



Jurusan Teknik Komputer dan Informatika

Politeknik Negeri Bandung

# Pertemuan 1 Pengenalan Teknik Pemrograman Praktikum

D4 Kelas 1A/1B

Dosen Pengampu :  
Zulkifli Arsyad

- Informasi Perkuliahan
  - Ketentuan Perkuliahan
  - Silabus Perkuliahan
  - Referensi
  - Tujuan Pembelajaran Pertemuan 1
- Pengantar Pemrograman Berorientasi Obyek
- Praktikum Pertemuan 1

## **Beban Perkuliahan**

- Praktikum 1 SKS = 170 Menit
- 07.00 – 09.30 (2A)
- 09.50 – 12.20 (2B)

## **Komponen Perkuliahan**

- Tugas, Quiz : 30%
- ETS : 30%
- EAS : 40%

## **Model Pembelajaran**

- Pemaparan Materi & Diskusi
- Praktikum Dikelas
- Quiz
- Penugasan
- Eksplorasi Mandiri
- Tugas Besar

1. Pertemuan 1 : Pengenalan Teknik Pemrograman (Silabus, Kontrak Kuliah, Komponen dasar PBO)
2. Pertemuan 2 : Fundamental Java Programming 1 (Data type, Variable, Operator, String)
3. Pertemuan 3 : Fundamental Java Programming 2 (Input Output, Big Integer, Control Flow, Array)
4. Pertemuan 4 : Class & Object 1 (Class, Object, Encapsulation)
5. Pertemuan 5 : Class & Object 2 Class Design Hint
6. Pertemuan 6 : Class Relationship
7. Pertemuan 7 : Inheritance, Abstract Class , Interface
8. Pertemuan 8 : Polymorphism
9. Pertemuan 9 : Exception Handling & Debugging
10. Pertemuan 10 : Assertion (JUnit)
11. Pertemuan 11 : Java Collection Framework
12. Pertemuan 12 : Clean Code
13. Pertemuan 13 : Generic Programming
14. Pertemuan 14 : Thread

- Horstmann, Cay S. "Core Java Volume I-Fundamentals Ninth Edition", PrenticeHall, 2013

- Waktu Perkuliahan
- Media Komunikasi & Diskusi : WaG
- Portal Pembelajaran : Elearning Polban
- Presensi : Aplikasi Presensi Akademik
- Monitoring Progres : Google Sheet

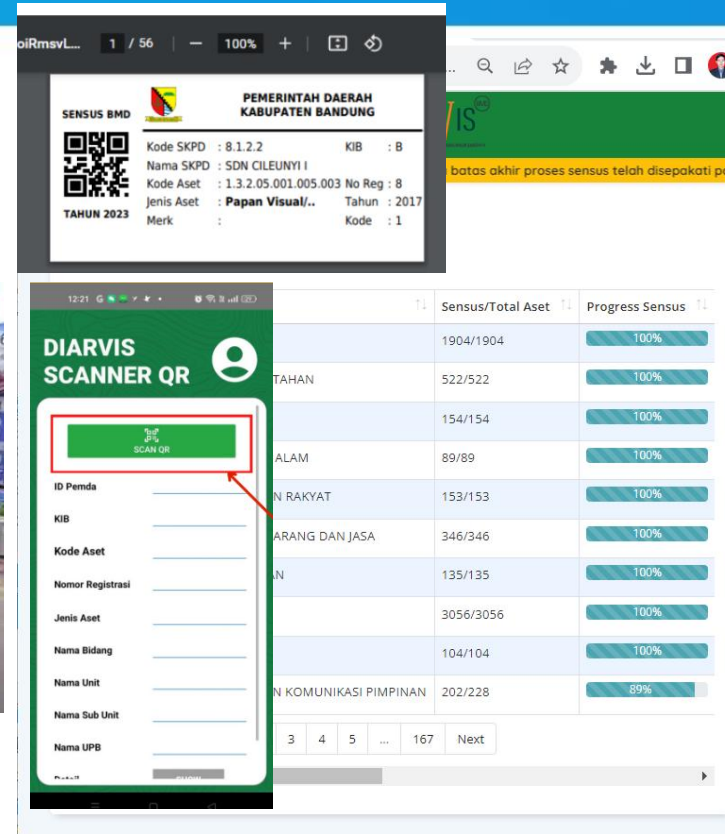
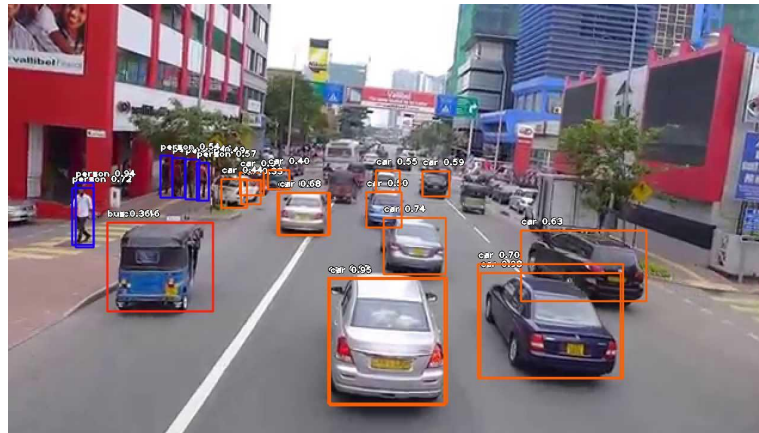
# Tujuan Pembelajaran Taksonomi Bloom

Indikator		Pengetahuan	Kognitif
<b>Audience</b>	Mahasiswa D4 Kelas 1A dan 1B		
<b>Behavior</b>	Mahasiswa dapat Mengenal mata kuliah Pemrograman Berorientasi Obyek	Pengetahuan Konseptual	C2 (Memahami) Pengetahuan Konseptual
	Mahasiswa dapat mengenali dan menyebutkan Teknologi saat ini yang menggunakan penerapan Pemrograman Berorientasi Object	Pengetahuan Konseptual	C2 (Memahami)
	Mahasiswa dapat melakukan setup Software environment	Pengetahuan Prosedural	C3 (Implementing)
	Mahasiswa dapat menganalisis input proses dan output yang dihasilkan potongan source code kasus yang dikaitkan dengan konsep pada mata kuliah DDP.	Pengetahuan Prosedural Pengetahuan Metakognitif	C3 (Implementing) C4 (aNALISIS)
<b>Condition</b>	Praktikum dilakukan di lab kelas		
<b>Degree</b>	tingkat akurasi Mahasiswa dapat menyelesaikan kasus $\geq 80\%$ selesai		

# Pengantar Teknik Pemrograman (P)



# Review



Python, Open AI

Python, Open CV

Andorid, Laravel

Android Java / Kotlin,  
Xamarin,

Menggunakan Penerapan Pemrograman Berorientasi Obyek

- Pemrograman Berorientasi Objek (PBO) adalah paradigma pemrograman yang berfokus pada pemodelan dunia nyata menggunakan konsep objek. Dalam PBO, program diorganisasi dalam bentuk objek-objek yang merepresentasikan entitas atau konsep dalam dunia nyata. Setiap objek memiliki atribut (data) dan metode (fungsi) yang berhubungan dengan objek tersebut.

# Contoh Object & Class

python

Copy code

```
# Contoh Class: Manusia
class Manusia:
    def __init__(self, nama, usia):
        self.nama = nama
        self.usia = usia

    def perkenalan(self):
        print(f"Halo, nama saya {self.nama} dan saya berusia {self.usia} tahun")

# Membuat objek-objek dari class Manusia
orang1 = Manusia("Alice", 25)
orang2 = Manusia("Bob", 30)

# Memanggil metode perkenalan untuk masing-masing objek
orang1.perkenalan()
orang2.perkenalan()
```

java

Copy code

```
// Contoh Class: Manusia
class Manusia {
    String nama;
    int usia;

    // Konstruktor
    Manusia(String nama, int usia) {
        this.nama = nama;
        this.usia = usia;
    }

    void perkenalan() {
        System.out.println("Halo, nama saya " + nama + " dan saya berusia " + usia + " tahun");
    }
}

public class ContohOOP {
    public static void main(String[] args) {
        // Membuat objek-objek dari class Manusia
        Manusia orang1 = new Manusia("Alice", 25);
        Manusia orang2 = new Manusia("Bob", 30);

        // Memanggil metode perkenalan untuk masing-masing objek
        orang1.perkenalan();
        orang2.perkenalan();
    }
}
```

# Konsep Utama PBO

## Interface

Interface adalah sebuah kontrak yang berisi deklarasi metode-metode tanpa implementasi. Semua metode dalam sebuah interface secara otomatis bersifat abstrak dan harus diimplementasikan oleh kelas yang mengimplementasikan interface tersebut.

## Abstract Class

Kelas abstrak dapat memiliki metode abstrak (tidak memiliki implementasi) dan metode konkrit (memiliki implementasi).

## Polimorpshm

Polimorfisme mengacu pada kemampuan berbagai objek untuk merespons metode yang sama dengan cara yang berbeda. Hal ini memungkinkan objek yang berbeda untuk merespons metode serupa berdasarkan sifat dan perilaku masing-masing

## Kelas & Object

Kelas adalah blueprint atau template yang mendefinisikan struktur dan perilaku suatu objek. Objek adalah instansi konkret dari sebuah kelas yang memiliki data dan perilaku sesuai dengan definisi kelasnya.

## Enkapsulasi

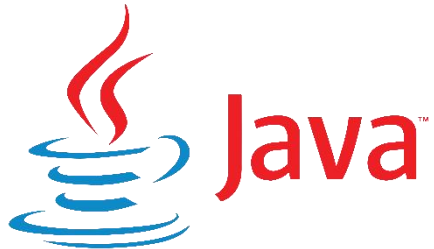
Enkapsulasi mengacu pada konsep menyembunyikan informasi. Hanya metode-metode tertentu yang dapat mengakses atau memanipulasi data internal dari suatu objek. Ini membantu dalam menjaga integritas data dan mencegah akses yang tidak sah

## Pewarisan / Inheritance

Pewarisan memungkinkan kelas yang lebih spesifik (kelas turunan) untuk mewarisi sifat dan perilaku dari kelas yang lebih umum (kelas induk). Ini membantu dalam membangun hierarki kelas dan mengurangi duplikasi kode.



# Bahasa Pemrograman Object Oriented



Java adalah salah satu bahasa pemrograman berorientasi objek yang sangat populer. Ia menawarkan dukungan yang kuat untuk konsep-konsep PBO seperti kelas, objek, pewarisan, enkapsulasi, dan polimorfisme



Kotlin adalah bahasa pemrograman modern yang dapat dijalankan di atas Java Virtual Machine (JVM), Android. Ia memiliki sintaks yang lebih ringkas dan ekspresif daripada Java, serta mendukung konsep-konsep PBO



Swift adalah bahasa pemrograman yang dikembangkan oleh Apple untuk pengembangan aplikasi iOS, macOS, watchOS, dan tvOS. Ia mendukung PBO dengan sintaks modern dan fitur-fitur seperti kelas, pewarisan, enkapsulasi, dan polimorfisme.



Python adalah salah satu bahasa pemrograman yang sangat populer dan sering digunakan dalam pengembangan berbagai jenis aplikasi kecerdasan buatan (AI).

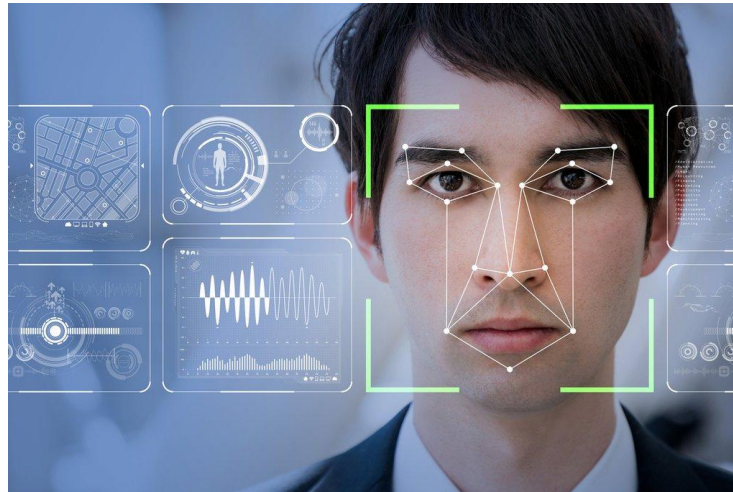
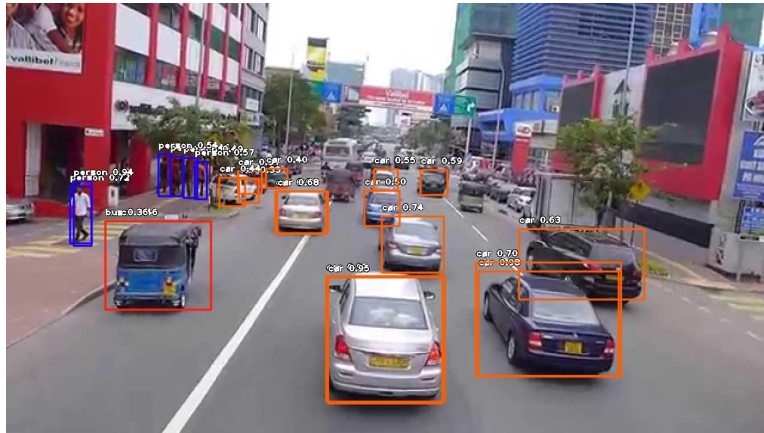


PHP (Hypertext Preprocessor) adalah bahasa pemrograman skrip yang digunakan terutama untuk pengembangan aplikasi web. PHP adalah salah satu bahasa pemrograman yang sangat populer dan banyak digunakan di seluruh dunia.



# Object Oriented Di Masa Kini

- Peningkatan Penggunaan dalam Pengembangan AI dan Machine Learning



# Object Oriented Di Masa kini

- Internet of Things (IoT)

## Analisis Perkerasan Kaku Segmen 1 - 234000 Selajambe – Cibogo – Cibeet

Aplikasi Survei Kondisi Perkerasan Jalan

File Bantuan

Video Workspace: Tabel Kerusakan Unit Sampel Aktif Ekstrak Foto

Kembali

Peta Unit Sampel -- Klik 2x Untuk Menandai/Batal Menandai Unit Sampel

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Distance: 270.63 m

Load Telenetri

Kalibrasi

Rake Video 00:00:34

Tambah Kerusakan

Unit Sampel: 6  
Jenis Kerusakan: Retak Garis  
Satuan: panel  
Tingkat Keparahan: Tinggi  
Panjang (px): 285  
Lebar (px): 258  
Estimasi Panjang (m): 2.317  
Estimasi Lebar (m): 2.633  
Estimasi Luas (m2): 6.101  
Dimensi Kerusakan: 6  
Latitude: -6.647633  
Longitude: 107.1663  
Kerapatan (%): 30.0

1 2 3 4 5 6 7  
11 12 13 14 15 16 17

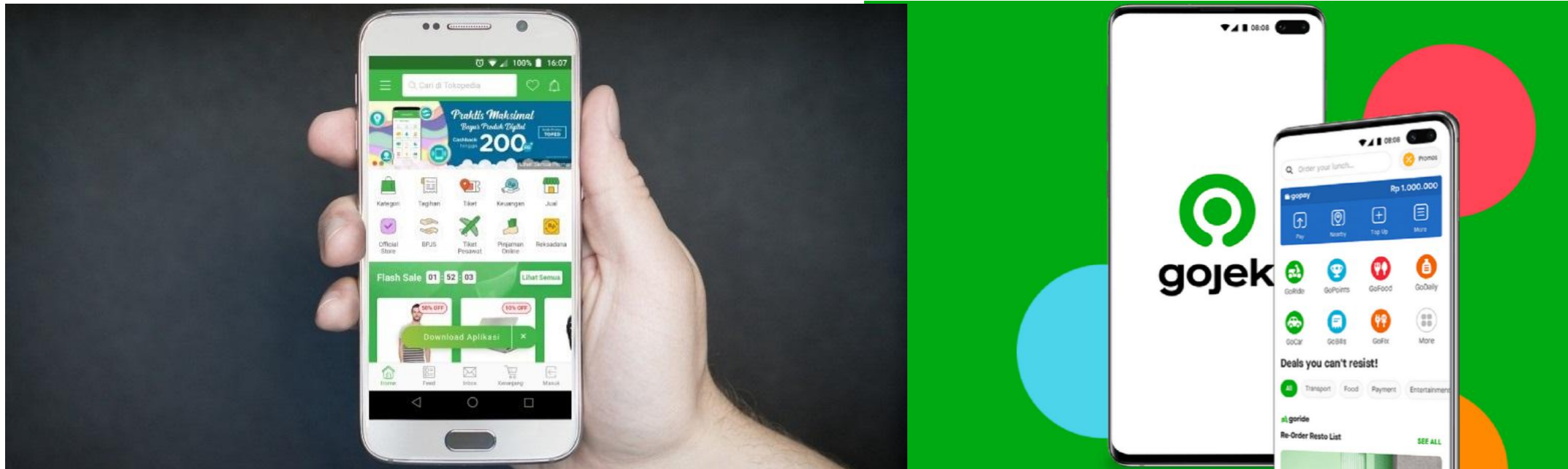
Batal Submit

Penjang Slab: 5.0 m  
Lebar Slab: 3.5 m  
Panjang Unit Sampel: 50.0 m  
Lebar Unit Sampel: 7.0 m



# Object Oriented Di Masa kini

- Microservices





- Java - The new programming language developed by Sun Microsystems in 1991.
- Java goes back to 1991, when a group of Sun engineers, led by Patrick Naughton and James Gosling , wanted to design a small computer language that could be used for consumer devices like cable TV switchboxes.
- Originally called Oak by James Gosling, one of the inventors of the Java Language.
- Java Authors: James Gosling and others

## Java White paper Buzzword

- Simple
  - Cleanup Version of C++ syntact
  - No need for header files, pointer arithmetic, structure union, operator overloading, virtual based class and soon.
- Object Oriented
  - Simply stated, object-oriented design is a programming technique that focuses on the data (= objects) and on the interfaces to that object
- Portable
  - The libraries that are a part of the system define portable interfaces

- **Distributed**

- Java has an extensive library of routines for coping with TCP/IP protocols like HTTP and FTP. Java applications can open and access objects across the Net via URLs with the same ease as when accessing a local file system.

- **Multithread**

- benefits of multithreading are better interactive responsiveness and real-time behavior.

- **Dynamic**

- In a number of ways, Java is a more dynamic language than C or C++. It was designed to adapt to an evolving environment, Libraries can freely add new methods and instance variables without any effect on their clients.

# Java Jargon

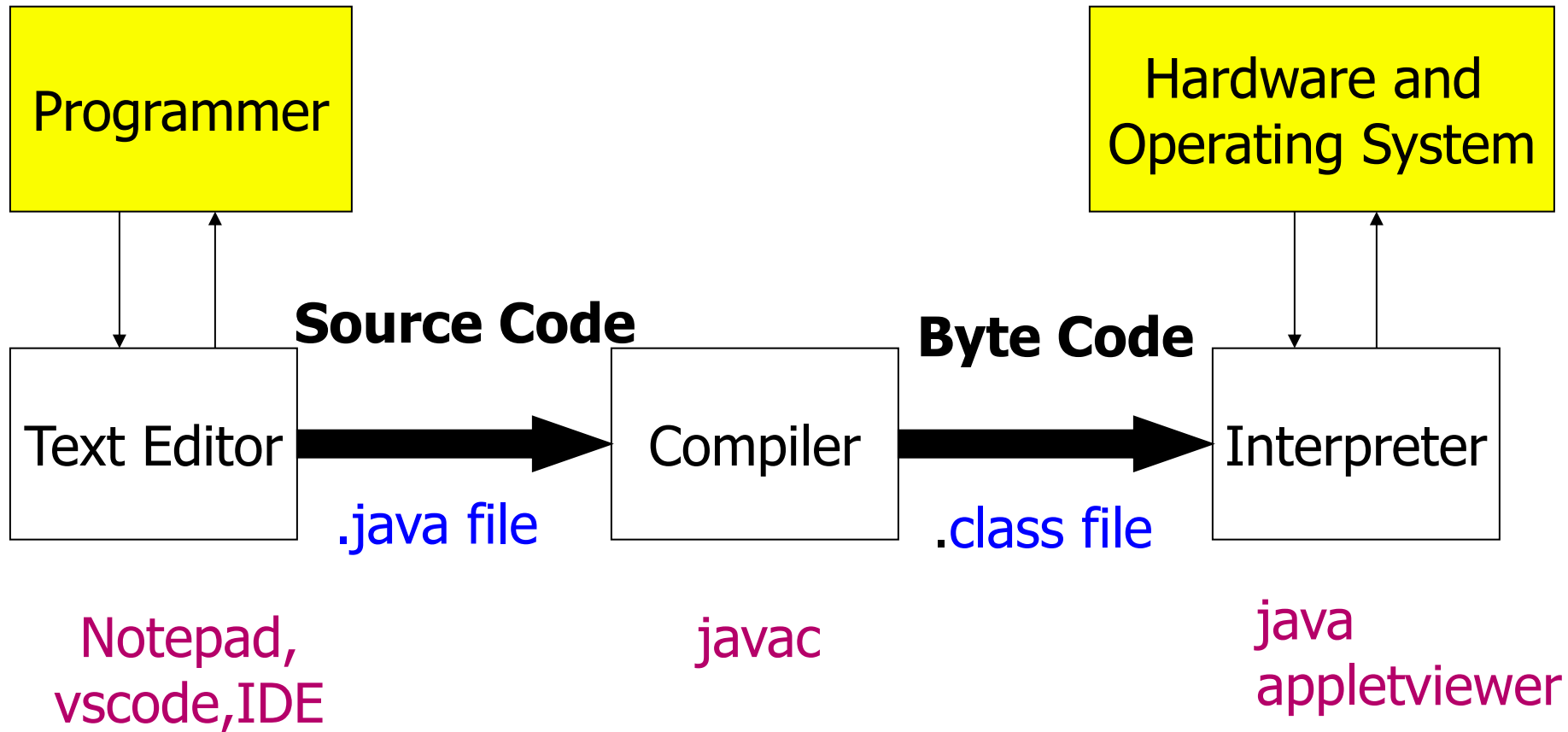
**Table 2.1** Java Jargon

Name	Acronym	Explanation
Java Development Kit	JDK	The software for programmers who want to write Java programs
Java Runtime Environment	JRE	The software for consumers who want to run Java programs
Server JRE	—	The software for running Java programs on servers
Standard Edition	SE	The Java platform for use on desktops and simple server applications
Enterprise Edition	EE	The Java platform for complex server applications
Micro Edition	ME	The Java platform for use on cell phones and other small devices
Java FX	—	An alternate toolkit for graphical user interfaces that is included in Oracle's Java SE distribution
OpenJDK	—	A free and open source implementation of Java SE. It does not include browser integration or JavaFX.
Java 2	J2	An outdated term that described Java versions from 1998 until 2006
Software Development Kit	SDK	An outdated term that described the JDK from 1998 until 2006
Update	u	Oracle's term for a bug fix release
NetBeans	—	Oracle's integrated development environment

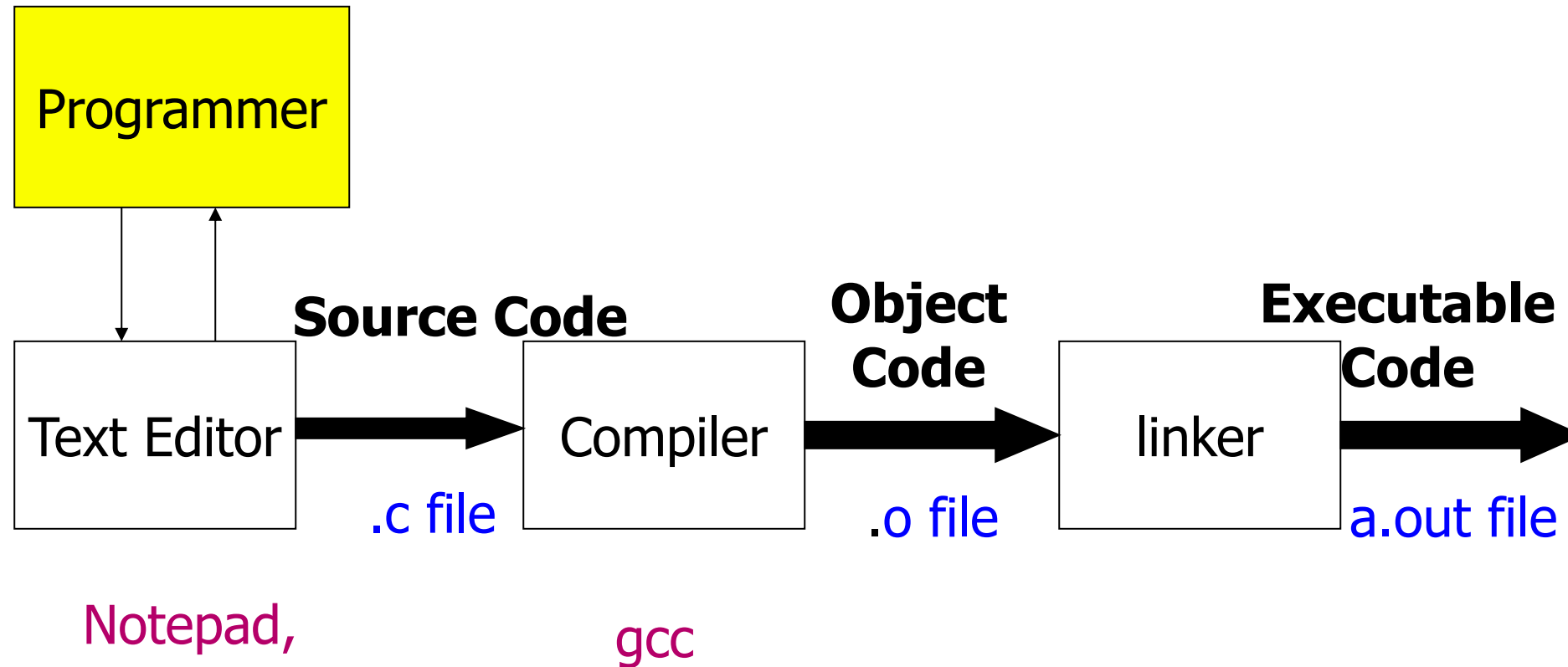
# Evolution of Java

- **Java 1.0 (1995):** Versi pertama dari Java, yang memperkenalkan bahasa pemrograman yang portabel dan platform-agnostik. Java Applets dan Java AWT (Abstract Window Toolkit) merupakan fitur-fitur awal yang memungkinkan pengembang untuk membuat aplikasi dan applet yang dapat dijalankan di berbagai platform.
- **Java 1.2 (J2SE 1.2) (1998):** Diperkenalkan konsep Swing untuk menggantikan AWT, memberikan antarmuka pengguna yang lebih baik dan lebih konsisten.
- **Java 1.5 (J2SE 5.0) (2004):** Juga dikenal sebagai Java 5 atau Java 1.5, versi ini memperkenalkan generik, enumerasi, perulangan for-each (enhanced for loop), dan anotasi.
- **Java 1.6 (Java SE 6) (2006):** Memperkenalkan dukungan untuk scripting dengan bahasa-bahasa dinamis melalui proyek "Scripting API" (javax.script package). Juga dikenal sebagai "Mustang".
- **Java 1.7 (Java SE 7) (2011):** Memperkenalkan fitur-fitur seperti switch statement dengan String, try-with-resources untuk mengelola sumber daya secara otomatis, dan penanganan eksepsi multi-catch.
- **Java 1.8 (Java SE 8) (2014):** Pembaruan yang paling mencolok adalah diperkenalkannya lambda expressions, Stream API untuk pemrosesan koleksi data, dan tanggal dan waktu yang lebih baik dengan java.time package.
- **Java 1.9 (Java SE 9) (2017):** Memperkenalkan modul sistem (Java Platform Module System), yang membantu mengatur dan mengisolasi kode, serta membuat pengembangan dan distribusi lebih efisien. Juga dikenal sebagai "Project Jigsaw".
- **Java 10 (2018):** Fitur-fitur baru termasuk tipe data var (lokal-variable type inference) dan peningkatan performa dengan mengenalkan aplikasi berat ringan (lightweight application).
- **Java 11 (2018):** Mengenalkan modul Java EE yang terintegrasi dalam platform Java SE, dan peningkatan keamanan dan performa.
- **Java 12-17 (2019-2021):** Setiap versi memiliki pembaruan dan peningkatan dalam berbagai bidang, termasuk sintaks bahasa, API baru, dan peningkatan performa.

# Java is Compiled



# Compiled Languages



# Praktikum Pertemuan 1



# Instruksi Praktikum 1

- Kerjakan 5 soal dibawah ini dengan mengikuti ketentuan sebagai berikut:
  - Isi sheet monitoring berdasarkan ketentuan yang ada di sheet tersebut.
  - Source code setiap pengerjaan soal, simpan di Github, lampirkan komentar dari hasil pengerjaan tersebut.
  - Buat laporan hasil pengerjaan berbentuk dokumen, upload laporan di folder Hasil Praktikum di folder hasil praktikum, laporan harus mencakup:
- Cover.
- Persoalan yang telah dikerjakan. Setiap persoalan, harus menjawab beberapa deskripsi berikut ini:
  - Screenshoot hasil akhir program.
  - Screenshoot setiap jawaban soal yang dipertanyakan.
  - Permasalahan yang dihadapi.
  - Solusi dari permasalahan yang dihadapi.
  - Nama teman yang membantu memecahkan permasalahan di persoalan ini.

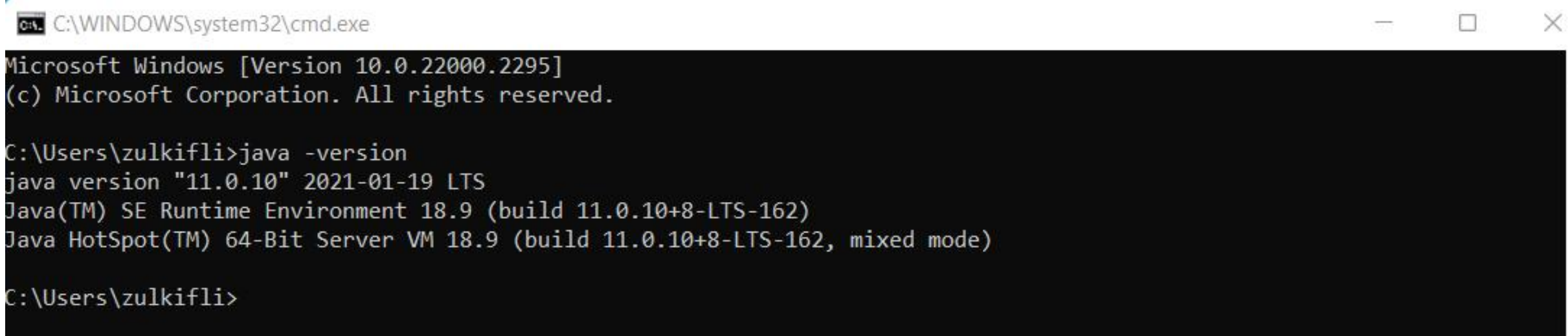
# Bobot Penilaian

NO	Pokok Bahasan	C1	C2	C3	C4	Bobot	Ket
1	Setup Software Environment			v		10	
2	Menjalankan Program menggunakan Metode Command Line			v		15	
3	Menjalankan Program menggunakan IDE			v		15	
4	Soal Analisis 1			v	v	30	
5	Soal Analisis 2			v	v	30	
Bobot						100	

Nilai 1 (Perlu perbaikan)	:	Langkah tidak dikerjakan atau tidak sesuai dengan petunjuk pada jobsheet. (64 – 74)
Nilai 2 (Mampu)	:	Langkah dikerjakan sesuai dengan petunjuk jobsheet namun dengan bantuan dari dosen pembimbing. (75-80)
Nilai 3 (Mahir)	:	Langkah dikerjakan sesuai dengan petunjuk jobsheet tanpa bantuan dari dosen pembimbing. (81 - 100)
Nilai batas lulus	:	80

# 1. Setup Software Environment

- Install Java Development Kit (JDK11)
- Setting JDK
  - Ikuti petunjuk pada 2.1.2 Setting up JDK
- Test pada command prompt *Java -version*



```
C:\WINDOWS\system32\cmd.exe

Microsoft Windows [Version 10.0.22000.2295]
(c) Microsoft Corporation. All rights reserved.

C:\Users\zul kifli>java -version
java version "11.0.10" 2021-01-19 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.10+8-LTS-162)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.10+8-LTS-162, mixed mode)

C:\Users\zul kifli>
```

## 2. Using Notepad & Command Line Tools

- Listing 2.1 Welcome.java

javac welcome.java

java welcome

### Listing 2.1 Welcome/Welcome.java

```
1  /**
2   * This program displays a greeting for the reader.
3   * @version 1.30 2014-02-27
4   * @author Cay Horstmann
5   */
6  public class Welcome
7  {
8      public static void main(String[] args)
9      {
10         String greeting = "Welcome to Core Java!";
11         System.out.println(greeting);
12         for (int i = 0; i < greeting.length(); i++)
13             System.out.print("=");
14         System.out.println();
15     }
16 }
```

### 3. Using an Integrate Development Environment

- Gunakan Netbeans/Eclipse
- Run Listing 2.1 Welcome.java

## 4. Soal analisis 1

```
public class Soal1 {  
    public static void main(String[] args) {  
        byte angka1 = 125;  
        byte angka2 = 6;  
        byte hasil = (byte) (angka1+angka2);  
  
        System.out.println("Hasil 1 "+hasil);  
    }  
}
```

- Berapa output yang keluar ? Tuliskan alasan dan referensinya

## 5. Soal Analisis 2

- Bagaimana output setelah dijalankan ?
- Tuliskan Teknik yang digunakan

```
public class Soal2 {  
    public static void main(String[] args) {  
        int i = 42;  
        String s = (i<40)?"life":(i>50)?"universe":"everything";  
        System.out.println(s);  
    }  
}
```

- Pemrograman Java case sensitive, pastikan tidak ada type atau kesalahan dalam membuat program dari kasus yang diberikan
- Pentingnya memahami terlebih dahulu range tipe data sebelum membuat suatu variable
- Ikuti step by step dengan baik agar tidak ada yang tahapan yang terlewat yang menyebabkan terjadinya issue.