

# Praktikum Pemrograman Berbasis Objek

Pertemuan 1

# Asisten Praktikum PBO 2025

Sesi 1

Kelas A

Sesi 2



M. Ainur Rafiq Noantaria  
140810230009

Bagas Diatama Wardoyo  
140810230061

Lukas Austin  
140810230011

Orlando Bloem Sutono  
140810230057

Sesi 1

Kelas B

Sesi 2



Dafa Ghani Abdul Rabbani  
140810230009

Hafizh Fadhl Muhammad  
140810230061

Hamud Abdul Aziz  
140810230009

Achmad Dzaki Azhari  
140810230061

# Rules Praktikum

1. Praktikan wajib mengikuti praktikum sesuai jadwal kelas masing-masing yaitu:

- Kelas A : Hari Senin pukul 16.00 - 18.00 WIB
- Kelas B : Hari Kamis pukul 16.00 - 18.00 WIB

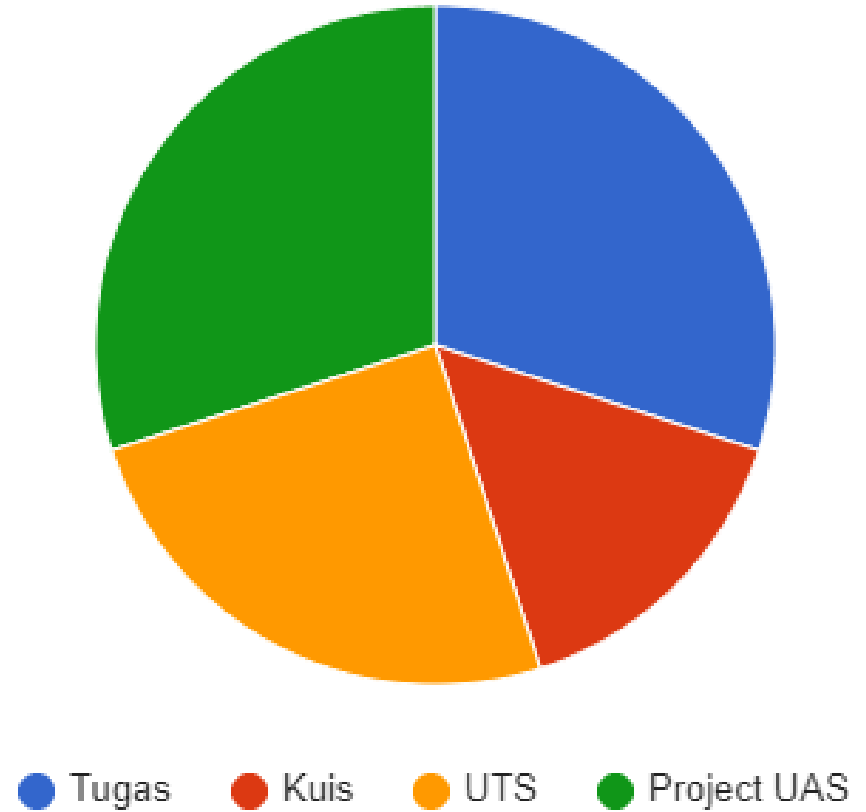
Apabila ada yang bentrok dengan jadwal yang lain harap menghubungi asprak.

2. Praktikan yang tidak mengikuti praktikum lebih dari **3** kali tidak diperkenankan untuk mengikuti UAS.
3. Platform yang digunakan untuk praktikum PBO adalah Discord dan Github Classroom.
4. Praktikan diharap dapat mengikuti praktikum dengan baik, tidak menyontek, dan menjaga kerja sama yang sehat.
5. Untuk peraturan mendatang khusus mengenai Discord dan Github Classroom dapat dipantau di channel announcement Discord Praktikum PBO 2025.

# Penilaian

Presentase nilai yang diambil adalah:

- 30% dari tugas
- 15% dari kuis
- 25% dari UTS
- 30% dari project UAS.



# Apa yang akan dipelajari?

01

## Konsep PBO

Memahami konsep PBO terutama dalam bahasa Java atau bahasa lainnya.

02

## Pemrograman dalam Java

Memahami struktur kode dan penerapan PBO dalam Java.

03

## Bahasa Pemrograman Java

Mengenal berbagai macam class dan framework dalam Java.

04

## Membuat Program Java

Menerapkan konsep PBO dengan membangun program Java.

# Materi Pertemuan 1

Gambaran Materi Pembelajaran Hari Ini

# Materi Pertemuan 1

1. Struktur Kode Java
2. Membuat Objek dari Class
3. Input & Output
4. Statements dalam Java
5. Compile & Run Kode Java
6. Built-in Methods

Lanjut →

# Struktur Code Java

Overview Mengenai Struktur Code Java



# Struktur Code Java (1)

```
public class HelloWorld {  
    public static void main(String args[]){  
        System.out.println("Hello World from Java!");  
    }  
}
```

## Class

- `public` : Access Modifier
- `class` : Class keyword
- `HelloWorld` : Name of Class

## Method

- `static` : Static keyword
- `void` : Return type
- `main` : Method
- `String args[]` : Parameter.

`String args[]` as default parameter for method  
`main`

# Struktur Code Java (2)

```
public class Ship {  
    // Variable  
    private int crew;  
  
    // Constructor: Method yang dilakukan saat instansiasi class menjadi object  
    public Ship(){  
        this.crew = 24;  
    }  
  
    // Setter method: Method untuk assign variable dalam class  
    public void setCrew(int crew){  
        this.crew = crew;  
    }  
  
    // Getter method: Method untuk mengambil nilai variable suatu class  
    public int getCrew(){  
        return this.crew;  
    }  
}
```

# Membuat Objek dari Class

Class Baru dapat Digunakan Apabila di Instansiasikan Menjadi Objek

# Membuat Object dari Class

```
public class Test {  
    public static void main(String args[]){  
        Ship ferry = new Ship();  
  
        if (ferry.getCrew() < 30){  
            ferry.setCrew(30);  
        }  
  
        System.out.println("The following ship has " + ferry.getCrew() + " crew.");  
    }  
}
```

Instansiasi class menjadi object dilakukan dengan keyword `new` diikuti dengan class yang akan dijadikan object

# Visualisasi Object dari Class

```
public class Test {  
    public static void main(String args[]){  
        Ship ferry1 = new Ship();  
        Ship ferry2 = new Ship();  
        Ship ferry3 = new Ship();  
  
        ferry1.setCrew(15);  
        ferry2.setCrew(25);  
        ferry3.setCrew(45);  
    }  
}
```



# Input & Output

Operasi Input dan Output melalui CLI dalam Java

# Input & Output

```
// import class Scanner dari package java.util
import java.util.Scanner;

public class Test {
    public static void main(String args[]){
        // Instansiasi objek scanner
        Scanner sc = new Scanner(System.in);

        //Input dengan objek scanner
        String name = sc.nextLine();
        int age = Integer.parseInt(sc.nextLine());

        // Output dengan System.out
        System.out.println("Namaku adalah " + name);
        System.out.println("Umurku " + age + " tahun.");

        // Menutup objek scanner supaya tidak terjadi memory leak
        sc.close();
    }
}
```

# Statement dalam Java

Overview mengenai If, Switch, and Looping Statements dalam Bahasa Java



# Statement dalam Java

Contoh if dan loop statement :

```
public class Loop{
    public static void main(String args[]){
        for(int i = 1; i ≤ 10; i++){
            if(i > 7){
                break;
            }
            System.out.println("Loop no. " + i);
        }
    }
}
```

Statement if, else, for, while, do-while, dan switch dalam bahasa pemrograman Java memiliki cara penulisan yang sama dengan bahasa pemrograman C++.

Berikut adalah contoh looping dan if for saat menggunakan Java, silahkan coba-coba statement lainnya.

# Compile & Run

Bagaimana Compile dan Menjalankan File Java yang Sudah Dibuat

# Compile and Run

## - Compile File Java

Untuk compile file java run command berikut di terminal:

```
javac FileName.java
```

Command diatas akan menghasilkan file baru dengan nama `FileName.java`

## - Run File Java

Untuk menjalankan file hasil compile run command berikut di terminal:

```
java FileName
```

Command diatas akan menjalankan hasil compile tanpa membawa type class-nya

# Important Things About Compile and Running Java

- Nama file `.java` harus sama dengan nama class-nya.

contoh : HelloWorld.java harus memiliki class bernama HelloWorld

- Dalam satu file `.java` dapat memiliki banyak class namun hanya diperbolehkan memiliki satu public class
- Disarankan untuk class yang bersifat public untuk memiliki filenya sendiri-sendiri

# Built-in Methods

Sudah Ada Built-in Methods dalam Java yang Dapat Digunakan Untuk Code yang Dibuat

# Built-in Methods (1)

Dalam bahasa pemrograman Java, ada banyak built-in method dari class seperti String, Integer, Character, dll. yang dapat digunakan untuk mempermudah membuat suatu program. Contohnya :

**string.isEmpty()**

↑ Mengecek apakah String kosong

Contoh penggunaan : **IF string.isEmpty() THEN ...**

**string1.equals(string2)**

↑ Membanding string dengan output true atau false

Contoh penggunaan : **IF string1.equals(string2) THEN ...**

**string1.concat(string2)**

↑ Menggabungkan dua string

Contoh penggunaan : **string3 = string1.concat(string2)**

# Built-in Methods (2)

Contoh penggunaan Built-in Methods:

```
public class Test{  
    public static void main(String args[]){  
        String word1 = "Pemrograman";  
        String word2 = "Berbasis";  
        String word3 = "Objek";  
        System.out.println(word1.concat(word2).concat(word3));  
    }  
}
```

Untuk method lainnya dapat dicek melalui google atau documentation dari oracle pada link :

<https://docs.oracle.com/en/java/javase/16/docs/api/index.html>

# Assignment!

Seperti biasa akan setiap selesai praktikum pasti ada tugas



# Snippets

Berhubung di mata kuliah PBO ini diminta untuk menguasai implementasi PBO di bahasa Java, C++ dan Python, maka kita sediakan repository snippets untuk ketiga bahasa tersebut dari setiap pertemuan praktikum.

Repository tersebut bisa diakses melalui tautan berikut :

[snippets-PBO-01](#)

Silahkan untuk kalian mencoba menjalankan dan mempelajari snippets yang disediakan pada repository tersebut. Selamat belajar!

# Assignment 1 Soal 1

soal1/Test.java

Buatlah program sederhana yang dapat menerima input berupa string sebanyak 3 kali lalu gabungkan setiap input string menjadi satu string dengan urutan dari string yang memiliki char paling sedikit.

## INPUT

string kata1, kata2, kata3

## OUTPUT

1. Jika ada 1 string atau lebih inputan yang KOSONG, Output = "Error, ada string kosong"
2. Jika string inputan TIDAK ADA YANG KOSONG, Output = gabungan dari semua string & gabungan dari semua string berhuruf kapital tanpa spasi sesuai urutan

## CONTOH

### INPUT

saya  
suka

### OUTPUT

Error, ada string kosong

### INPUT

saya  
suka  
mie

### OUTPUT

miesayasuka  
MIESAYASUKA

## ► HINT

# Assignment 1 Soal 2

soal2/Test.java

Buatlah class **Makanan** dengan variable nama, harga, dan rating serta method constructor, setter, dan getter, lalu buat file baru berisi class **Test** dengan method main yang menginstansiasi class **Makanan**.

INGAT : Gunakan constructor, setter dan getter!

CONTOH : **Makanan** food = new **Makanan**()

INPUT

nama : String  
harga : int  
rating : String

OUTPUT

[nama] dengan harga Rp [harga] sangat [rating]

CONTOH

```
input  :  
nama   = Mie  
harga  = 10000  
rating = mantap
```

```
output :  
Mie dengan harga Rp 10000 sangat mantap
```

► HINT

# Teknis Pengumpulan

Pengerjaan dan pengumpulan tugas akan dilakukan di Github Classroom

Kelas A:

Link Tugas Kelas A

Kelas B:

Link Tugas Kelas B

Accept assignment terlebih dahulu lalu link akun Github dengan slot nama yang sesuai di Github Classroom

# Teknis Pengumpulan

Format setiap file `.java` didahului dengan Nama, NPM, Kelas, Tanggal, dan Deskripsi

Cara menambah comment di java

```
// untuk single line  
/* untuk multiple line */
```

Contoh Format

```
/*  
  Nama      : Jane Doe  
  NPM       : 140810230099  
  Kelas     : A  
  Tanggal   : 26 Agustus 2025  
  Deskripsi : Class jawaban exercise-01 soal-01  
*/
```

# Deadline Pengumpulan

Kelas A:

31 Agustus 2025, 23:59 WIB

Kelas B:

03 September 2025, 23:59 WIB

Waktu yang dilihat adalah waktu last commit. Jika ada yang commit melewati deadline walaupun sudah commit sebelumnya akan dianggap telat

# Terima Kasih!

Do you have any questions? Please use respective class discussion channel on Discord.

Semangat terus menjalani kuilahnya!! 🔥 🔥 🔥