

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



NAMA : JERI
NIM : 193030503055
KELAS : A
MODUL : I (DASAR PBO)

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2020**

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II



Nama : JERI
NIM : 193030503055
Kelas : A
Modul : DASAR PBO

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	8
BAB II Pembahasan	60	50
BAB III Kesimpulan	20	10
Daftar Pustaka	5	5
Lampiran	5	5
Jumlah	100	78

Penilai
Asisten Praktikum

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

1.1. TUJUAN

Setelah menyelesaikan modul ini, mahasiswa diharapkan mampu :

1. Memahami dasar-dasar pemrograman berorientasi obyek.
2. Memahami enkapsulasi.
3. Membuat kelas dan objek.

1.2. LANDASAN TEORI

Pemrograman berorientasi objek (object-oriented programming disingkat OOP) 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 OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Istilah-istilah dalam OOP:

1. Kelas atau Class

Merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. 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.

2. Objek

Membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer, objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

3. Abstraksi

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan.

4.Enkapsulasi

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.

5.Polimorfisme

Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim.

Identifier adalah suatu pengenalan atau pengidentifikasi yang kita deklarasikan agar kompiler dapat mengenalinya. Identifier sendiri dapat berupa nama variabel, konstanta fungsi, kelas, template maupun namespace. Biasanya identifier ini terdiri dari 3 bentuk, yaitu privat public dan protected

1. public: anggota pribadi dari kelas dasar hanya dapat diakses oleh anggota yang kelas dasar.
2. private: anggota publik dari kelas dasar dapat diakses oleh anggota yang kelas dasar, anggota kelas turunan serta anggota yang berada di luar kelas dasar dan kelas turunan.
3. protected: anggota dilindungi dari kelas dasar dapat diakses oleh anggota kelas dasar serta anggota kelas turunan.

BAB II

PEMBAHASAN

2.1. LANGKAH KERJA

Buatlah 2 program dengan tema buah bagi yang nomor ganjil dan jika nomor genap maka tema hewan dengan menggunakan prinsip DASAR PEMROGRAMAN BERORIENTASI OBJEK.

2.2. PEMBAHASAN

2.2.1. Program Hewan Pertama

Input Program

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

using namespace std;

class hewan
{
private:
    string nama;
    string habitat;
    string ciri_khusus;

public:
    infoHewan();
    infoGajah();
    hewan(string, string, string);

};

hewan::infoHewan(){

    cout<<endl;
    cout<<"=====Modul I===== "<<endl;
    cout<<"Info salah satu jenis hewan"<<endl;
```

```
};
```

```
hewan::hewan(string nm, string hbt, string cr_k)  
{
```

```
    nama=nm;
```

```
    habitat=hbt;
```

```
    ciri_khusus=cr_k;
```

```
}
```

```
hewan::infoGajah()
```

```
{
```

```
    cout<<"Nama      :" <<nama<<endl;
```

```
    cout<<"Habitat   :" <<habitat<<endl;
```

```
    cout<<"Ciri Khusus :" <<ciri_khusus<<endl;
```

```
}
```

```
int main()
```

```
{
```

```
    string nm="Gajah";
```

```
    string hbt="padang rumput";
```

```
    string cr_k="Belalai dan Gading";
```

```
    hewan gjh(nm, hbt, cr_k);
```

```
    gjh.infoHewan();
```

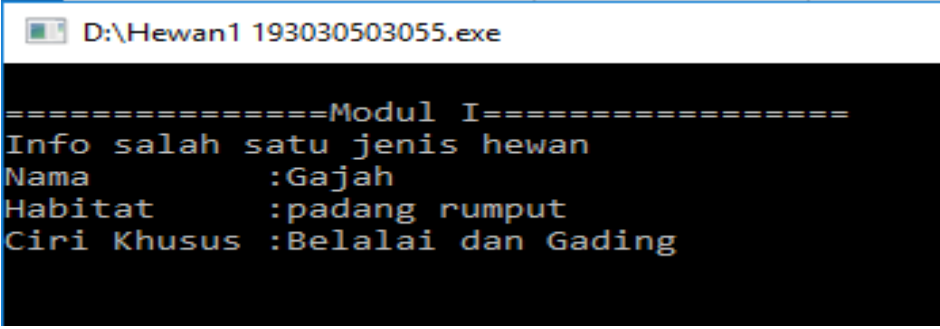
```
    gjh.infoGajah();
```

```
    _getche();
```

```
    return 0;
```

```
}
```

Output Program



```
D:\Hewan1 193030503055.exe

====Modul I====
Info salah satu jenis hewan
Nama      :Gajah
Habitat   :padang rumput
Ciri Khusus :Belalai dan Gading
```

Gambar 2.1. Program hewan 1

Penjelasan :

Dalam program ini saya menggunakan class, dimana kelas yang saya gunakan adalah class hewan. Didalam class terdapat private dan public. Private yang saya gunakan adalah string dan public menggunakan perintah info.

Fungsi pada `cout<<` adalah untuk menampilkan teks kelayar dimana teks yang akan ditampilkan adalah `<<"====Modul I===="` dan diakhiri dengan perintah `endl` yang memiliki fungsi untuk mengganti baris. Dan begitu juga yang lain.

Dalam program utamanya pemanggilan kelas digunakan dan `cout <<` akan mencetak nama, habitat, dan ciri khusus dari hewan dan dilanjutkan dengan perintah `int main` dimana `cout` yang dicetak tadi akan menampilkan info hewan seperti nama : gajah dan begitu juga yang lain sesuai info yang dimasukan.

Perintah `getche` berfungsi untuk membaca karakter dengan menampilkan karakter yang di inputkan ke layar. Dan `return 0` setiap fungsi harus memiliki nilai balik.

2.2.2 Program hewan kedua

Input Program

```
#include<iostream>
using namespace std;
class anjing
{
public:
```

```

anjing(int);
int jlhTelinga();
int jlhkaki();

private:
int telinga;
int kaki;
};

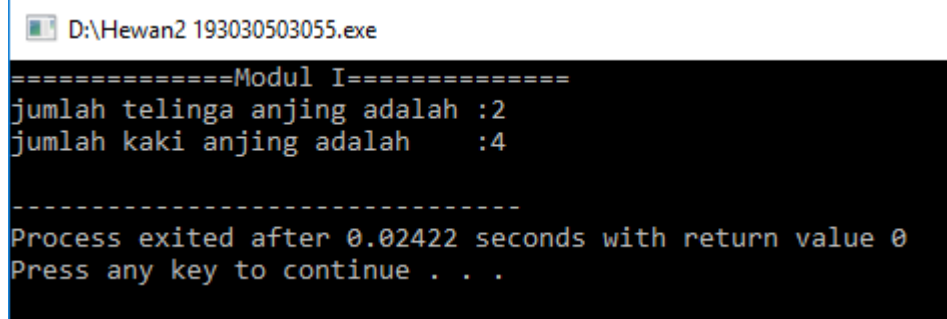
anjing::anjing(int t)
{
telinga=t;
kaki=t;
}
int anjing::jlhTelinga()
{
return telinga;
}
int anjing::jlhkaki()
{
return kaki;
}
int main()
{

anjing t(2);
anjing k(4);
cout<<"=====Modul I===== "<<endl;
cout<<"jumlah telinga anjing adalah : " <<t.jlhTelinga()<<endl;
cout<<"jumlah kaki anjing adalah  : " <<k.jlhkaki()<<endl;

return 0;
}

```

Output Program



```
D:\Hewan2 193030503055.exe
====Modul I====
jumlah telinga anjing adalah :2
jumlah kaki anjing adalah :4
-----
Process exited after 0.02422 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.2. program hewan 2

Penjelasan :

Code diatas adalah termasuk kelas hewan yang bernama Anjing. Didalam publik berisikan method atau function yang akan mengakses kelas hewan Anjing tersebut. Sedangkan, private bertipe data int.

Constuktornya adalah anjing::anjing dan code method dari kelas hewan Anjing yang bisa mengakses member function dengan tanda binnary scope resoution operator (::).

Dalam Program utamanya cout akan mencetak jumlah telinga dan kaki anjing adalah 2 dan 4 karena int anjing:: jlh telinga yang dimasukan adalah 2 dan int anjing:: jlh kaki yang dimasukan adalah 4.

BAB III

KESIMPULAN

No bullet

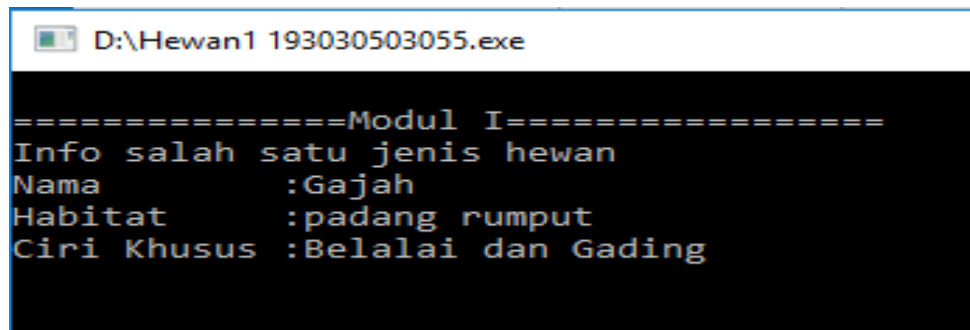
Pemrograman berorientasi objek (object-oriented programming disingkat OOP) merupakan pemograman yang berorientasikan kepada objek.

- Semua data dan fungsi di dalam paradigma dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.
- Class atau kelas merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu.
- Namun intinya: kelas adalah representasi benda(object) dari dunia nyata.

DAFTAR PUSTAKA

- Diana. Asisten Praktikum Teknik Informatika. Algoritma dan Pemrograman II. 2020. *Cover laporan hasil dan Laporan hasil praktikum Algoritma dan pemrograman II*. Universitas Palangka Raya. Fakultas Teknik. Jurusan Teknik Informatika.
- Surakarta. 2015. *189-konsep-dasar-pemrograman-berorientasi-objek* .
<http://technopark.surakarta.go.id/id/media-publik/komputer-teknologi-informasi/189-konsep-dasar-pemrograman-berorientasi-objek> (Diakses pada tanggal 6 April 2020 pukul 20:00 WIB).

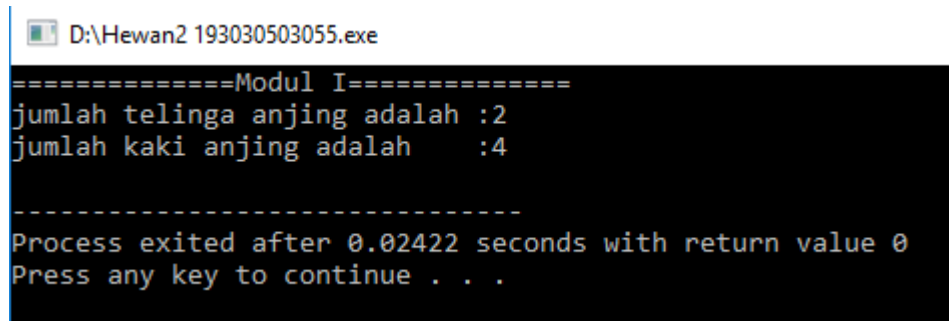
LAMPIRAN



```
D:\Hewan1 193030503055.exe

====Modul I====
Info salah satu jenis hewan
Nama      :Gajah
Habitat   :padang rumput
Ciri Khusus :Belalai dan Gading
```

Gambar 2.1. Program hewan 1



```
D:\Hewan2 193030503055.exe

====Modul I====
jumlah telinga anjing adalah :2
jumlah kaki anjing adalah   :4

-----
Process exited after 0.02422 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.2. program hewan 2