PENGANTAR

Fundamen Pengembangan Aplikasi

PENGENALAN KULIAH

- Bobot 6 SKS
- 28 Pertemuan (14 Pekan)
- Tiap pertemuan dilaksanakan secara asinkron & sinkron
- Dilaksanakan secara luring/daring

- Bahan Kajian Utama
  - Bahasa Pemrograman Java
  - Pemrograman Berorientasi Objek
  - XML (Extensible Markup Language)
  - GUI (Graphical User Interface)

CAPAIAN PEMBELAJARAN

- **KK 2 – Keterampilan Analitis**

  Mampu menganalisis masalah dan mendefinisikan kebutuhan teknologi informasi untuk menyelesaikannya

- **KK 4 – Keterampilan Desain**
- **Mampu mendesain solusi teknologi informasi untuk memenuhi kebutuhan pengguna**

- **KK 6 – Kemampuan Evaluasi**

  Mampu mengevaluasi dampak lokal dan global teknologi informasi pada individu, organisasi, dan masyarakat

CAPAIAN PEMBELAJARAN MATA KULIAH (CPMK)

CPL	CPMK	Rumusan CPMK	Indikator	Penilaian	Bobot
KK2	**KK2.SIF202.1**	Mahasiswa dapat **menguraikan** suatu permasalahan pemrograman dasar dan pemrograman berorientasi objek yang perlu dijabarkan **objek-objek** beserta **interaksinya** yang terkait proses penyelesaian masalah tersebut.	Mahasiswa dapat menguraikan suatu permasalahan pemrograman dasar dan pemrograman berorientasi objek yang perlu dijabarkan objek-objek beserta interaksinya yang terkait proses penyelesaian masalah tersebut dengan benar.	• Tes Tertulis = 75% • Penugasan = 25%	25%
KK4	**KK4.SIF202.1**	Mahasiswa dapat **merancang solusi** terkait sebuah permasalahan pemrograman dasar dan pemrograman dengan pendekatan **berorientasi objek** dan framework **MVC** (Model, View, Controller) menggunakan alat bantu	Mahasiswa dapat merancang solusi terkait sebuah permasalahan pemrograman dasar dan pemrograman dengan pendekatan berorientasi objek dan framework MVC menggunakan alat bantu secara efektif dan efisien.	• Penugasan Proyek = 75% • Tes Praktik = 25%	40%

CAPAIAN PEMBELAJARAN MATA KULIAH (CPMK)

CPL	CPMK	Rumusan CPMK	Indikator	Penilaian	Bobot
KK4	**KK4.SIF202.2**	Mahasiswa dapat **membaca** rancangan pemrograman dasar dan pemrograman berorientasi objek dan framework MVC, serta menunjukkan hasil eksekusinya.	1. Mahasiswa dapat membaca rancangan pemrograman dasar serta berorientasi objek dan framework MVC dengan benar 2. Mahasiswa dapat menunjukkan hasil eksekusi rancangan pemrograman dasar serta berorientasi objek dan framework MVC dengan benar.	• Tes Tertulis = 75% • Penugasan = 25%	25%
KK6	**KK6.SIF202.1**	Mahasiswa dapat **mengevaluasi** kinerja aplikasi/sistem berdasarkan pendekatan berorientasi objek dan MVC	Mahasiswa dapat mengevaluasi kinerja aplikasi/sistem berdasarkan pendekatan berorientasi objek dan MVC dengan tepat sehingga menghasilkan aplikasi yang efisien.	• Tes Tertulis = 50% • Penugasan = 50%	10%

TOPIK MATERI (1)

1. Pengantar FPA
2. Java: Input & Output
3. Java: Percabangan
4. Java: Perulangan
5. Java: Method dan Parameter
6. Konsep Object Oriented
7. Studi Kasus Object Oriented
8. Class & Object
9. Object-Oriented Programming
10. Enkapsulasi & Multiclass
11. Pewarisan
12. Abstract & Interface
13. Polymorphism
14. Exception

TOPIK MATERI (2)

15. Konsep MVC (Model, View, Controller) dan FXML
16. Komponen Dasar FXML
17. TableView
18. Grafik
19. XML – Xstream
20. Multiple Scenes/Stages
21. Penerapan Struktur Data dalam Aplikasi GUI

22. Pengantar SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependencies Inversion)
23. Studi Kasus SOLID
24. - 25. Progres Tugas Besar
26. Review dan Persiapan UAS
27. - 28. Informatics Expo

ATURAN PERKULIAHAN

- Minimal hadir di 21 kali pertemuan (75%)
- Tiap pertemuan dapat terdiri dari pertemuan sinkron dan asinkron
- Presensi kehadiran di 15 menit awal
  - *Scanning* (Sistem UII Gateway)
  - Aktivitas Pembelajaran
- Tidak ada pengumpulan tugas atau ujian susulan
- Perizinan lewat sistem NKMD
- Tidak ada tugas atau ujian pengganti
- Menjunjung tinggi kejujuran

KONTRAK BELAJAR

- Setiap pertemuan diakhiri dengan quiz menggunakan aplikasi Safe Exam Browser (SEB)
- Tidak ada kuliah kosong, jika tidak terselenggara akan diganti di waktu yang lain dengan kesepakatan/pemberitahuan
- Bergabung di grup WA kelas
- Apresiasi poin diberikan kepada yang:
  - Aktif menjawab
  - Aktif bertanya

CATATAN PENTING

- Faktor-faktor yang menyebabkan kegagalan di mata kuliah ini:
  - Presensi < 75% (konsekuensinya tidak dapat ikut UAS dan ujian remedial)
  - Tidak mau bertanya ketika tidak paham
  - Tidak mau belajar dan mencoba/praktik dengan sungguh-sungguh
  - Mengandalkan ChatGPT dan sejenisnya
  - Tidak aktif dalam kelompok

PENGENALAN JAVA

Fundamen Pengembangan Aplikasi

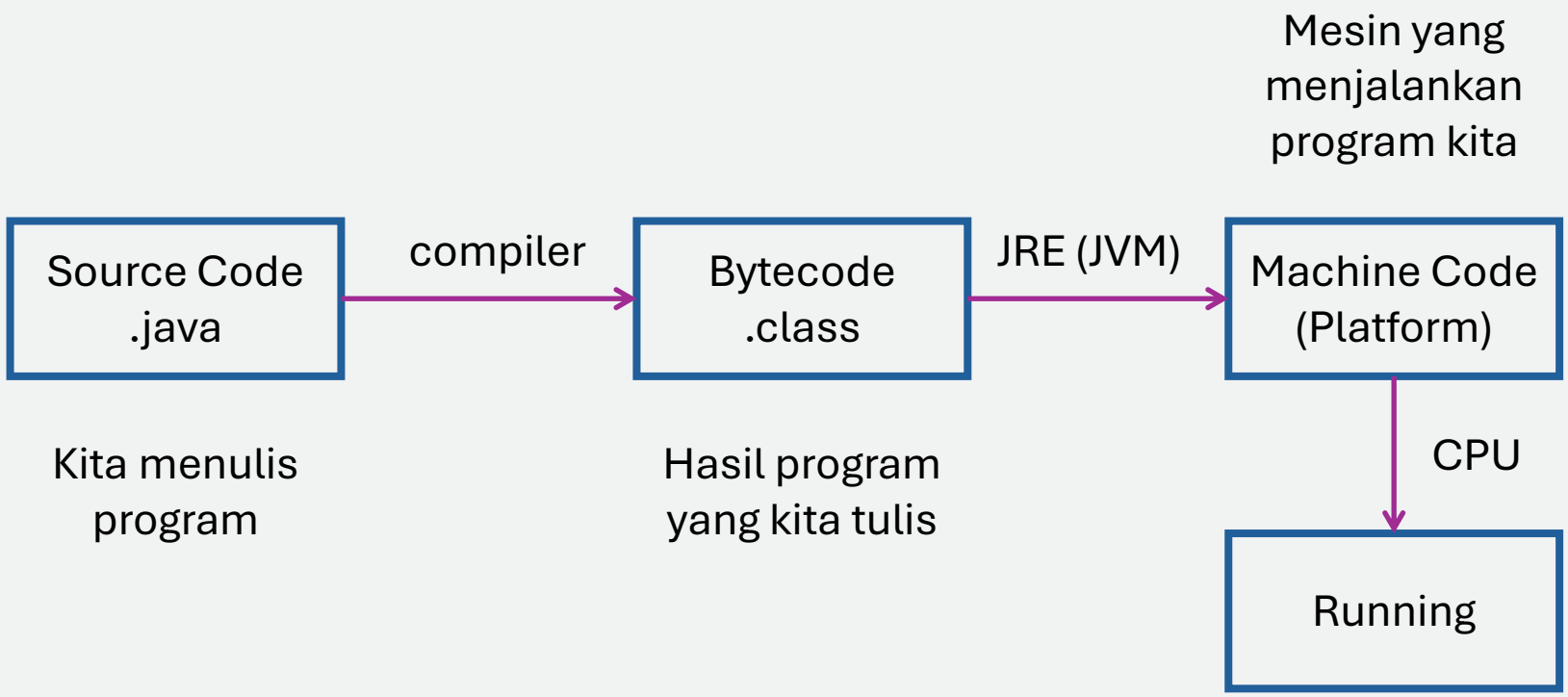
UNIVERSITAS ISLAM INDONESIA

PENGENALAN JAVA

- Bahasa Pemrograman Java
  - Open Source
  - Multiplatform
  - Object Oriented Programming

- Perkakas Java
  - JVM (Java Virtual Machine)
  - JRE (Java Runtime Environment)
  - JDK (Java Development Kit)

Source Code & Bytecode

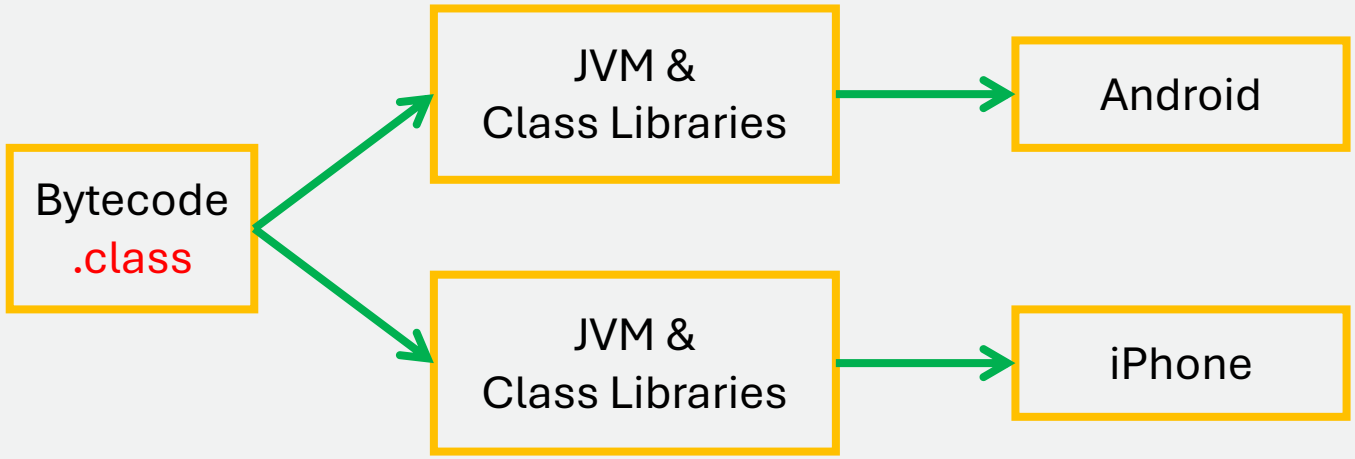


JRE, JVM, & JDK

- Java Runtime Environment (JRE)
  - Menjalankan bytecode (.class)
  - Platform dependent
  - Terdiri dari mesin dan library

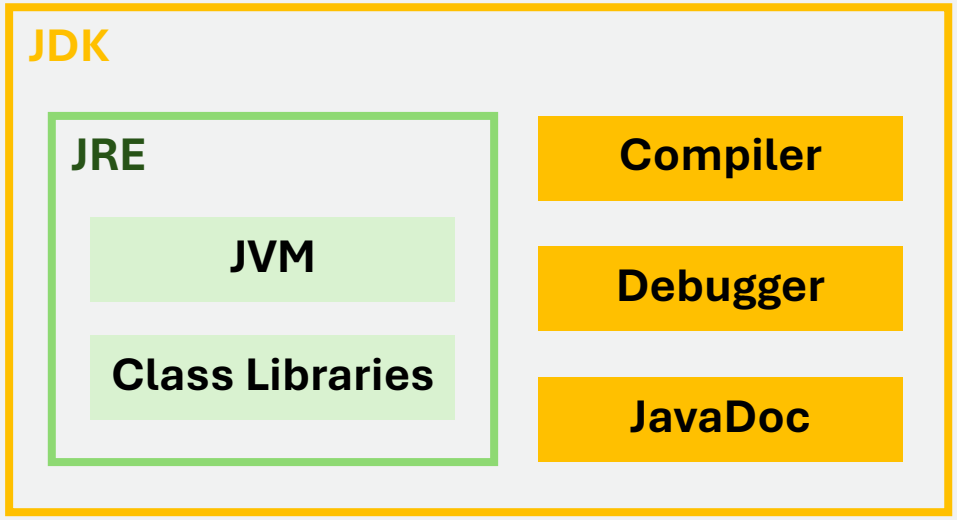
JRE, JVM, & JDK

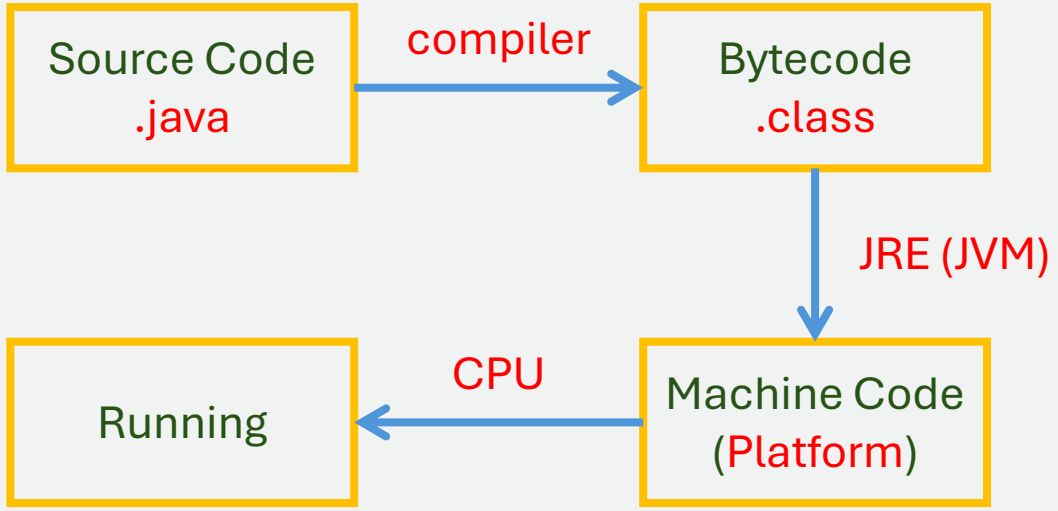
- Java Virtual Machine (JVM)
  - Menerjemahkan bytecode ke kode mesin dengan memanfaatkan class library
  - Platform dependent



JRE, JVM, & JDK

- Java Development Kit (JDK)
  - Kit untuk mengembangkan (membuat, menguji, mendokumentasi, menjalankan) aplikasi dengan bahasa Java





KAKAS YANG DIGUNAKAN

- IDE Visual Studio Code (VS Code)
  - IDE *(Integrated Development Environment)* yang digunakan untuk menulis, mengoreksi, dan mengompilasi program Java.
- Java SE Development Kit 9 (JDK 9)
  - Paket aplikasi untuk pengembangan aplikasi berbasis Java.
- Scene Builder
  - Aplikasi yang digunakan untuk pengembangan tampilan antarmuka. Scene Builder dapat terintegrasi dengan VS Code.
- Star UML
  - Aplikasi yang digunakan untuk membantu pembuatan diagram UML (Unified Modelling Language).

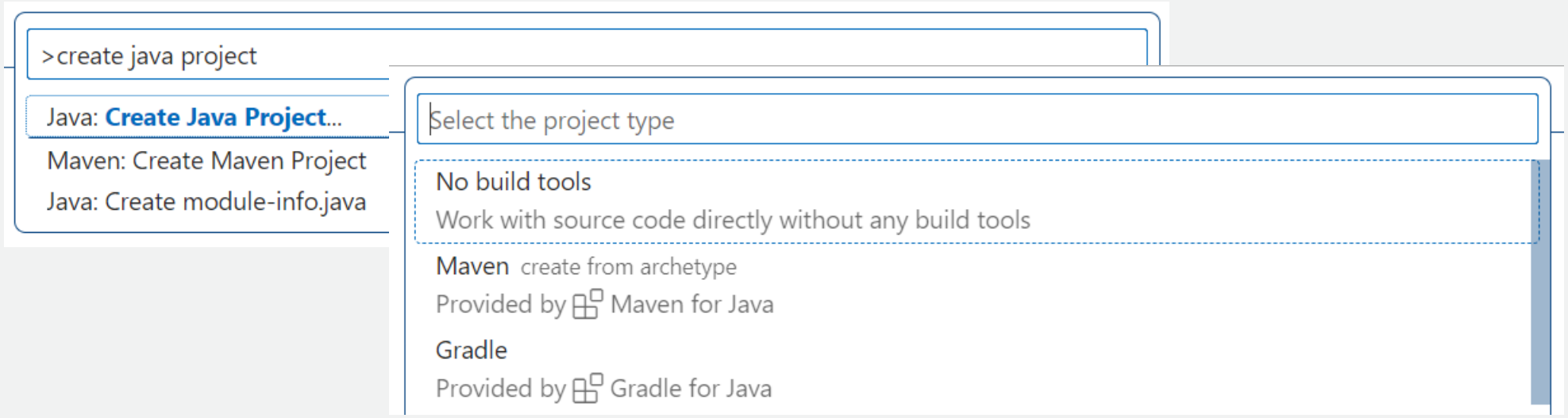
URUTAN INSTALASI

- Hal yang perlu diperhatikan adalah urutan instalasi dari kakas-kakas yang digunakan pada mata kuliah ini
- Berikut adalah urutan instalasi yang disarankan untuk memudahkan proses sinkronisasi:
  - Java SE Development Kit 9 (JDK 9)
  - Scene Builder
  - Visual Studio Code
  - Star UML
- Pada VS Code, tambahkan Extension Pack for Java

Hello, World!

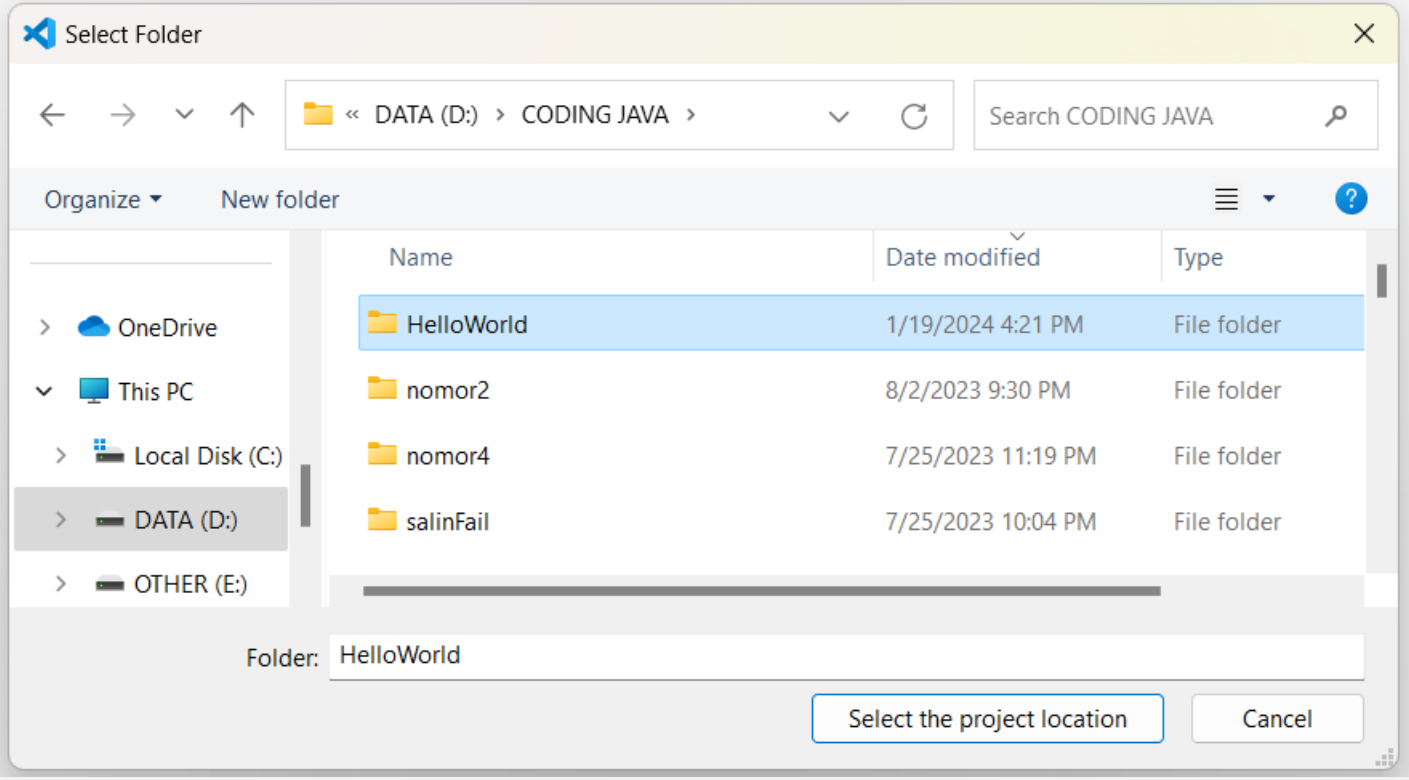
Membuat Project Java di VS Code (1)

1. Buat folder sebagai tempat menyimpan file-file project
2. Buka aplikasi VS Code kemudian pilih menu View > Command Palette
3. Ketikkan perintah create java project lalu pilih opsinya.
4. Pilih No build tools lalu klik.



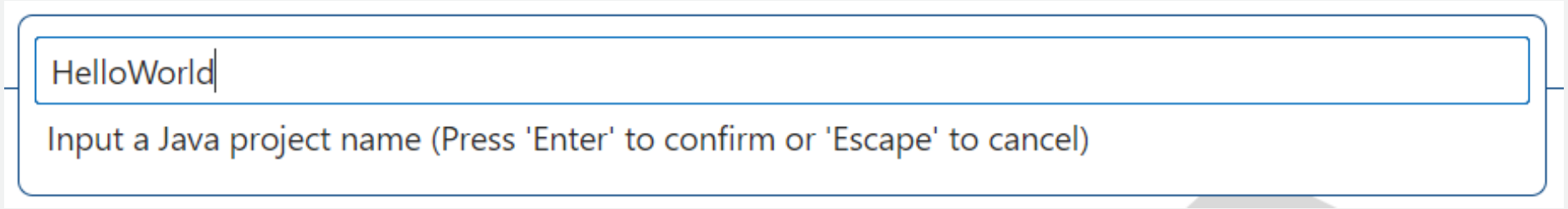
Membuat Project Java di VS Code (2)

1. Pilih folder project yang sudah dibuat.
2. Kemudian klik tombol Select the project location

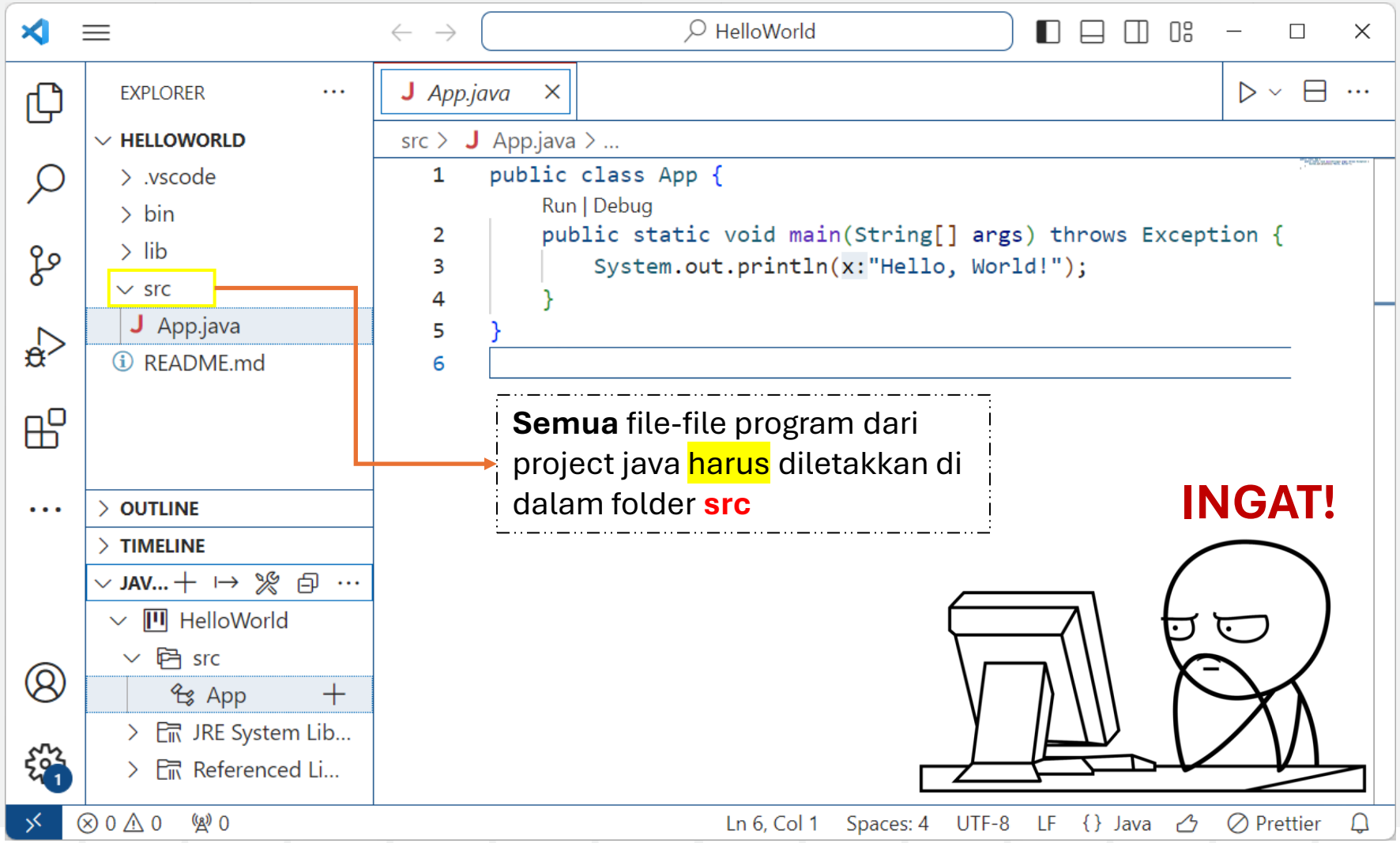


Membuat Project Java di VS Code (3)

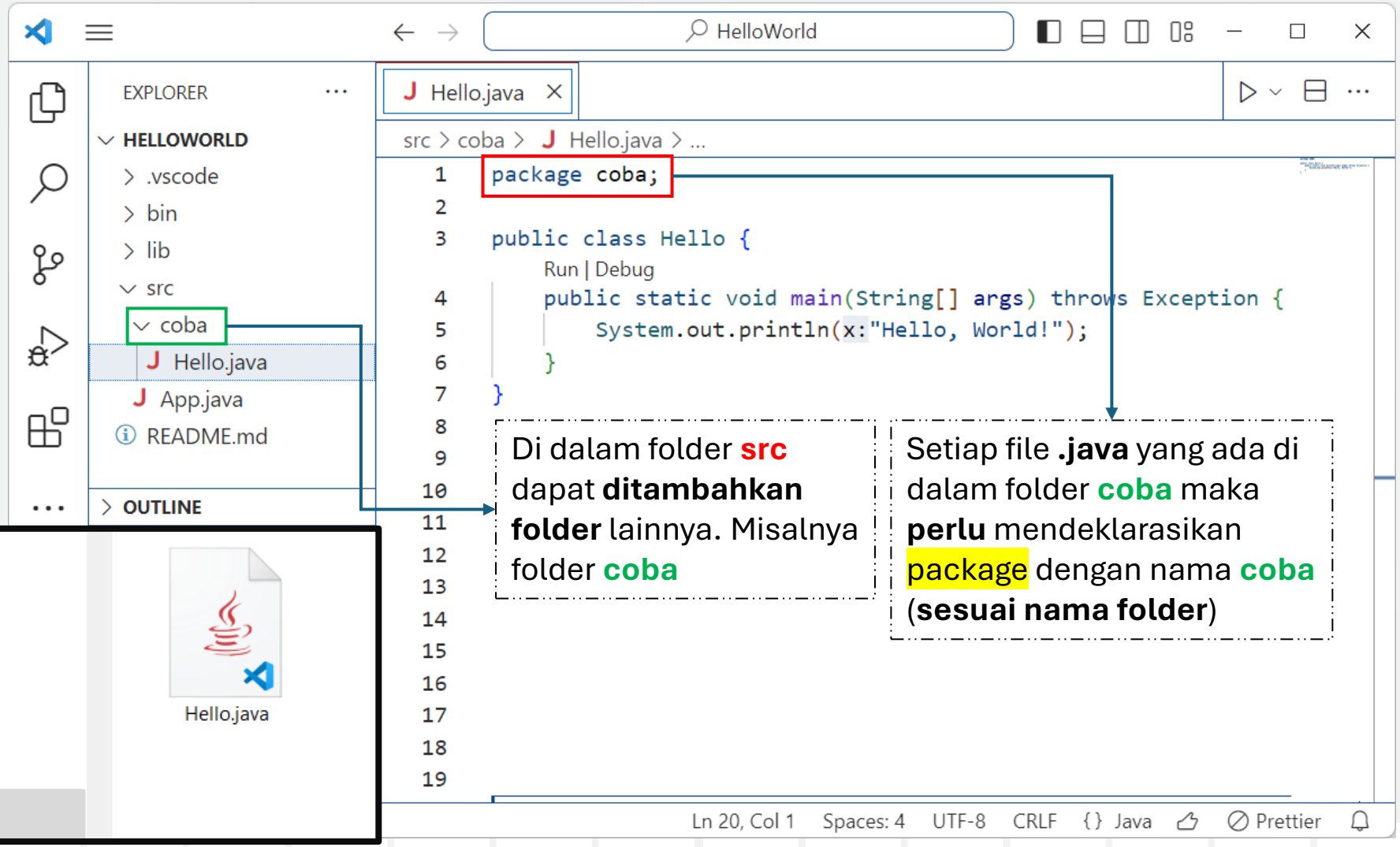
- Setelah itu, ketikkan nama project yang Anda inginkan
- Lalu tekan tombol Enter
- Jika berhasil maka jendela project akan terbuka dan siap digunakan



Membuat Project Java di VS Code (4)



Membuat Project Java di VS Code (5)



MENULIS PROGRAM JAVA

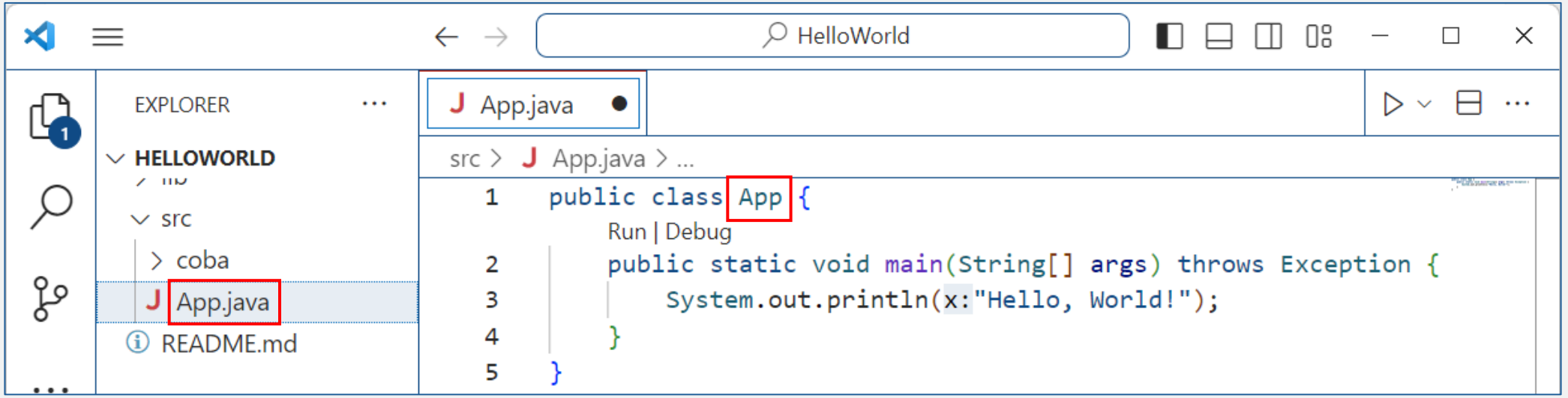
```java
public class App {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws Exception {
        // TODO code application logic here
        System.out.println("Hello, World!");
    }
}
```

ATURAN PENULISAN PROGRAM JAVA (1)

- **Nama Variabel, Method, dan Class.**
  - Bahasa pemrograman Java bersifat ***case sensitive*** yang berarti bahwa **namavariabel** TIDAK sama dengan **NamaVariabel**.
  - Tidak boleh menggunakan ***reserved word*** atau ***keyword*** java karena keyword-keyword ini telah "dipesan" oleh program java
    - Beberapa contoh reserved word : **abstract, continue, for, new, switch, default, package, synchronized, boolean, do, if, private, this, break, double, implements, protected, else, import, public, throws, case, enum, return, catch, extends, int, short, try, char, final, interface, static, void, class, finally, long, const, float, native, super, while**
  - Tidak diperbolehkan untuk menggunakan **simbol-simbol operator**
    - Misalnya: ***kelasKu**, **%umur**, dll
  - Pemberian nama tidak boleh diawali dengan **bilangan**
    - Misalnya: **7namaVariabel**, **99design**
  - Selalu dimulai dengan **huruf abjad**, atau **underscore** (_), atau **tanda dolar** ($) kemudian dapat pula diikuti dengan angka.
  - Misalnya: **_namaVariabel**, **$kodeDasar**, **nama123**, dll

ATURAN PENULISAN PROGRAM JAVA (2)

- **Penulisan Class.**
  - **Nama file** untuk ***class public*** **HARUS SAMA** dengan **nama class**.
    - Sebagai contoh ketika Anda membuat kelas public bernama **ContohKelas** maka Anda **HARUS** menyimpan dengan nama **ContohKelas.java** dan ***jika bukan public maka bebas***.
  - **Catatan Penting**: Aturan penamaan pada kelas menggunakan konsep **CamelCase** atau huruf kapital di setiap karakter pertama dan tanpa spasi.



STRUKTUK PEMROGRAMAN JAVA

Fundamen Pengembangan Aplikasi

UNIVERSITAS ISLAM INDONESIA

Konvensi Pemrograman Java

- Ekstensi File
  - Java Source Code : .java
  - Java Bytecode : .class
- Struktur Source Code
  - Komentar Awal (Tentatif)
  - *Package*
  - *Import Statement*
  - Deklarasi Kelas (Class)

```java
/*
 * Ini adalah aplikasi utama
 * Menggunakan class Scanner
 */

package coding;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

    }
}
```

```java
package coba;

import java.util.Scanner;

public class Coba {


   public static void main(String[] args) {

      Scanner baca = new Scanner(System.in);
      int nim = baca.nextInt();
      System.out.println("NIM saya: " + nim);

      //Ini tidak dieksekusi
      /* Ini juga tidak dieksekusi */
   }

}
```

Nama folder yang berisi sekumpulan file .java Anda

Definisi dari library yang digunakan

Deklarasi nama kelas (class)

Method utama yang akan **dieksekusi** pertama kali

Statemen atau Ekspresi yang ada di dalam method utama

```
package coba;

import java.util.Scanner;

public class Coba {


   public static void main(String[] args) {

      Scanner baca = new Scanner(System.in);
      int nim = baca.nextInt();
      System.out.println("NIM saya: " + nim);

      //Ini tidak dieksekusi
      /* Ini juga tidak dieksekusi */
   }

}
```

Blog Program dibuka dengan kurung kurawal buka (**{**) dan ditutup dengan kurung kurawal tutup (**}**)
Berbeda dengan Python yang cukup menggunakan indentasi

Blok program (blok **class**)

Blok program (blok **method**)

```java
package coba;

import java.util.Scanner;

public class Coba {


   public static void main(String[] args) {

      Scanner baca = new Scanner(System.in);
      int nim = baca.nextInt();
      System.out.println("NIM saya: " + nim);

      //Ini tidak dieksekusi
      /* Ini juga tidak dieksekusi */
   }

}
```

Statement atau Ekspresi
- Bagian terkecil dari program
- **Harus** diakhiri dengan titik koma (**;**)

```java
package coba;

import java.util.Scanner;

public class Coba {


   public static void main(String[] args) {

      Scanner baca = new Scanner(System.in);
      int nim = baca.nextInt();
      System.out.println("NIM saya: " + nim);

      //Ini tidak dieksekusi
      /* Ini juga tidak dieksekusi */
   }

}
```

Komentar adalah bagian program yang ***tidak dieksekusi***

ALUR SEKUENSIAL

- Instruksi dieksekusi satu per satu, mulai dari instruksi pertama sampai instruksi terakhir.
- Setiap instruksi dieksekusi tepat satu kali.

- Contoh:

```
public static void main(String[] args) {
        int a = 5;
        int b = 12;
        int c = a + b;
        System.out.print(c);
}
```

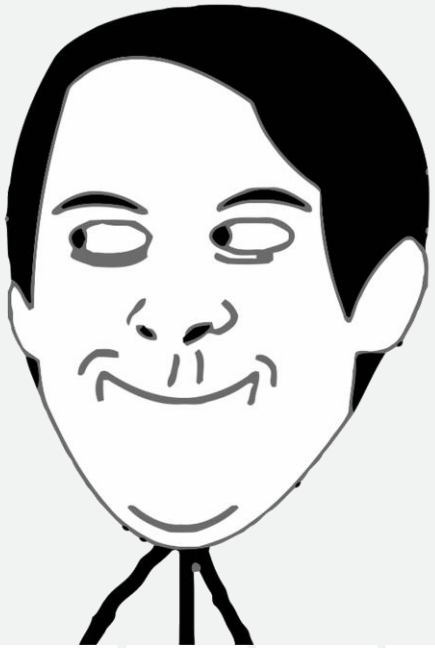
ALUR SEKUENSIAL

- Instruksi dieksekusi satu per satu, mulai dari instruksi pertama sampai instruksi terakhir.
- Setiap instruksi dieksekusi tepat satu kali.

- Contoh:

```
public static void main(String[] args) {
        int a = 5;
        int c = a + b;
        int b = 12;
        System.out.print(c);
}
```

Jika ditukar?



Indentasi Penulisan Source Code

- Penulisan kode dalam satu baris disarankan TIDAK LEBIH dari 80 karakter
- Indentasi menggambarkan kelompok/ bagian dari satu blok diagram

```
public void ortu(){
        //Aku bagian dari public void ortu

        public void anak(){
                //Aku bagian dari public void anak
        }
}
```

Menjorok ke kanan

Menjorok ke kanan

Indentasi Penulisan Source Code

- Berbeda dengan dalam Python, indentasi Java BUKAN bagian dari struktur kode (tidak berpengaruh terhadap logika program)
- Kedua program di bawah ini tidak berbeda, walaupun indentasi tidak
- sama

```
public void ortu(){
    // … kode-kode lain
    public void anak(){
        if (A==B){
            C = A*A;
        }
    }
}
```

```
public void ortu()
{
// … kode-kode lain
public void anak()
{if (A==B){C = A*A;
}}}
```

Latihan Menampilkan Teks

```
public static void main(String[] args) {
        String nama = "Nama Saya";

        System.out.print("Nama saya adalah: ");
        System.out.println(nama);
}
```

- Ganti nilai “Nama Saya” dengan nama Anda!
- Tambahkan baris program untuk menampilkan
  “NIM Saya: *<isi dengan NIM Anda>*”

TERIMA KASIH