

LAPORAN HASIL PRAKTIKUM ALGORITMA DAN PEMROGRAMAN II



NAMA : Ahmad Daffa Fahrezi
NIM : 193010503008
KELAS : A
**MODUL : I (DASAR PEMROGRAMAN
BERORIENTASI OBJEK)**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA**

2020

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



Nama : Ahmad Daffa Fahrezi
NIM : 193010503008
Kelas : A
Modul : I (DASAR PEMROGRAMAN
BERORIENTASI OBJEK)

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	8
BAB II Pembahasan	60	50
BAB III Kesimpulan	20	15
Daftar Pustaka	5	5
Lampiran	5	3.
Jumlah	100	81

Penilai

Asisten Praktikum

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

1.1. TUJUAN

Setelah menyelesaikan modul ini, mahasiswa diharapkan mampu :

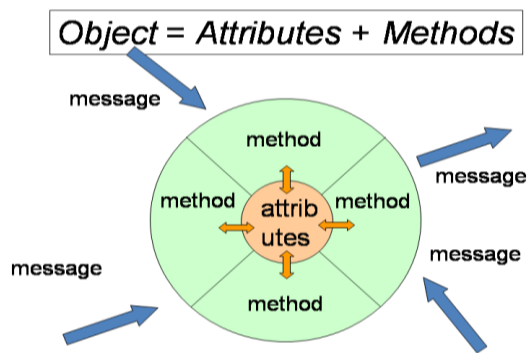
- Memahami dasar-dasar pemrograman berorientasi objek
- Memahami enkapsulasi
- Membuat kelas dan objek

1.2. DASAR TEORI

1.2.1. Perbedaan Pemrograman Tradisional dan Berorientasi Objek

Perbedaan pemrograman tradisional dan berorientasi objek adalah pada cara menyelesaikan suatu permasalahan. Pada pemrograman tradisional dalam memecahkan suatu masalah, masalah akan di bagi menjadi fungsi – fungsi yang lebih kecil, sedangkan pada pemrograman berorientasi objek (PBO) setiap masalah diselesaikan dengan cara di bagi-bagi ke dalam objek-objek.

Pada PBO di lakukan pembungkusan data (attributes) dan fungsi (behavior) ke paket yang di sebut kelas. Attributes merupakan data yang menggambarkan status internal sebuah objek dan biasanya merupakan “member variables” pada C++ , tidak dapat di akses dari luar (enkapsulasi), dan juga sebagai “state”, Methods merupakan fungsi yang mengakses status internal sebuah objek dan biasanya merupakan “ member functions” pada C++, dapat di akses dari luar, memanipulasi atribut, dan di sebut juga “behavior”. Berikut ini merupakan gambaran mengenai objek.



Kelas (class) terdiri dari model objek yang memiliki atribut (data members) dan *Behaviors (member function)*, dan *Member functions* yaitu *Methods* yang dipanggil sebagai response sebagai pesan. Kelas didefinisikan dengan *keyword class*.

Mode Akses akses yang ada pada kelas ada tiga, yaitu ***private*** yang merupakan *default* mode akses dan dapat diakses oleh *member functions*, ***public*** yang dapat diakses oleh setiap Accesible fungsi dalam program, dan ***protected*** yang biasanya digunakan untuk pewarisan.

Fungsi *Constructor* merupakan *member functions* khusus yang menginisialisasi data members dan memiliki nama yang sama dengan nama kelas. Fungsi *Constructor* dipanggil saat membuat objek dari kelas dan tidak memiliki tipe balikan.

Member functions yang didefinisikan di luar kelas dilakukan dengan menggunakan *binary scope resolution operator (::)* yang berfungsi untuk “mengikat” nama fungsi ke nama kelas dan mengidentifikasi fungsi dari suatu kelas tertentu.

Berikut ini merupakan format dari *member functions*.

```

NilaiBalikan NamaKelas::NamaFungsi( ) {
...
}

```

Members functions yang didefinisikan di dalam kelas tidak membutuhkan scope resolution operator dan nama kelas.

1.2.2. Konsep Dasar Pemrograman Berorientasi Objek (PBO)

Konsep dasar Pemrograman Berorientasi Objek (object-oriented programming) merupakan pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung PBO mengklaim bahwa PBO lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

a. Istilah-istilah dalam PBO:

1. Objek

Untuk mempermudah pemahaman, maka disini akan dijelaskan melalui analogi. Pada dasarnya semua benda yang ada di dunia nyata dapat dianggap sebagai objek. Misalnya rumah, mobil, sepeda, motor, gelas, komputer, meja, sepatu, dll. Setiap objek memiliki atribut sebagai status (*state*) dan tingkah laku sebagai *behavior*.

Contoh objek : Motor. Maka attribute (*state*) nya adalah pedal, roda, jeruji, speedometer, warna, jumlah roda. Sedangkan tingkah laku (*behavior*) nya adalah kecepatan menaik, kecepatan menurun, dan perpindahan gigi motor.

Analogi pemrograman berorientasi objek sama dengan penggambaran pada dunia nyata seperti contoh di atas. Dalam PBO, *state* disimpan pada variabel dan tingkah laku disimpan pada *method*.

Dalam bahasa teoretis PBO, Objek berfungsi untuk membungkus data dan fungsi bersama menjadi satu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

2. Class

Class yaitu template untuk membuat objek. Class merupakan prototipe atau blue prints yang mendefinisikan variabel-variabel dan method-method secara umum. Objek merupakan hasil instansiasi dari suatu class. Proses pembentukan objek dari suatu kelas disebut sebagai instantiation. Objek disebut juga sebagai instances.

Dalam bahasa teoretis PBO, class merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object.

Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan PBO). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

3. Attributes

Atribut adalah data yang membedakan antara objek satu dengan yang lainnya. Contoh Objek : VolcanoRobot (a volcanic exploration vehicle), mempunyai atribut sebagai berikut:

- Status ~> exploring, moving, returning home
- Speed ~> in miles per hour
- Temperature ~> in Fahrenheit degrees

Dalam class, atribut sering disebut sebagai variabel. Atribut dibedakan menjadi dua jenis yaitu Instance Variable dan Class Variable. Instance variable adalah atribut untuk tiap objek dari kelas yang sama. Tiap objek mempunyai dan menyimpan nilai atributnya sendiri. Jadi, tiap objek dari class yang sama boleh mempunyai nilai yang sama atau berbeda. Class Variable adalah atribut untuk

semua objek yang dibuat dari class yang sama. Semua objek mempunyai nilai atribut yang sama. Jadi semua objek dari class yang sama mempunyai hanya satu nilai yang value nya sama.

4. Behavior

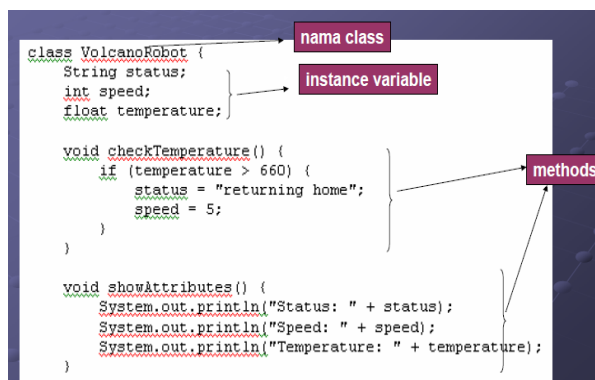
Behavior/tingkah laku adalah hal-hal yang bisa dilakukan oleh objek dari suatu class. Behavior dapat digunakan untuk mengubah nilai atribut suatu objek, menerima informasi dari objek lain, dan mengirim informasi ke objek lain untuk melakukan suatu tugas (task).

Contoh behavior untuk objek VolcanoRobot:

- check current temperature
- begin a survey
- report its current position

Dalam class, behavior disebut juga sebagai methods. Methods sendiri adalah serangkaian statements dalam suatu class yang handle suatu task tertentu. Cara objek berkomunikasi dengan objek yang lain adalah dengan menggunakan method.

- Contoh Object, class, attributes, dan behavior:



```
class VolcanoRobot {
    String status;
    int speed;
    float temperature;

    void checkTemperature() {
        if (temperature > 660) {
            status = "returning home";
            speed = 5;
        }
    }

    void showAttributes() {
        System.out.println("Status: " + status);
        System.out.println("Speed: " + speed);
        System.out.println("Temperature: " + temperature);
    }
}
```

Gambar 1.1 . . .

lelak gambar di atas

```

public static void main(String[] arguments) {
    VolcanoRobot dante = new VolcanoRobot();
    dante.status = "exploring";
    dante.speed = 2;
    dante.temperature = 510;

    dante.showAttributes();
    System.out.println("Increasing speed to 3.");
    dante.speed = 3;
    dante.showAttributes();
    System.out.println("Changing temperature to 670.");
    dante.temperature = 670;
    dante.showAttributes();
    System.out.println("Checking the temperature.");
    dante.checkTemperature();
    dante.showAttributes();
}
}

```

b. Konsep Konsep PBO:

1. Enkapsulasi (Encapsulation)

Enkapsulasi: Pembungkusan variabel dan method dalam sebuah obyek yang terlindungi serta menyediakan interface untuk mengakses variabel tersebut. Variabel dan method yang dimiliki oleh suatu objek, bisa ditentukan hak aksesnya. Dalam OOP, konsep enkapsulasi sebenarnya merupakan perluasan dari struktur dalam bahasa C.

Contoh: jam tangan. Dalam hal ini, penting sekali untuk mengetahui waktu, sedangkan cara jam mencatat waktu dengan baik antara jam bertenaga baterai atau bertenaga gerak tidaklah penting kita ketahui.

Dengan kata lain enkapsulasi berfungsi untuk memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam/dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

Dengan menggunakan PBO maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bagian administrasi untuk

mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

Salah satu pemrograman pendukung PBO adalah bahasa pemrograman Java. Java adalah salah satu bahasa pemrograman OOP. Bahasa ini awalnya dibuat oleh James Gosling. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda yang berbeda, java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

BAB II

PEMBAHASAN

2.1. Buatlah 2 program dengan tema buah bagi yang nomor ganjil dan jika nomor genap maka tema hewan dengan menggunakan prinsip dasar pemrograman berorientasi objek !

PROGRAM BUAH 1

```
#include <iostream>

#include <string>

using namespace std;

class buah{
public:
    int harga;
    string nama;
    buah(int iHarga, string iNama){
        harga=iHarga;
        nama=iNama;
    }
    void cetakData(){
        cout<<"Harga = "<<harga<<endl;
        cout<<"Nama = "<<nama<<endl;
        cout<<endl;
    }
};

int main(){
    buah mangga(10000, "Mangga sekilo");
```

```

        buah apel(20000, "Apel sekilo");

        buah jeruk(30000, "Jeruk sekilo");

        mangga.cetakData();

        apel.cetakData();

        jeruk.cetakData();

        return 0;

    }

```

PROGRAM BUAH 2

```

#include<iostream>
#include<string.h>
using namespace std;

class Buah
{   public:
    string Says() { return "?"; }
};

class Mangga: public Buah
{   public:
    string Says() { return "Mangga..."; }
};

int main()
{   Mangga* m = new Mangga();

    Buah* b = m;

    cout << "Buah Mannga : Manis " << b->Says() << endl;
}

```

```
cout << "Warna Dalam Mangga : kuning " << b->Says() << endl;  
return 0;  
}
```



2.2. Pembahasan program

```
#include<iostream>  
#include <string>  
using namespace std;
```

Pada program ini dengan menggunakan bahasa C++ yang bersifat yaitu case-sensitive, yang berarti bahwa huruf besar dan huruf kecil akan dianggap berbeda. Tidak seperti bahasa pemrograman Pascal, dimana penggunaan huruf besar dan kecil pada coding tidak akan mempengaruhi program.

Pada bagian program pertama ini digunakan dua header, yaitu `#include<iostream.h>` dan `#include <string>`

- Include merupakan salah satu jenis pengarah preprocessor yang digunakan untuk membaca file.
- `#include<iostream.h>` diperlukan pada program yang melibatkan objek `cout` dan `cin`.
- `#include <string>` diperlukan untuk memanggil nama-nama yang di deklarasikan.

```
class buah{  
    public:  
        int harga;
```

```
string nama;  
  
buah(int iHarga, string iNama){  
  
    harga=iHarga;  
  
    nama=iNama;  
  
}
```

Pada bagian program di atas, yaitu ***Class Buah {*** yang diperlukan untuk membuat objek dan mendefinisikan variabel dan method-method yang ada pada program, Tanda ‘{’ dan ‘}’ berfungsi untuk menandakan awal dan akhir sebuah definisi fungsi dan program, dan pada program ini juga menggunakan dua mode akses yaitu public. Dan pada mode akses public ini menggunakan tipe data integer, sedangkan mode akses. Tanda ‘;’ artinya menyatakan bahwa akhir dari sebuah pernyataan.

```
void cetakData(){  
  
    cout<<"Harga = "<<harga<<endl;  
  
    cout<<"Nama = "<<nama<<endl;  
  
    cout<<endl;  
  
}  
  
};
```

cout. *cout* berfungsi untuk menampilkan keluaran atau output ke layar monitor.

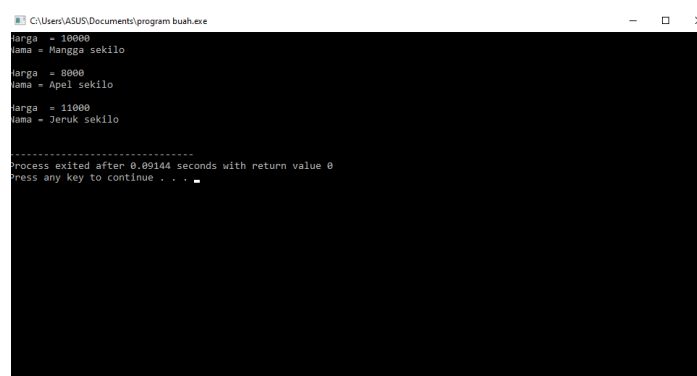
Tanda << (dua buah tanda kurang dari berurutan) merupakan sebuah operator yang disebut operator “penyisipan/peletakan”. Operator ini akan mengarahkan operand (data) yang terletak di sebelah kanannya obyek yang terletak

di sebelah kiri. *Cout* juga dapat melakukan output secara beruntun. Artinya, satu penulisan *cout* dapat diikuti oleh banyak operator. *Endl* merupakan suatu fungsi manipulator yang digunakan untuk menyisipkan karakter NewLine atau mengatur pindah baris

```
int main(){  
  
    buah mangga(10000, "Mangga sekilo");  
  
    buah apel(20000, "Apel sekilo");  
  
    buah jeruk(30000, "Jeruk sekilo");  
  
    mangga.cetakData();  
  
    apel.cetakData();  
  
    jeruk.cetakData();  
  
    return 0;  
  
}
```

➡ Pada bagian program **Int main()** merupakan bagian utama dari program. **Return** digunakan karena pada main program menggunakan tipe data integer, dan tanda '**}'**' menandakan berakhirnya program.

Berikut adalah outputnya:



```
C:\Users\ASUS\Documents\program buah.exe  
harga = 10000  
nama = Mangga sekilo  
  
harga = 20000  
nama = Apel sekilo  
  
harga = 30000  
nama = Jeruk sekilo  
  
Process exited after 0.00144 seconds with return value 0  
Press any key to continue . . .
```

↗
nomor
gambar

```
C:\Users\ASUS\Documents\mencobanya.exe
Buah Mangga : Manis ?
Warna Dalam Mangga : kuning ?
-----
Process exited after 0.06498 seconds with return value 0
Press any key to continue . . .
```

Penjelasan hasil

OUTPUT

Kurang
kelengkapan



BAB III

KESIMPULAN

Dapat di tarik kesimpulan sebagai berikut, Pemrograman Berorientasi Objek (PBO) merupakan pemrograman yang berorientasikan terhadap objek, dimana semua data dan fungsi yang dibungkus ke dalam kelas-kelas atau objek-objek. dan enkapsulasi adalah Pembungkusan variabel dan method dalam sebuah obyek yang terlindungi serta menyediakan interface untuk mengakses variabel tersebut. Variabel dan method yang dimiliki oleh suatu objek, bisa ditentukan hak aksesnya.

Objek berfungsi untuk membungkus data dan fungsi bersama menjadi satu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

DAFTAR PUSTAKA

1 enter

- Tim Dosen Algoritma Pemrograman II. 2020. *Modul Praktikum Algoritma Pemrograman II*. Universitas Palangka Raya : UPR. Fakultas Teknik.
- Auliarosalinda. 2013. *Konsep Dasar Pemrograman Berorientasi*.
<http://blogspot.com/3013/03/konsep-dasar-pemrograman-berorientasi.html>
Diakses 05 April 2020 jam 19.00 WIB.
- Adityarizki. 2012. *Konsep Dasar Pemrograman Berorientasi Objek*.
<http://www.net/2012/6/konsep-dasar-pemrograman-berorientasi-objek/>
Diakses 05 April 2020 jam 20.30 WIB.

LAMPIRAN

```
C:\Users\ASUS\Documents\program buah.exe
Harga = 10000
Nama = Mangga sekilo

Harga = 8000
Nama = Apel sekilo

Harga = 11000
Nama = Jeruk sekilo

-----
Process exited after 0.09144 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\ASUS\Documents\mencobanya.exe
Buah Mangga : Manis ?
Warna Dalam Mangga : kuning ?

-----
Process exited after 0.06498 seconds with return value 0
Press any key to continue . . .
```

nomor gambar