

LAPORAN PRATIKUM PEKAN 7
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN
“OBJECT ORIENTED PROGRAMMING”



Dosen Pengampu:
DR. Wahyudi. .S.T.M.T

Asisten Lab:
Jovantri Immanuel Gulo

Disusun Oleh:
Marcello Bayu Denar Widjaksono
2511532011

Fakultas Teknologi Informasi
Departemen Informatika
Universitas Andalas
2025

KATA PENGANTAR

Puji dan syukur senantiasa penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan ini guna memenuhi laporan praktikum mata kuliah Algoritma Pemrograman, dengan judul: “Object Oriented Programing”.

Pada kesempatan ini, Penulis mengucapkan terima kasih Bapak DR. Wahyudi. .S.T.M.T yang telah memberikan tugas laporan praktikum. Penulis juga ingin mengucapkan terima kasih kepada pihak-pihak yang telah mendukung serta membantu penulis dalam penyelesaian laporan praktikum Algoritma Pemrograman.

Penulis menyadari sepenuhnya bahwa laporan ini masih jauh dari sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang penulis miliki. Oleh karena itu, penulis mengharapkan segala bentuk saran dan masukan serta kritik yang membangun dari berbagai pihak.

Akhir kata, penulis berharap Laporan ini dapat memberikan manfaat bagi perkembangan dunia pendidikan.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN	3
1.1 Latar Belakang	3
1.2 Tujuan	3
1.3 Manfaat Pratikum.....	3
BAB II PEMBAHASAN	3
2.1 Teori	3
2.2 Kode Pemrograman	4
BAB III KESIMPULAN	9
3.1 Kesimpulan	9
3.2 Saran	9
DAFTAR PUSTAKA	10

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat mengembangkan sebuah program, kita kerap dihadapkan pada situasi di mana sebuah logika serupa harus diimplementasikan berulang kali dengan sedikit perbedaan dalam cara penyelesaiannya. Menulis ulang kode yang hampir sama dalam banyak bagian program tidak hanya memakan waktu, tetapi juga mengurangi efisiensi dan kemudahan pemeliharaan. Permasalahan ini dapat diatasi melalui penerapan konsep Object-Oriented Programming (OOP), yang dirancang untuk meningkatkan reuseabilitas dan modularitas kode. Meskipun konsep OOP diterapkan dalam berbagai bahasa pemrograman, masing-masing memiliki ciri khas implementasinya termasuk dalam bahasa pemrograman Java.

1.2 Tujuan

1. Menjelaskan konsep Object Oriented Programing di bahasa pemrograman Java.
2. Menjelaskan penerapan konsep Object Oriented Programing di bahasa pemrograman Java.

1.3 Manfaat Pratikum

1. Memahami konsep Object Oriented Programing di bahasa pemrograman Java.
2. Mampu menerapkan Object Oriented Programing di bahasa pemrograman Java.

BAB II PEMBAHASAN

2.1 Teori

Object Oriented Programming (OOP) atau dalam bahasa Indonesia dikenal dengan pemrograman berorientasi objek merupakan sebuah teknik pemrograman yang berorientasi Objek. Pada OOP, fungsi dan variabel dibungkus dalam sebuah objek atau class yang dapat saling berinteraksi, sehingga membentuk sebuah program.

a. Class dan Object

- Class adalah rancangan atau cetak biru dari sebuah objek. Class berisi definisi variabel (atribut / property) dan fungsi (method) yang menggambarkan sebuah objek.
- Objek adalah sebuah variabel yang merupakan instance atau perwujudan dari class .

Berikut ini adalah contoh dari class :

```
class NamaClass {  
    String atribut1;  
    String atribut2;  
    void namaMethod(){ ... }  
    void namaMethodLain(){ ... }  
}
```

Lalu kita akan membuat objek(instance) seperti dibawah ini:

```
NamaClass namaObj = new NamaClass();
```

Kata kunci new berfungsi untuk membuat objek baru dari class tertentu, setelah membuat objek seperti yang diatas, baru kita bisa mengakses atribut dan method dari objek tersebut.

Kita akan menggunakan tanda titik(.) berfungsi untuk mengakses atribut dan method yang telah kita buat sebelumnya, contohnya seperti di bawah ini:

```
namaObj.namaMethod();
```

b. This Keyword

Kata kunci This ini merujuk pada objek yang digunakan saat itu dalam suatu method atau constructor. This sering digunakan untuk menghindari kebingungan saat suatu class memiliki nama attribute yang sama dengan method atau constructor parameter. Contohnya seperti dibawah ini:

```
public class Main {
    int x; // Class variable x

    // Constructor with one parameter x
    public Main(int x) {
        this.x = x; // refers to the class variable x }

    public static void main(String[] args) {
        Main myObj = new Main(5);
        System.out.println("Value of x = " + myObj.x);
        //outputnya 5
    }
}
```

2.2 Kode Pemrograman

Disini saya akan menjelaskan beberapa program yang telah saya buat dibawah ini:

a. Program Bilangan Prima

```
public class BilanganPrima_2511532011 {
    public static boolean isPrime(int n) {
        int factor = 0;
        for (int i = 1; i <= n; i++) {
            if (n % i == 0) {
                factor++;
            }
        }
        return (factor == 2);
    }

    public static void main (String [] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("input nilai n =");
        int a = input.nextInt();
        if (isPrime(a)) {
            System.out.println(a + " " +
"bilangan prima");
        } else {
            System.out.println(a + " " + "bukan
bilangan prima");
        }
    }
}
```

Penjelasan:

1. Kita akan membuat method untuk menentukan bilangan prima berdasarkan input user(n).
2. Inisialisasi factor = 0.
3. Perulangan dimulai dengan menginisialisasi $i = 1; i \leq n; i++$
4. Jika $n \% i == 0$ maka factor ++, lalu return jika factornya == 2
5. Kemudian kita gunakan method bilangan prima tadi, jika bilangan yang diinput tadi prima maka print bil prima, jika tidak print bukan bilangan prima.

b. Program Mahasiswa

```

public class Mahasiswa_2511532011 {
    private int nim;
    private String nama, nim2;

    public void setNim (int nim) {
        this.nim = nim;
    }

    public void setNim2 (String nim2) {
        this.nim2 = nim2;
    }

    public void setNama (String nama) {
        this.nama=nama;
    }

    public int getNim() {
        return nim;
    }

    public String getNim2() {
        return nim2;
    }

    public String getNama() {
        return nama;
    }

    public void cetak() {
        System.out.println("Nim :" + nim);
        System.out.println("Nama :" + nama);
    }

    public void cetak2() {
        System.out.println("Nim :" + nim2);
        System.out.println("Nama :" + nama);
    }
}

```

Penjelasan:

1. Program ini dibuat dengan menggunakan konsep dari object oriented programming (oop).
2. Kita akan membuat class mahasiswa_nim
3. Di dalam class tersebut kita menginisialisasi atribut nim dan nama, nama2

4. Kita membuat fungsi setNim, setNim2 (berfungsi untuk menetapkan value Nim berdasarkan Nim yang diinput user) dan setName.
5. Kita membuat fungsi getNim, getNim2 dan getName (fungsi-fungsi ini berfungsi untuk mengembalikan nilai Nim dan Nama)
6. Kita juga membuat fungsi cetak1 dan cetak2 (berfungsi seperti print tapi disini kita langsung memprint nim dan nama)

c. Program PanggilMahasiswa

```
public class PanggilMahasiswa_2511532011 {
    public static void main(String [] args) {
        Mahasiswa_2511532011 a = new Mahasiswa_2511532011();
        a.setNim(23532);
        a.setName("Rahmat");
        System.out.println(a.getNim());
        System.out.println(a.getName());
        a.cetak();
    }
}
```

Penjelasan:

1. Kita membuat objek yaitu Mahasiswa_nim dengan nama objek a = new Mahasiswa_nim, untuk dapat mengakses class Mahasiswa_nim.
2. menetapkan nim dengan menggunakan a.setNim(23532).
3. menetapkan nama dengan menggunakan a.setName("Rahmat").
4. Lalu print a.getNim() (print nilai nim yang dikembalikan)
5. Lalu print a.getName()

d. Program PanggilMahasiswa2

```
public class PanggilMahasiswa2_2511532011 {
    public static void main(String [] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("NIM: ");
        String X = input.nextLine();
        System.out.print("NAMA: ");
        String Y = input.nextLine();
        Mahasiswa_2511532011 a = new Mahasiswa_2511532011();
        a.setNim2(X);
        a.setName(Y);

        if(X.startsWith("25")) {
            System.out.println(Y + " anda angkatan 2025");
        }
        if(X.contains("1153")) {
            System.out.println(" anda mahasiswa
Informatika");
        }
        a.cetak2()
    }
}
```

```

        input.close();
    }
}

```

Penjelasan:

1. Inputkan nim (value nim yang diinputkan akan disimpan ke dalam variabel x)
2. Inputkan nama (value nama yang diinputkan akan disimpan ke dalam variabel y)
3. Kita panggil objeknya
4. a.setNim(x) (menetapkan nim berdasarkan var x)
5. a.setNama(y) (menetapkan nama berdasar var y)
6. jika x dimulai dengan “25” maka print anda angkatan 2025.
7. Jika x mengandung “1153” print anda mahasiswa informatika.

e. Program String1

```

public class String1_2511532011 {
    public static void main (String [] args ) {
        String salam = "Assalamualaikum";
        System.out.println(salam.toUpperCase());
        System.out.println(salam.toLowerCase());
        System.out.println(salam.indexOf("salam")); //s dari salam
    }
}

```

Penjelasan:

1. Kita menginisialisasi var salam dengan tipe string dan valuenya = “assalamualaikum”
2. Lalu kita memprint var salam yang diubah seluruh hurufnya menjadi kapital.
3. Lalu kita memprint var salam yang diubah seluruh hurufnya menjadi kecil.
4. Print index beberapa huruf “s” dari “salam” mulai muncul.

f. Program String2

```

public class String2_2511532011 {
    public static void main ( String [] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Nama Depan: ");
        String firstName = input.nextLine();
        System.out.print("Nama Belakang: ");
        String lastName = input.nextLine();
        String txt1 = "Dosen\intelektual\kampus";

        System.out.println("Nama Lengkap: " +firstName
+ " " + lastName);
        System.out.println("Nama Lengkap: "
+firstName.concat(lastName))
    }
}

```

```

        System.out.println(txt1);

        int x = 10;
        int y = 20;
        int z = x + y ;

        System.out.println("x + y =" +z );

        String a = "10";
        String b = "20";
        String c = a + b;

        System.out.println("String a + String b = " + c
    );
        String v = a + y;
        System.out.println("String a + Integer y = " +
v );
    }
}

```

Penjelasan kode program:

1. Inisialisasi x,y,z dengan tipe int dan x=10, y=20, z=x+y.
2. Inisialisasi firstName, lastName, txt1, a,b,c,v dengan tipe string dan txt1=" Dosen\"intelektual\"kampus" a="10", b="20", c=a+b dan v = a+y.
3. Input nama depan dan nama belakang
4. Print ("nama lengkap": firstName + lastName)
5. Print txt1
6. Print("x+Y=" + z)
7. Print("sting a+string b =" +c)
8. Print v

BAB III

KESIMPULAN

3.1 Kesimpulan

Konsep Object Oriented Programming atau yang dikenal dengan OOP memudahkan kita dalam menerapkan sebuah logika (yang perlu - diimplementasikan berulang kali) dalam sebuah program sehingga menghemat waktu kita serta menjaga efisiensi dari program yang kita buat.

3.2 Saran

Berdasarkan praktikum yang telah dilakukan mengenai Konsep OOP dalam bahasa pemrograman Java. Penulis harapkan kita sebagai mahasiswa informatika dapat menguasai konsep OOP dengan mengimplementasikan nya dalam membuat sebuah program.

DAFTAR PUSTAKA

- [1] Oracle, “Lesson: Object-Oriented Programming Concepts”, The Java™ Tutorials: Learning the Java Language, Oracle Documentation, 2023 [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/java/concepts/> [Diakses: 12-November-2025].
- [2] Oracle, “Polymorphism – Java™ Tutorials”, The Java™ Tutorials: Learning the Java Language, Oracle Documentation, 2023 [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/java/IandI/polymorphism.html> [Diakses: 12-November-2025].
- [3] Oracle, “Inheritance – Learning the Java Language”, The Java™ Tutorials: Learning the Java Language, Oracle Documentation, 2023 [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html> [Diakses: 12-November-2025].
- [4] Oracle, “Lesson 8: Object-Oriented Programming”, Essentials of the Java Programming Language, Part 2, Oracle Documentation, 2023 [Daring]. Tersedia pada: <https://www.oracle.com/java/technologies/oop.html> [Diakses: 12-November-2025].
- [5] Oracle, “Trail: Learning the Java Language”, The Java™ Tutorials, Oracle Documentation, 2023 [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/java/index.html> [Diakses: 12-November-2025].

