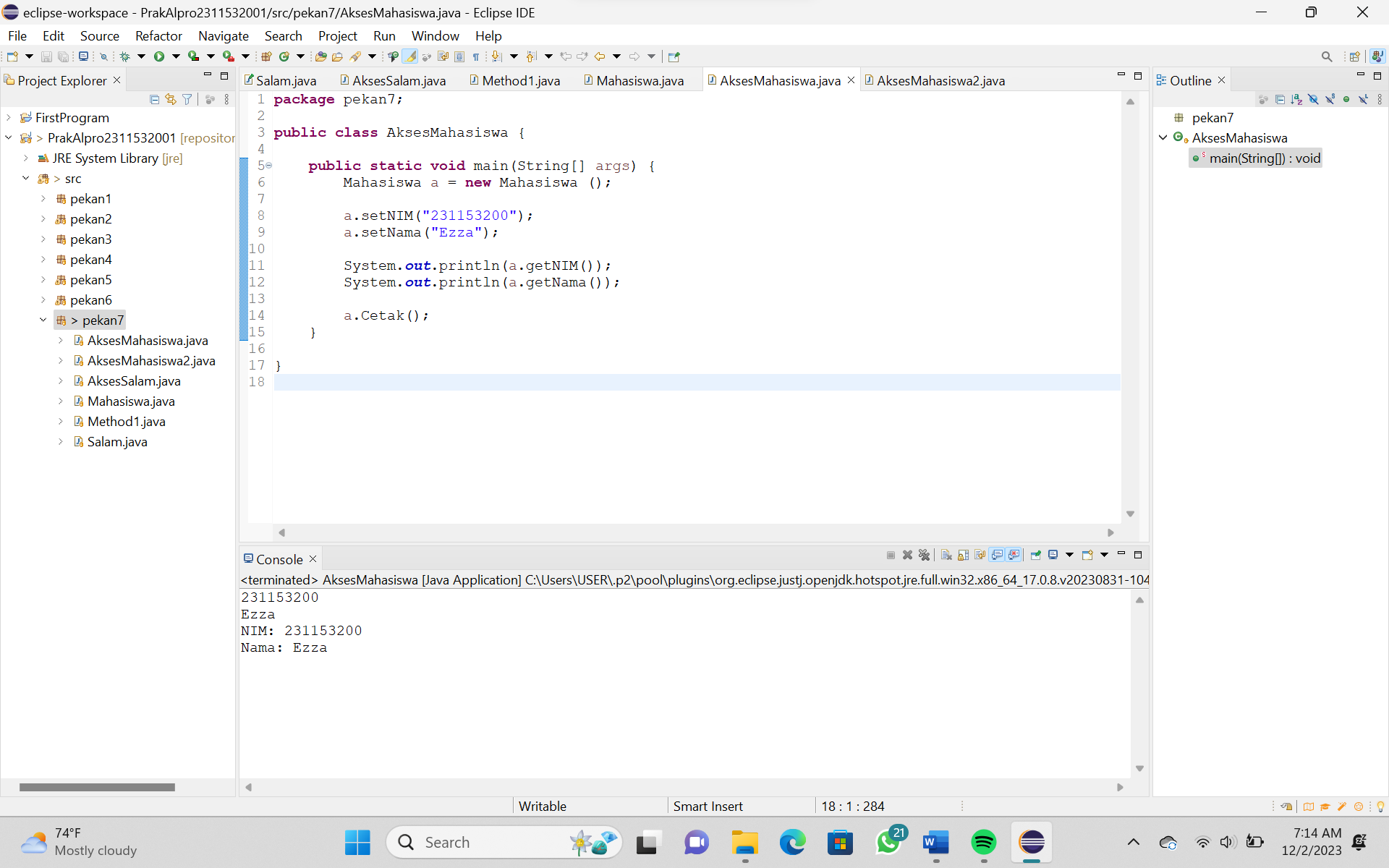
* 1. Gunakan method public static void main (String[] args).

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

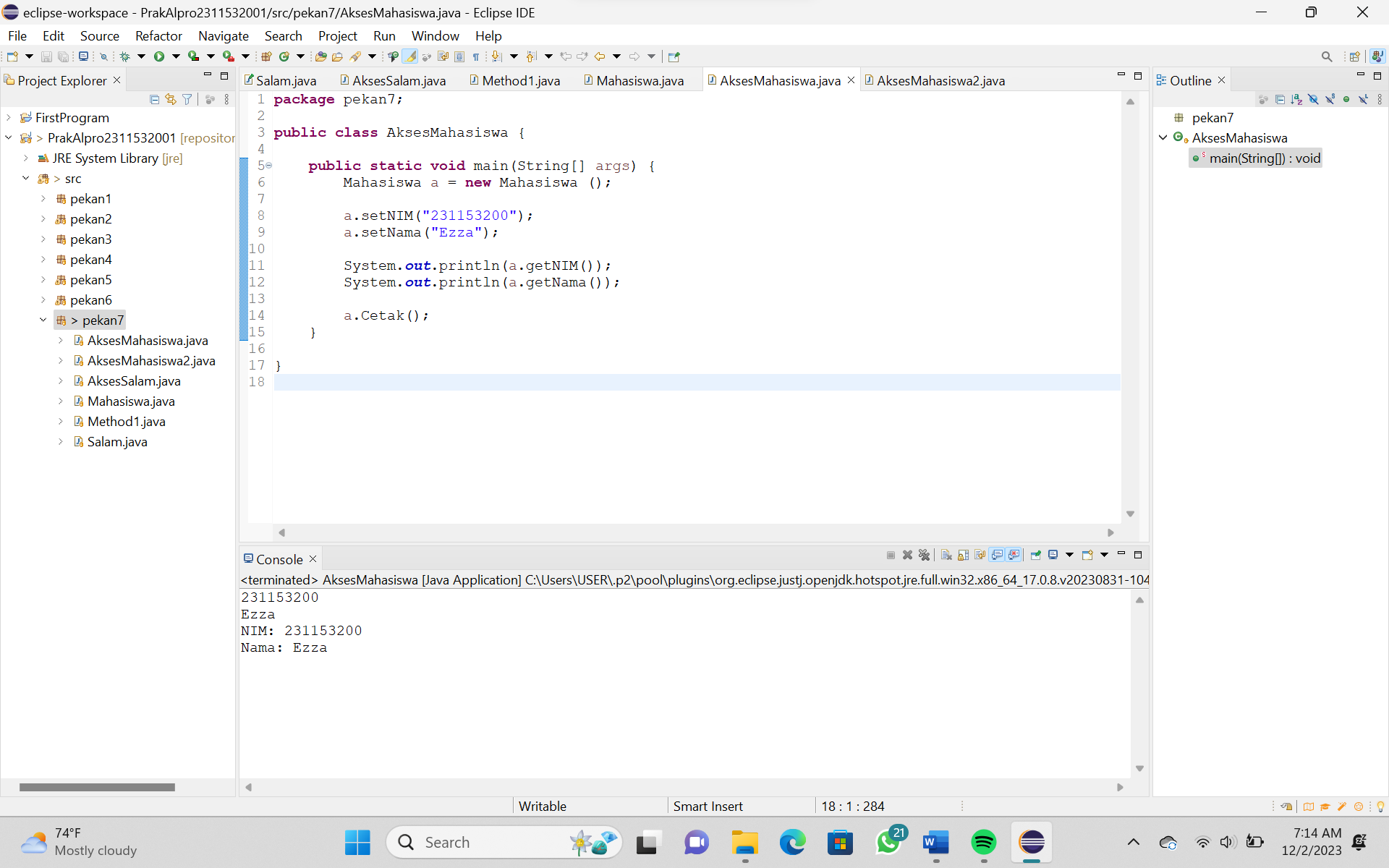
* 1. Tambahkan object Mahasiswa untuk memanggil blok kode Mahasiswa di class sebelumnya.

Mahasiswa a = **new** Mahasiswa ();

* 1. Deklarasikan variabel setter.



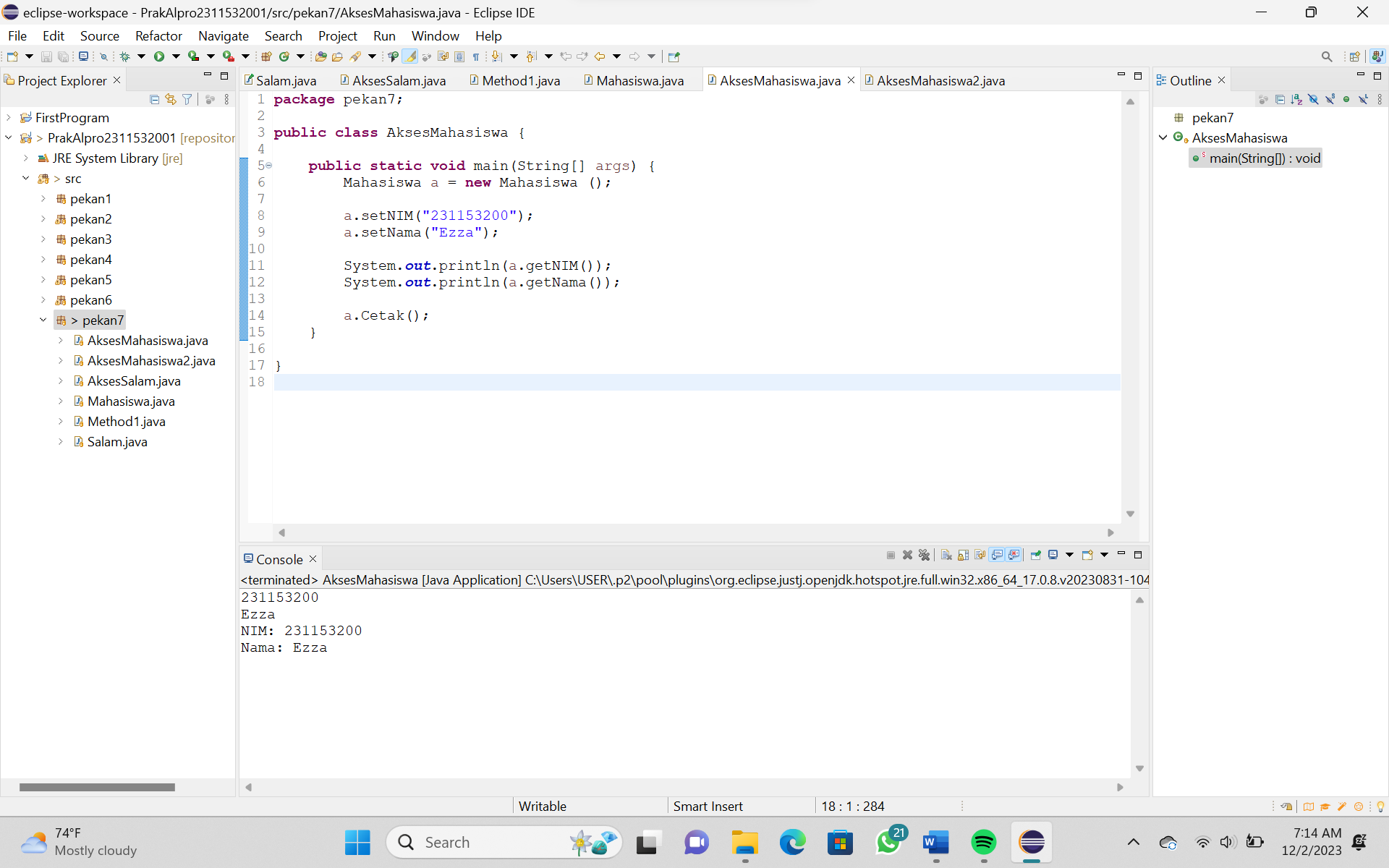
* 1. Buatkan perintah cetak output.



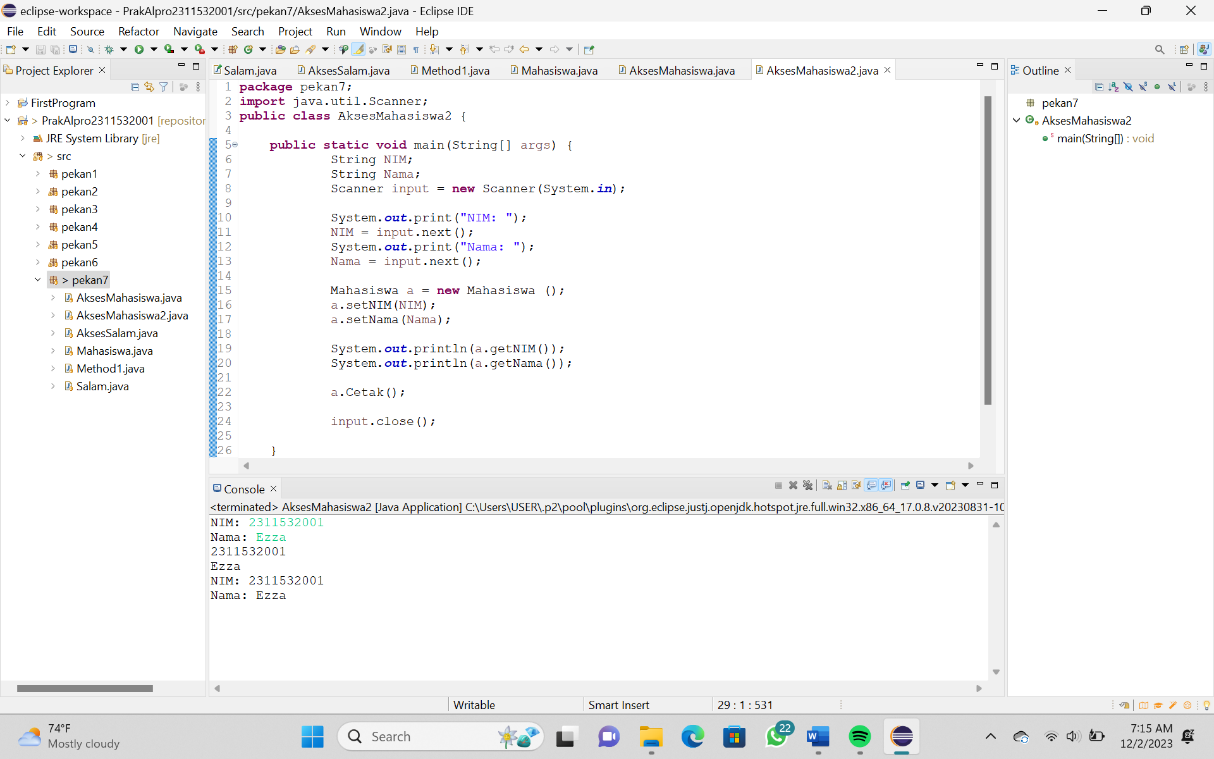
* 1. Panggil perintah cetak.

a.Cetak();

* 1. Jalankan program.

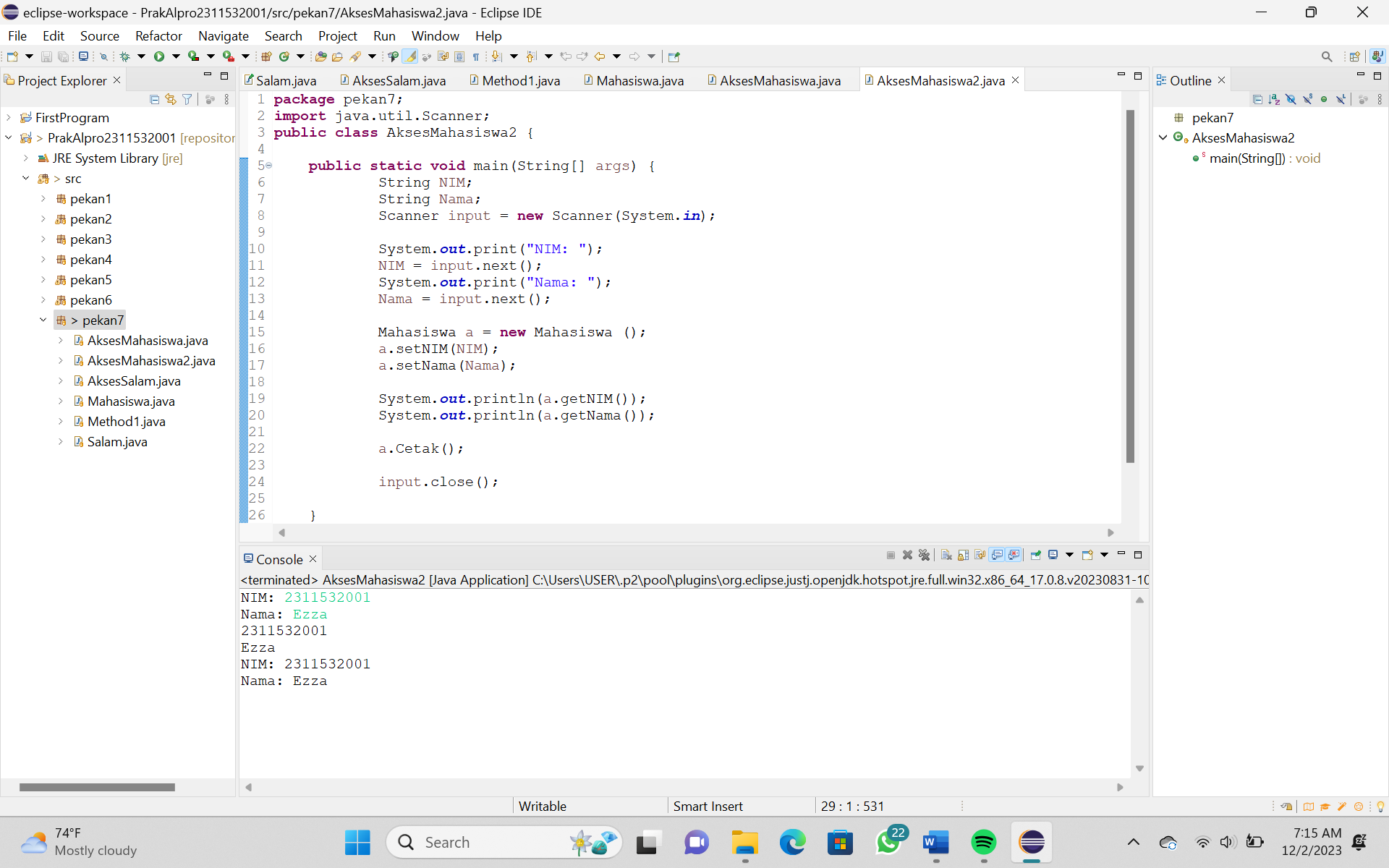


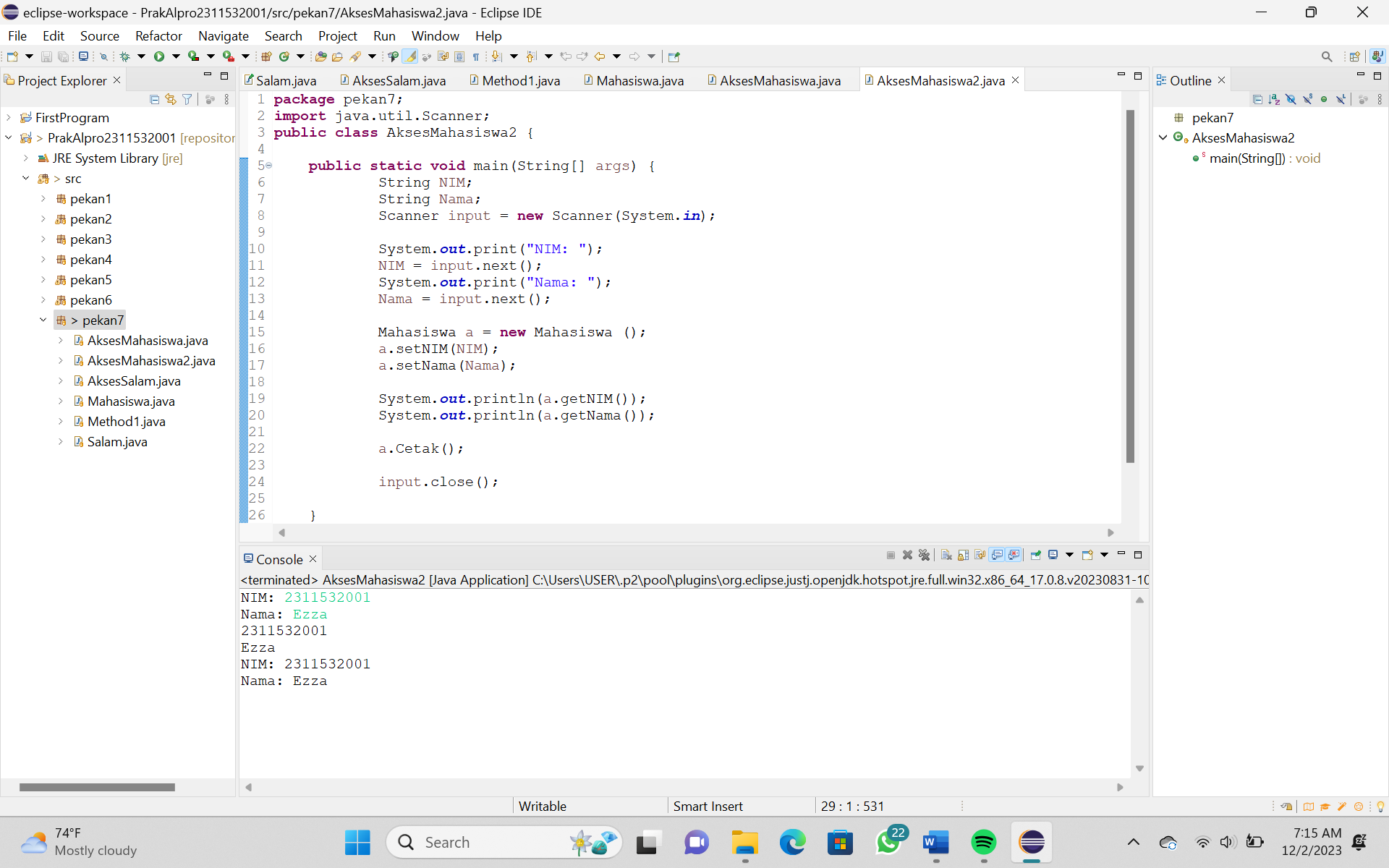
1. Akses Mahasiswa (Menggunakan Object Scanner)
2. Buat class baru dan namakan sesuai perintah atau yang diinginkan.



1. Buat object scanner seperti berikut ini.

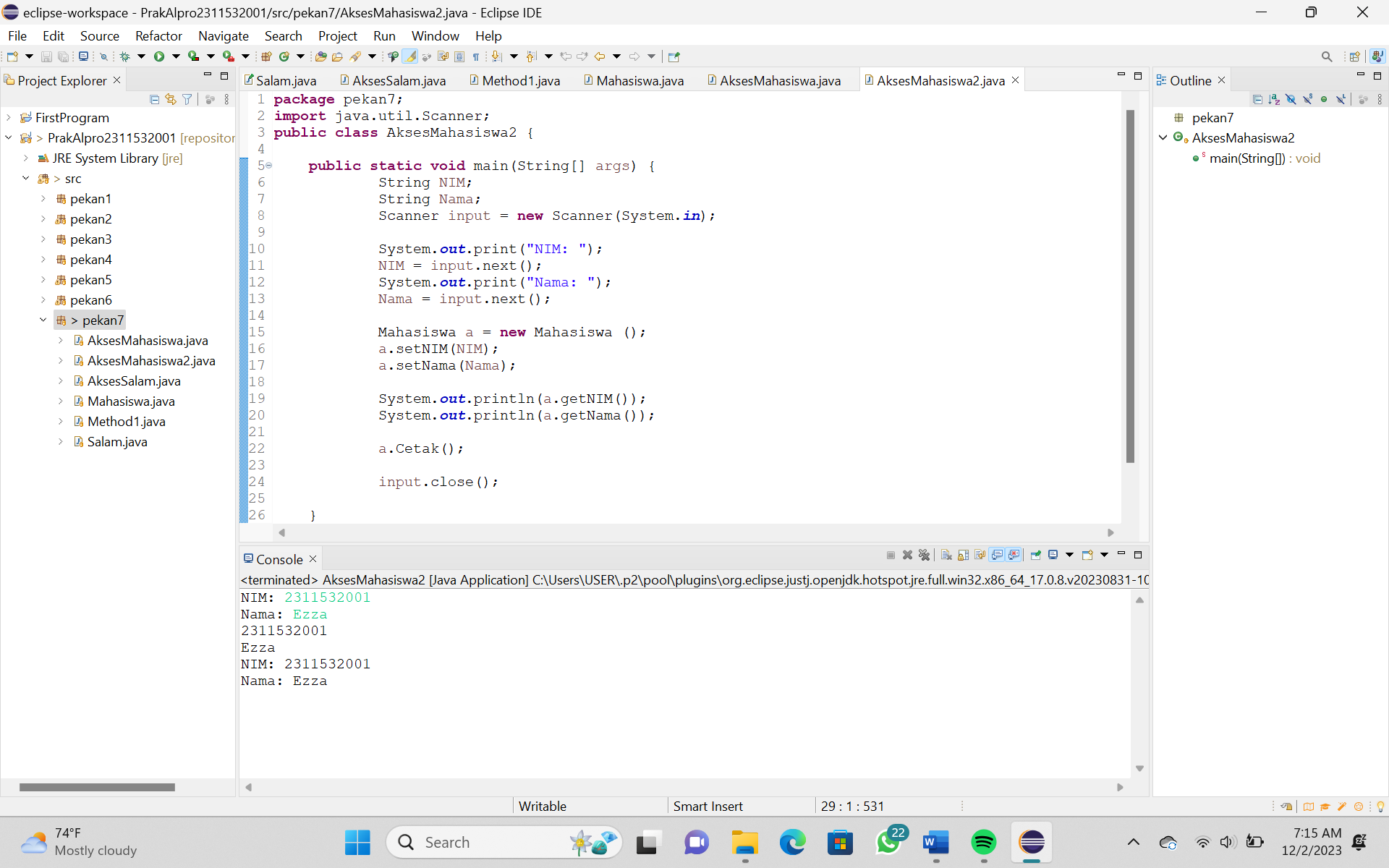
Scanner input = **new** Scanner(System.***in***);

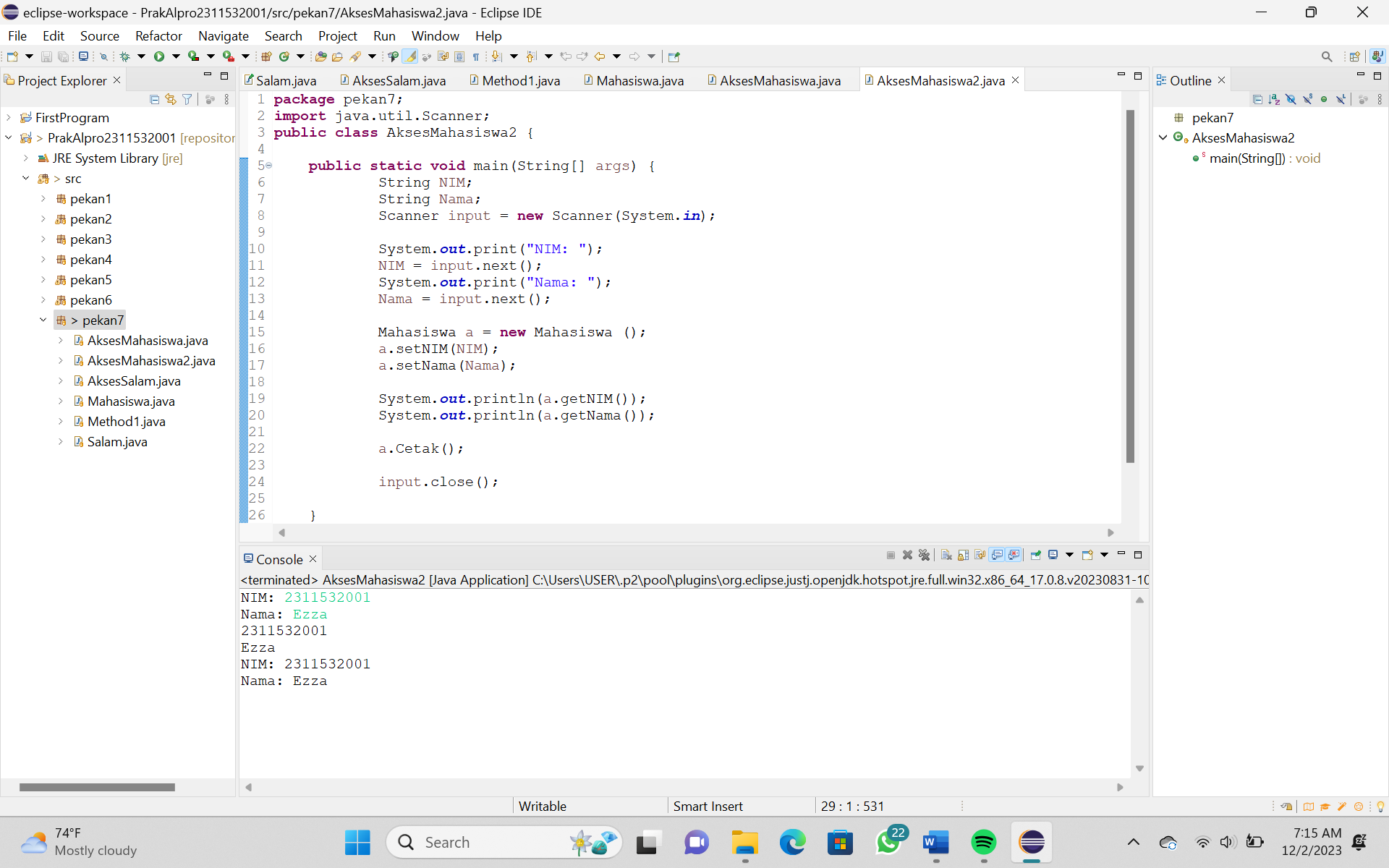
1. Deklarasikan variabel.  
   
2. Buatkan perintah output seperti berikut.



1. Tambahkan object Mahasiswa untuk memanggil blok kode Mahasiswa di class sebelumnya.

Mahasiswa a = **new** Mahasiswa ();

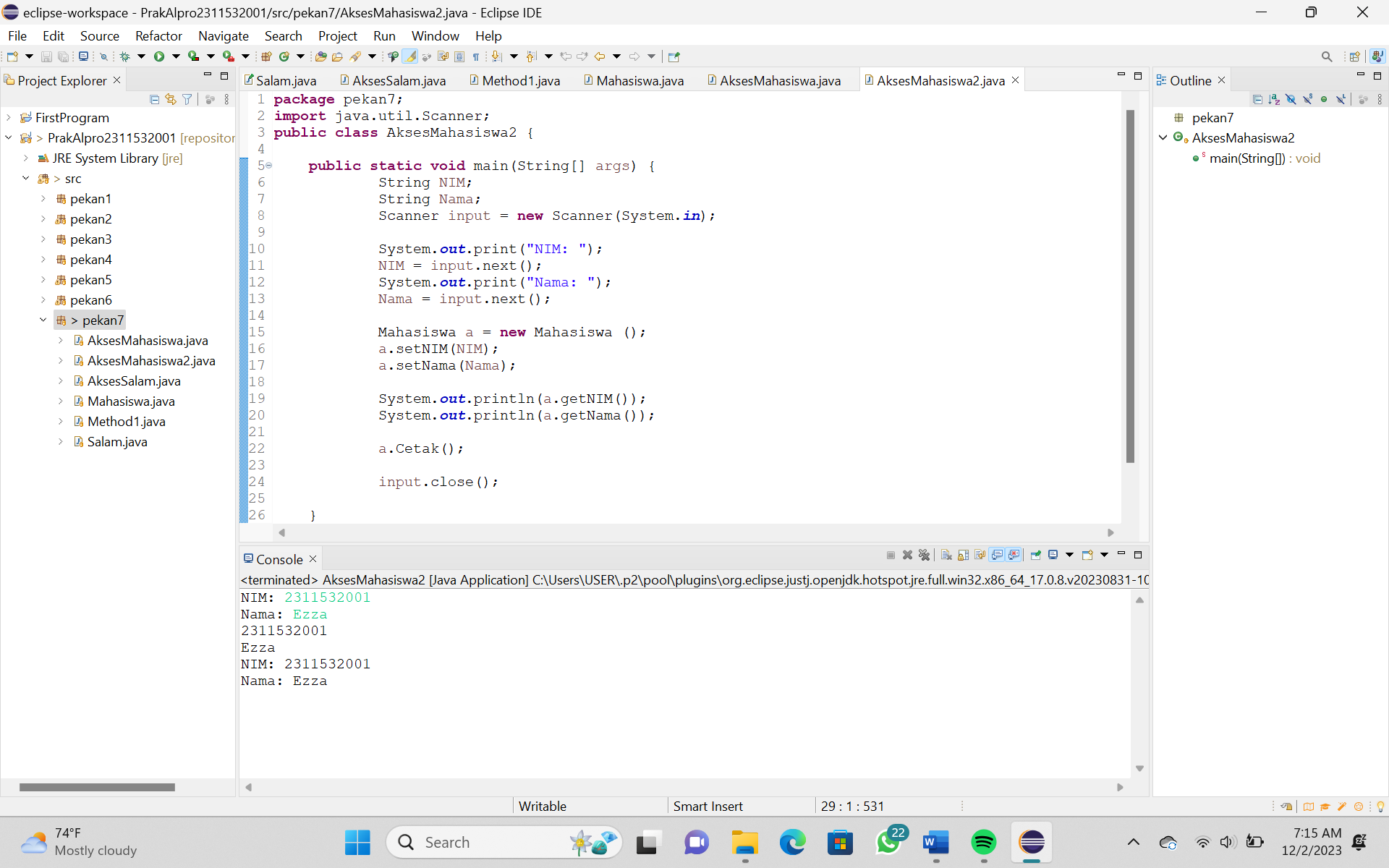
1. Deklarasikan variabel setter.  
   
2. Buatkan perintah cetak output.



1. Panggil perintah Cetak.

a.Cetak();

1. Jalankan program.



1. KESIMPULAN

Dari praktikum yang telah dilakukan, maka dapat diambil kesimpulan tentang hubungan object dan method dalam Bahasa Java. Object menggunakan atribut dan metode yang didefinisikan dalam class. Method seringkali digunakan untuk melakukan operasi atau manipulasi terhadap atribut objek. Sehingga, konsep pemrograman berorientasi objek (OOP) mendorong pembuatan class dan objek untuk mengorganisir kode.