

Laporan Praktikum

MODUL 2

Pengenalan Environment OOP



**Institut Teknologi
Telkom
Purwokerto**

Nama :

Nathaya Elang Mariantaka (2211103128)

Nama Dosen :

Nicolaus Euclides Wahyu Nugroho, S.Kom., M.Cs.

**PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2023**

BAB I. TUJUAN

I. Tujuan

- a. Memahami lingkungan dasar IntelliJ IDEA serta cara meng-compile dan menjalankan program
- b. Membandingkan pemrograman terstruktur dengan pemrograman berorientasi objek dengan membuat program dalam bahasa C++ dan Java.

II. Tool

- Code Blocks
- IntelliJ 2023

III. Dasar Teori

a. Pengantar Pemrograman Berorientasi Objek

Apa itu OOP ? Merupakan teknik membuat suatu program berdasarkan objek dan apa yang bisa dilakukan objek tersebut. Object-oriented program terdiri dari objek-objek yang berinteraksi satu sama lain untuk menyelesaikan sebuah tugas.

Kenapa menggunakan OOP ? Kode-kode di-breakdown agar lebih mudah di-manage. Breakdown berdasarkan objek-objek yang ada pada program tersebut. Dianjurkan diimplementasikan untuk program dengan berbagai ukuran karena lebih mudah untuk men-debug. Pemrograman procedural mengatur program dalam barisan-barisan linier yang bekerja dari atas ke bawah. Kumpulan tahapan yang dijalankan setelah yang lain berjalan. Baik untuk program kecil yang berisi sedikit code. Tidak dianjurkan diimplementasikan pada program berukuran besar, karena susah untuk dimanage dan di-debug

OOP mempunyai karakteristik sebagai berikut :

1. Abstraksi

Menemukan hal-hal yang penting pada suatu objek dan mengabaikan hal-hal yang sifatnya insidental.

2. Enkapsulasi

Pengkapsulan adalah proses pemaketan data objek bersama method-methodnya

3. Pewarisan (Inheritance)

Proses penciptaan kelas baru (subclass/kelas turunan) dengan mewarisi karakteristik dari kelas yang sudah ada (superclass/kelas induk), ditambah karakteristik unik kelas baru itu.

4. Reuseability

Reuseability adalah kemampuan untuk menggunakan kembali kelas yang sudah ada.

5. Polymorphism

Polymorphism berasal dari bahasa Yunani yang berarti banyak bentuk.

IV. GUIDED

Membandingkan Program Terstruktur dan Program Berorientasi Objek

a. Pemrograman Terstruktur (Bahasa C++)

```
#include <iostream>

void helloWorld(int jmlh_looping) // Contoh Prosedur
{
    for (int i = 0; i < jmlh_looping; i++)
    {
        std::cout << i + 1 << ". Hello World...." << std::endl;
    }
}

int tambah(int bil1, int bil2) // Contoh Fungsi
{
    return (bil1 + bil2);
}

int main()
{
    int bil1 = 5, bil2 = 2;
    int jmlh_looping = 5;
    std::cout << std::endl;
    std::cout << "-Display Hello World-" << std::endl;
    helloWorld(jmlh_looping); // Pemanggilan Prosedur
    std::cout << std::endl << "-Pemanggilan Fungsi Tambah-" << std::endl;
    // Pemanggilan Fungsi
    std::cout << "Hasil Operasi Tambah : " << tambah(bil1, bil2) << std::endl;

    return 0;
}
```

PENJELASAN :Menampilkan Pesan "Hello World": Program mencetak pesan "Hello World...." sejumlah kali yang ditentukan oleh nilai `jmlh_looping` (5). Ini dilakukan dengan bantuan prosedur `helloWorld`.

Melakukan Penambahan Dua Angka: Program melakukan penambahan antara dua bilangan, `bil1` dan `bil2`, dan mencetak hasilnya. Ini dilakukan dengan bantuan fungsi `tambah`.

b. Pemrograman Berorientasi Objek (Bahasa C++)

```

#include <iostream>
#include <conio.h>

class DemoObjectOriented
{
public:
    void helloWorld(int jmlh_looping);
    int tambah(int bil1, int bil2);
};

void DemoObjectOriented::helloWorld(int jmlh_looping)
{
    for (int i = 0; i < jmlh_looping; i++)
    {
        std::cout << i + 1 << ". Hello World...." << std::endl;
    }
}

int DemoObjectOriented::tambah(int bil1, int bil2)
{
    return (bil1 + bil2);
}

int main()
{
    int bil1 = 5, bil2 = 2;
    int jmlh_looping = 5;
    DemoObjectOriented Demo; // Pembentukan Objek
    std::cout << std::endl;
    std::cout << "-Display Hello World-" << std::endl;
    Demo.helloWorld(jmlh_looping); // Pemanggilan Prosedur
    std::cout << std::endl << "-Pemanggilan Fungsi Tambah-" << std::endl;
    // Pemanggilan Fungsi
    std::cout << "Hasil Operasi Tambah : " << Demo.tambah(bil1, bil2) << std::endl;

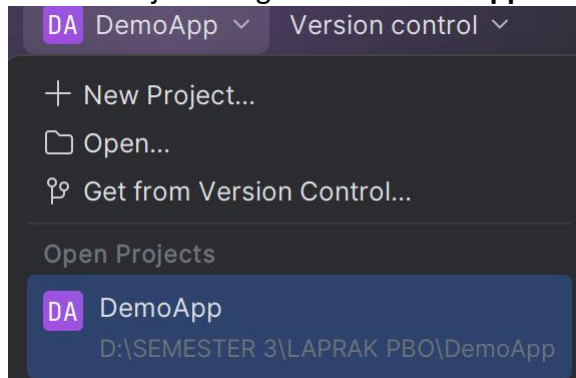
    getch();
    return 0;
}

```

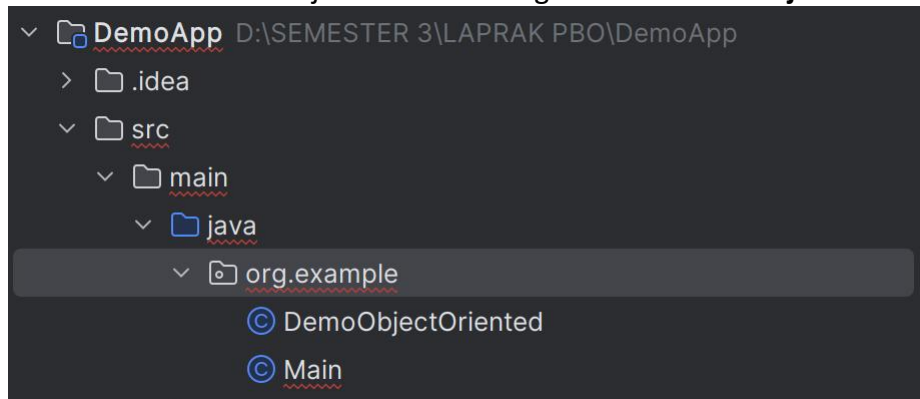
PENJELASAN : Hampir sama keluaran Hello world pada program diatas tetapi ia memanggil metode tambah(bil1, bil2); yang menghitung hasil penambahan dari bil1 dan bil2 yang memiliki nilai masing-masing 5 dan 2, sehingga hasilnya adalah 7. Hasil dari pemanggilan fungsi tersebut dicetak ke layar, yaitu : "Hasil Operasi Tambah : 7".

c. Pemrograman Berorientasi Objek (Bahasa Java)

1. Buat Project dengan nama **DemoApp**



2. Buat Class dalam Project tersebut dengan nama **DemoObjectOriented**



3. Dalam file class **DemoObjectOriented.java** tuliskan code berikut ini

DemoObjectOriented.java

```
import java.io.*;

class DemoObjectOriented {
    public void helloWorld(int jumlah_looping) {
        System.out.print("\n-Display Hello World-\n");
        for (int i = 0; i < jumlah_looping; i++) {
            System.out.print((i + 1) + ". Hello World....\n");
        }
    }

    public int tambah(int bil1, int bil2) {
        return (bil1 + bil2);
    }
}
```

Main.java

```
public class Main {  
    public static void main(String[] args) {  
        int jumlah_looping = 5;  
        int bil1 = 5;  
        int bil2 = 2;  
        DemoObjectOriented Demo = new DemoObjectOriented();  
        Demo.helloWorld(jumlah_looping);  
        System.out.print("\n-Pemanggilan Fungsi Tambah-\n");  
        System.out.print(" Hasil Operasi Tambah : " + Demo.tambah(bil1, bil2) + "\n");  
    }  
}
```

-Display Hello World-

```
1. Hello World....  
2. Hello World....  
3. Hello World....  
4. Hello World....  
5. Hello World....
```

-Pemanggilan Fungsi Tambah-
Hasil Operasi Tambah : 7

PENJELASAN : Program ini membantu memahami konsep dasar OOP dalam Java, yaitu penggunaan kelas, metode, dan objek. Selain itu, program ini juga menunjukkan cara mendeklarasikan variabel, memanggil metode, dan mencetak hasil ke layar dengan

Hasil Output:

Program mencetak pesan "Hello World...." sebanyak 5 kali sesuai dengan nilai jumlah_looping.
Program juga mencetak hasil penambahan dari bil1 dan bil2, yaitu 7.

V. GUIDED 2

1. Membuat kelas untuk menampilkan kalimat "Selamat Datang di Praktikum PBO di dalam C++ dan Java

```
#include <iostream>
```

```
class Welcome
{
public:
    void display()
    {
        std::cout << "Selamat Datang di Praktikum PBO" << std::endl;
    }
};

int main()
{
    Welcome W;
    W.display();
    // Menggunakan return 0; untuk mengakhiri program
    return 0;
}
```

BENTUK 1

Output

```
/tmp/9es5GgZjR3.o
Selamat Datang di Praktikum PBO
```

PENJELASAN : Program di atas adalah contoh program sederhana yang menunjukkan penggunaan kelas dan objek. Program ini memiliki tujuan mencetak pesan "Selamat Datang di Praktikum PBO" ke layar komputer.

```
#include <iostream>
#include <conio.h>
class Welcome
{
public:
void display();
};
void Welcome::display()
{
cout<<" Selamat Datang di Praktikum PBO "<<endl;
}
void main()
{
Welcome W;
W.display();
getch();
}
```

BENTUK 2

PENJELASAN : Akan mengeluarkan Output yang sama dengan program diatas

Hasil Output :

"Selamat Datang di Praktikum PBO" ke layar komputer.

PERBEDAAN :

1. Program Pertama (Tanpa <conio.h>)
2. Program Kedua (Dengan <conio.h>)
3. Perbedaan utama antara kedua program tersebut adalah penggunaan <conio.h> dalam program kedua. <conio.h> adalah header yang tidak termasuk dalam standar bahasa C++ dan biasanya digunakan untuk mengakses fungsi-fungsi yang berkaitan dengan kontrol tampilan dan input keyboard dalam beberapa lingkungan seperti DOS dan Windows

Welcome.java

```
public class Welcome {  
    public void display() {  
        System.out.println("Selamat Datang di Praktikum PBO");  
    }  
}
```

Main.java

```
public class Main {  
    public static void main(String[] args) {  
        Welcome welcome = new Welcome();  
        welcome.display();  
    }  
}
```

```
"C:\Program Files\Java\jdk-21\bin\java.exe"  
Selamat Datang di Praktikum PBO
```

3. Membuat Fungsi Input dan Output dalam C++

```
#include <iostream>

class Mahasiswa
{
private:
    char nama[20];
    char nim[20];

public:
    void inputData();
    void display();
};

void Mahasiswa::inputData()
{
    std::cout << "\n-Input Data Mahasiswa-" << std::endl;
    std::cout << "Masukkan Nama Mahasiswa      : ";
    std::cin >> nama;
    std::cout << "Masukkan Nomor Induk Mahasiswa   : ";
    std::cin >> nim;
}

void Mahasiswa::display()
{
    std::cout << "\n-Display Data Mahasiswa-" << std::endl;
    std::cout << "Nama : " << nama << std::endl;
    std::cout << "NIM  : " << nim << std::endl;
}

int main()
{
    Mahasiswa mhs;
    mhs.inputData();
    mhs.display();

    // Tahan layar hingga pengguna menekan Enter
    std::cin.get();

    return 0;
}
```

```
-Input Data Mahasiswa-
Masukkan Nama Mahasiswa      : Nathaya
Masukkan Nomor Induk Mahasiswa   : 2211103128
-Display Data Mahasiswa-
Nama : Nathaya
NIM  : 2211103128
```

3. Membuat Fungsi Input dan Output beserta Looping (perulangan) dalam Java**Mahasiswa.java**

```

import java.io.*; //Pemanggilan Library dalam Java
class Mahasiswa
{
    private String Nama[]= new String[3]; //Deklarasi array
    // dalam Java
    private String NIM[]= new String[3];
    private int i;
    public void inputData()
    {
        BufferedReader b;
        b=new BufferedReader(new InputStreamReader
        (System.in));
        try //Penjelasan Exception lebih lanjut di berikutnya
        {
            System.out.println("-Input Data Mahasiswa-");
            for (i=0;i<3;i++) // Perulangan (looping)
            {
                System.out.print ("Masukkan Nama : ");
                Nama[i] =b.readLine (); //Pembacaan
                inputan melalui keyboard (seperti "cin" dalam C++)
                System.out.print ("Masukkan NIM : ");
                NIM[i] =b.readLine ();
            }
        }
        catch (Exception E){}
    }
    public void display()
    {
        System.out.println("");
        System.out.println("-Display Data Mahasiswa-");
        for (i=0;i<3;i++)
        {
            System.out.println("Nama : "+Nama[i]);
            System.out.println("NIM : "+NIM[i]);
            System.out.println("");
        }
    }
}

```

Main.java

```

public static void main(String args[])
{
    Mahasiswa M=new Mahasiswa(); //Pembentukan Objek
    M.inputData(); // Pemanggilan method
    M.display ();
}

```

TIDAK BISA JALAN -> ALASAN :

PENJELASAN : Masalahnya adalah bahwa program ini menggunakan `BufferedReader` untuk membaca input dari pengguna, tetapi program ini tidak menangani pengecualian yang mungkin terjadi saat membaca input. Ini dapat menyebabkan kesalahan kompilasi. Program ini harus menangani pengecualian `IOException` yang dapat muncul saat membaca input. Program ini juga sebaiknya menambahkan perintah `throws IOException` ke metode `main` atau menangani pengecualian di dalamnya.

LINK REPOSITORY GITHUB :

<https://github.com/NathayaElang/Laprak-PBO-Modul1>