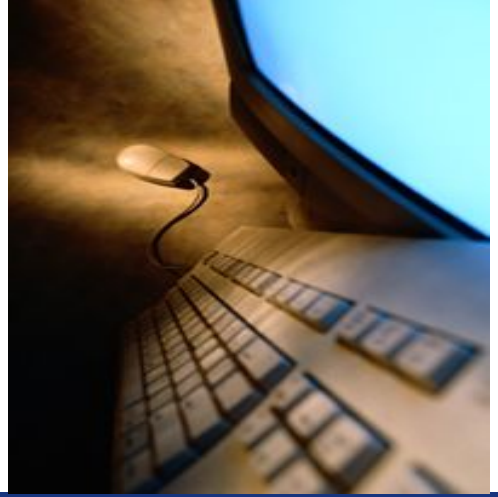
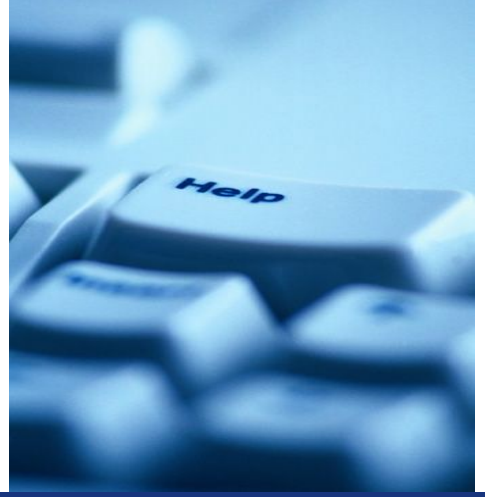
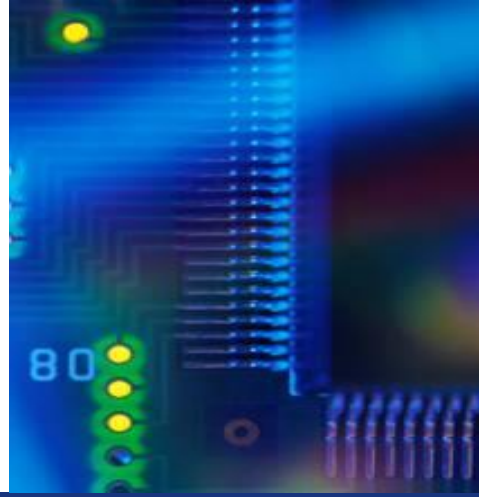
UNIVERSITAS
BUDI LUHUR



FAKULTAS TEKNOLOGI INFORMASI

Mobile Programming

[PG119/ 3 SKS]

Riskiana Wulan, MKom

REVIEW JAVA


Java Basics for Android Development

What are Variables?

- Variables are used to store data that can be used and manipulated in a program.
- Each variable has a type that determines the kind of data it can store.

Basic Data Types:

- int: stores integers (whole numbers) like 10, -5
- double: stores decimal numbers like 3.14, 9.99
- boolean: stores true or false values
- String: stores text, enclosed in double quotes, e.g., "Hello"

Data types example:

```
int age = 25;

double price = 19.99;

boolean isAvailable = true;

String name = "John Doe";
```


Basic Operators

Arithmetic Operators:

- +: Addition, -: Subtraction, *: Multiplication, /: Division
- **Example**:
  - int sum = 5 + 10; // sum is 15
  - double result = 10.0 / 4; // result is 2.5

Relational Operators (comparison):

- ==: equal to, !=: not equal to
- **Example**:
  - boolean isEqual = (5 == 5); // true
  - boolean isNotEqual = (5 != 3); // true


Control Structures

What are Control Structures?

- They determine the flow of the program, controlling the decision-making.

Types of Control Structures:

- **if-else**: Executes different code blocks based on a condition.
- **for loop**: Repeats a block of code a specific number of times.
- **while loop**: Repeats a block of code while a condition is true.

For loop example

```
for (int i = 0; i < 5; i++) {
    System.out.println("i is: " + i);
}
```

If-Else example:

```
int number = 10;
if (number > 5) {
    System.out.println("Number is greater than 5");
} else {
    System.out.println("Number is less than or equal
to 5");
}
```

While Loop Example

```
int i = 0;
while (i < 5) {
    System.out.println("i is: " + i);
    i++;
}
```


Object Oriented Programming

What are Classes and Objects?

Class: A blueprint or template to create objects.

- Contains attributes (variables) and methods (functions).

Object: An instance of a class.

- **Example**: A Car class can have attributes like model and color, and methods like startEngine().

Example of a Class

Car.java

```
public class Car {
    String model;
    String color;

    void startEngine() {
        System.out.println("Engine started");
    }
}
```


What are Methods?

Methods:

- Define actions or behaviors that objects can perform.
- Can accept parameters and return values.
- **Syntax**:

<returnType> <methodName>(<parameters>)
{ // body }

```
public void drive(int speed) {
   System.out.println("Driving at " + speed + "
km/h");
}
```


What are Constructors?

Constructors:

- Special methods used to initialize new objects.
- Same name as the class and no return type.
- Automatically called when a new object is created.

```
public class Car {
    String model;
    String color;

    // Constructor
    public Car(String model, String color) {
        this.model = model;
        this.color = color;
    }
```


Access Modifiers

- **What are Access Modifiers?**
  - Control the visibility of class members (attributes and methods).
- **Types of Access Modifiers**:
  - public: Accessible from anywhere.
  - private: Accessible only within the same class.
  - protected: Accessible within the same package and subclasses.

Example of Access Modifiers

```
public class Car {
    public String color;   // Accessible from anywhere
    private String model; // Only accessible within the class

    public String getModel() {
        return model;
    }

    public void setModel(String model) {
        this.model = model; // Accessing private variable
    }
}
```


Summary

- **Variables and Data Types**: Store and manipulate data in your program.
- **Basic Operators**: Perform operations like addition, subtraction, and comparisons.
- **Control Structures**: Manage the flow of your program with conditions and loops.
- **Classes and Objects**: Blueprint of data and behavior in OOP.
- **Methods**: Actions that objects can perform.
- **Constructors**: Initialize objects with specific values.
- **Access Modifiers**: Control the visibility of class members.


Hands-On Activity: Build a Java Class

Content:

- Task: Create a Person class with name and age, and a method to display info

Instructions:

- Step 1: Define class and attributes
- Step 2: Add a constructor
- Step 3: Add a method displayInfo()
- Step 4: Create a main method to instantiate and call the method


Java vs Android: What's Different?

Java: A general-purpose programming language used to write the application logic, algorithms, and core functionality.

Android Development: Uses Java (or Kotlin) as the language to write the app's logic, but also includes the **Android SDK (Software Development Kit)** to provide components that make up mobile applications.

Android SDK:

- Provides libraries, tools, and frameworks for building Android apps.
- Includes components that Java alone does not cover, such as:
  - **UI Components**: Buttons, TextViews, and custom layouts.
  - **Activities**: Entry point for user interaction.
  - **Intents**: Used to launch other activities or services.
  - **XML-based Layouts**: Used to define the user interface visually.


Key Android Concepts

Activity:

- A single, focused screen of an app with which the user interacts.
- Example: A login screen, a settings page, etc.
- **Lifecycle**: An activity goes through several states like onCreate(), onStart(), onPause(), and onDestroy().

Intent:

- A messaging object that lets components interact with each other.
- Example: Launching a new activity, sharing data between apps.
- Types:
  - **Explicit Intent**: Specify which activity to open.
  - **Implicit Intent**: Request an action without specifying a particular activity.

XML-based Layouts:

- Android uses **XML** to define the user interface.
- Layouts, buttons, text fields, and other UI components are defined in XML files, separate from the Java code.


Summary

- **Java**:
  - Handles the core logic and computation.
  - Works across multiple platforms.
- **Android Development**:
  - Uses Java (or Kotlin) for the logic, but Android SDK adds:
    - **UI Components**
    - **Activity and Intent system**
    - **XML-based layouts** for visual interface.

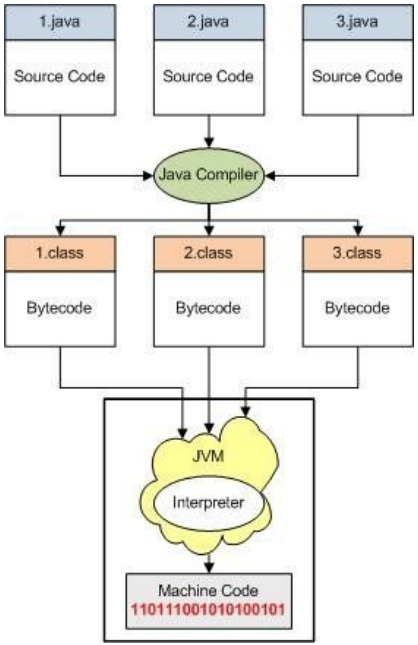
Java is essential for writing the logic of Android apps.

The **Android SDK** provides the necessary tools to create an interactive and functional mobile app experience with components like Activities, Intents, and XML layouts.

Reference

https://drive.google.com/drive/folders/1qsXsPrHUWnf7LR333bTBJ3py9CUA30Un?usp=sharing

How Java Programs work?



Source Code: You write Java code in `.java` files.

- The source code is written by developers using a text editor or an IDE like **Eclipse** or **IntelliJ or Android Studio**.
- It is stored in files with the **`.java`** extension.

Compilation: The Java compiler (`javac`) converts the code into platform-independent **bytecode**.

- T**he Java source code** is not directly understood by computers. The **Java Compiler (`javac`)** translates the **`.java`** files into **bytecode**.
- **Bytecode (.class)**: The result of the compilation is a `.class` file, containing Java **bytecode**. This bytecode is platform-independent and can run on any device that has a **Java Virtual Machine (JVM)**.

Execution: The **JVM** interprets or compiles bytecode into machine code for the operating system to execute.

Tools: Use the **JDK** to write and compile programs, and the **JRE** to run them.

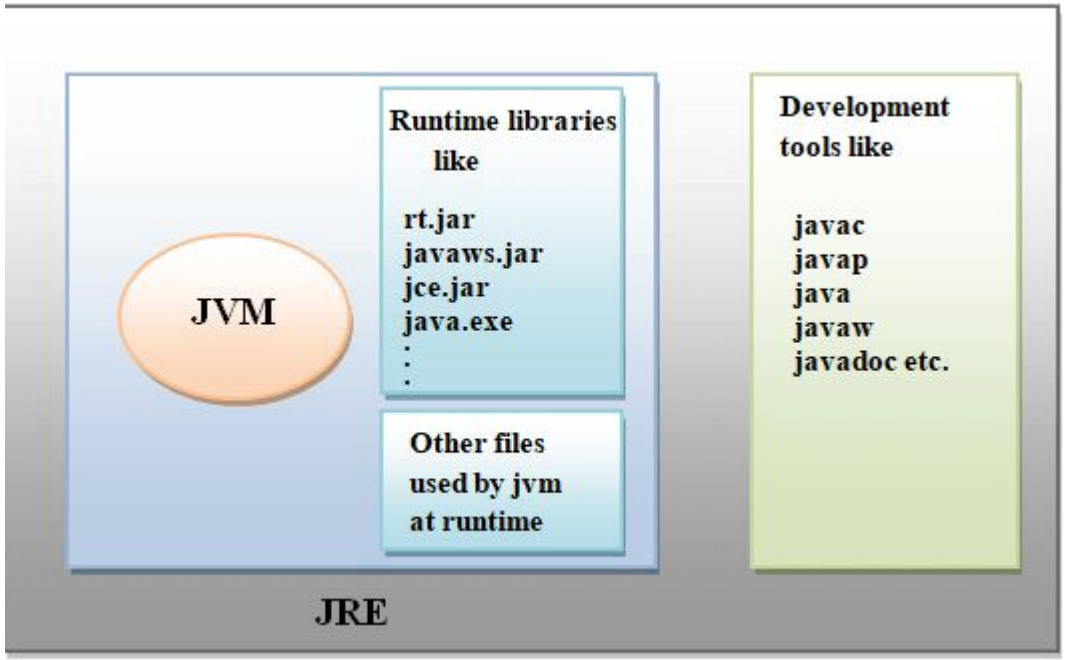

Execution Process

- **Java Virtual Machine (JVM)**:
  - The **JVM** is a key component of the Java platform. It executes the **bytecode**.
  - The JVM converts bytecode into **machine-specific instructions** for the operating system and hardware where the program is running.
  - The JVM ensures **platform independence**, meaning the same bytecode can run on any system that has a compatible JVM.

Java Runtime Environment (JRE) vs Java Development Kit (JDK)

- **Java Runtime Environment (JRE)**:
  - A subset of the Java Development Kit (JDK). It contains everything needed to **run** Java programs (JVM, libraries, and other necessary components).
- **Java Development Kit (JDK)**:
  - The full development environment for writing Java programs. It includes:
    - The **JRE** (to run programs).
    - **Development tools** like the compiler (javac) and debugger.

JDK



https://www.refreshjava.com/java/jdk-jre-jvm


Exercise

Kerjakan Soal dengan ketentuan sebagai berikut:

1. **NIK dengan akhiran 1 atau 6 : Soal no.1**
2. **NIK dengan akhiran 2 atau 7 : Soal no 2**
3. **Nik dengan akhiran 3 atau 8 : Soal no 3**
4. **NIK dengan akhiran 4 atau 9 : Soal no 4**
5. **NIK dengan akhiran 5 atau 0 : Soal no.5**

Kumpulkan screenshot dari console log beserta java class pada Java exercise (elearning)


Soal 1

Buatlah Car class dengan attributes sebagai berikut:

- **String model**
- **String color**
- **int year**

Ketentuan:

- **Tambahkan Constructor dengan menginisialisasi semua atribut**
- **Buat sebuah method startEngine() yang mencetak "The <color> <model> car from <year> is starting its engine."**
- **Pada main() method, buatlah dua object car dan masing-masing panggil startEngine() method**


Soal 2

Buatlah Person class dengan atribute:

- **String name**
- **int birthYear**

Ketentuan:

- **Tambahkan Constructor yang menginisialisasi semua atribute**
- **Buat method calculateAge(int currentYear) yang mengembalikan umur person berdasarkan tahun sekarang.**
- **Pada main() method, buatlah sebuah object dari Person class, panggil calculateAge() method dan print umur dengan format**

  “Name: <name>, Age: <calculated_age>”


Soal 3

Buatlah sebuah class Book dengan atribute berikut:

- **String title**
- **String author**
- **int pages**

Ketentuan:

- **Tambahkan constructor untuk menginisialisasi semua atribute**
- **Buat method isThick() yang mengembalikan nilai true jika Book memiliki lebih dari 300 pages, sebaliknya kembalikan nilai false**
- **Pada main() method, buatlah sebuah object book dan print apakah buku tersebut tebal (is thick) dengan message” The book <title> by <author> is thick: <true/false>**


Soal 4

Buat class Laptop dengan atribute sebagai berikut

- **String brand**
- **Double price**

Ketentuan:

- **Tambahkan constructor untuk menginisialisasi semua atribute**
- **Buat method isExpensive() yang mengembalikan nilai "true" jika harga dari laptop tersebut lebih dari 50000, sebaliknya kembalikan nilai "false"**
- **Pada main() method, buatlah dua object laptop, lalu print apakah mereka mahal (expensive) atau tidak dengan format : "The <brand> is expensive: <true/false>**

Soal 5

Buat Movie class dengan attibut berikut

- **String title**
- **double rating**

Ketentuan:

- **Tambahkan constructor untuk menginisialisasi semua attribut**
- **Buat method isBlockbuster() yang mengembalikan nilai true jika ratingnya 8 keatas, sebaliknya kembalikan nilai false**
- **Pada main() method, buatlah sebuah movie object dan print apakah termasuk blockbuster dengan format: "The movie <title> is a blockbuster: <true/false>**


Soal Android Application

Adding a Third Activity and Passing Data Between Activities

Objective:

1. Create a ThirdActivity and pass data from SecondActivity to ThirdActivity.
2. Send data back from ThirdActivity to MainActivity.

Project dikumpulkan maksimal Hari Jumat tgl 11 Oktober, pukul 17.00