

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : BENIE AGUSTI KAHARAP**  
**NIM : 193030503076**  
**KELAS : (A)**  
**MODUL : II PEWARISAN**

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

**2020**

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



Nama : Benie Agusti Kaharap  
NIM : 193030503076  
Kelas : (A)  
Modul : II Pewarisan

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

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

Dalam PBO, kita mengambil realita kehidupan sehari-hari. Kita melakukan pengamatan bahwa manusia secara alami sering melakukan pengelompokan atas objek atau benda. Sejauh ini kita mengetahui cara untuk melakukan pengelompokan-pengelompokan atas objek-objek yang serupa (menjadi kelas objek).

Pewarisan ada dua jenis yaitu pewarisan tunggal dan pewarisan jamak. Pada *protected access*, **protected** members dapat diakses oleh member kelas dasar, friend.

kelas dasar, member kelas turunan, dan friend kelas turunan. Kelas turunan dapat.

merujuk/mengakses langsung **public** dan **protected** data member kelas induk dengan.

menggunakan nama atribut yang yang diakses

## BAB II

### PEMBAHASAN

#### 2.1 Program pertama

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

class hewan{
protected:
int berat , tinggi;
public:
void set_values(int a, int b){
berat = a;
tinggi = b;
```

```

}
};

class Tampilkan{
public:
void Say(string i){
cout << "" << i << "" << endl;
}
};

class kucing: public hewan, public Tampilkan{
public:
int tinggi(){
return (berat);
}
};

class Anakkucing: public hewan, public Tampilkan{
public:
int berat(){
return (tinggi);
}
};

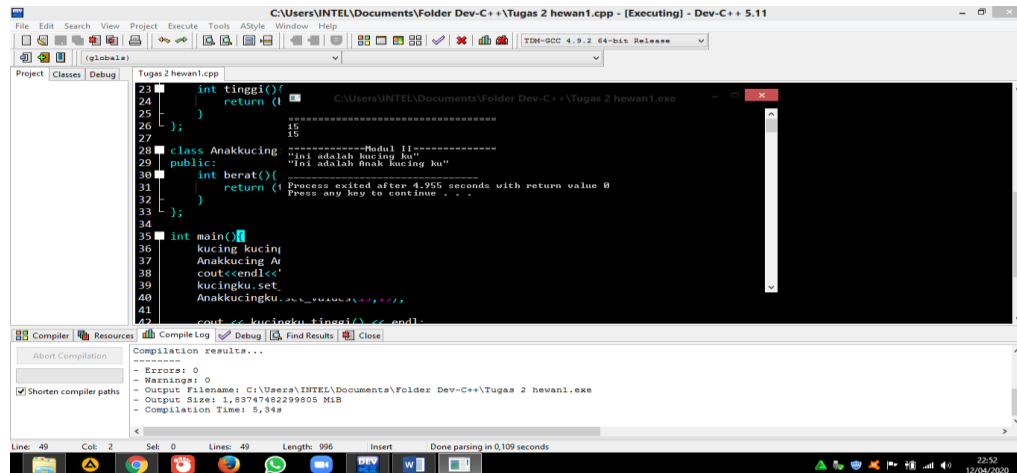
int main(){
kucing kucingku;
Anakkucing Anakkucingku;
cout<<endl<<"===== "<<endl;
kucingku.set_values(15,15);
Anakkucingku.set_values(15,15);

cout << kucingku.tinggi() << endl;
cout << Anakkucingku.berat() << endl;
cout<<endl<<"=====Modul II===== "<<endl;
kucingku.Say("ini adalah kucing ku");
Anakkucingku.Say("Ini adalah Anak kucing ku");

return 0;
}

```

Hasil output : Program 1



```
23 int tinggi(){
24     return t;
25 }
26
27
28 class Anakkucing
29 public:
30     int berat(){
31         return (
32     );
33 };
34
35 int main()
36 {
37     kucing kucing;
38     Anakkucing An;
39     kucingku.set
40     Anakkucingku.
41     cout << kucingku.tinggi() << endl;
42 }
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\INTEL\Documents\Folder Dev-C++\Tugas 2 hewan1.exe
- Output Size: 1,8374782299805 MB
- Compilation Time: 0,349

## 2.2 Program kedua

```
#include<iostream>
using namespace std;
class anjing
{
    public:
        anjing(int);
        int jlhTelinga();
        int jlhkaki();

    protected:
        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 II===== "<<endl;
    cout<<"jumlah telinga anjing adalah : " <<t.jlhTelinga()<<endl;
    cout<<"jumlah kaki anjing adalah   : " <<k.jlhkaki()<<endl;

    return 0;
}

```

Hasil output : Program 2

```

1 //=====Modul II=====
2 using namespace std;
3 class anjing
4 {
5     public:
6         anjing(int);
7         int jlhTelinga();
8         int jlhkaki();
9
10    protected:
11        int telinga;
12        int kaki;
13};
14
15 anjing::anjing(int t)
16 {
17     telinga=t;
18     kaki=t;
19 }
20 int anjing::jlhTelinga()

```

Output:

```

=====Modul II=====
jumlah telinga anjing adalah :2
jumlah kaki anjing adalah :4
Process exited after 10.47 seconds with return value 0
Press any key to continue . . .

```

PENJELASAN :

**Pada program 1** saya menggunakan namaHewan kucing yang menghitung berat dan tinggi. Disini saya menggunakan untuk menampilkan **Public class** kucing yang di **Private**.

**Pada program 2** saya menggunakan namaHewan anjing yang mana tidak jauh berbeda dari Program 1 disini saya menggunakan **Protected** untuk menampilkan **Public class** anjing yang di **Private** untuk hasil output.

## **BAB III**

### **KESIMPULAN**

1. Pemrograman berorientasi objek (object-oriented programming disingkat OOP) merupakan pemograman yang berorientasikan kepada objek.
2. 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.
3. Class atau kelas merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu.
4. Fungsi objek adalah untuk 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.

### **DAFTAR PUSTAKA**

<https://www.duniailkom.com/tutorial-belajar-oop-php-pengertian-inheritance-pewarisan/>

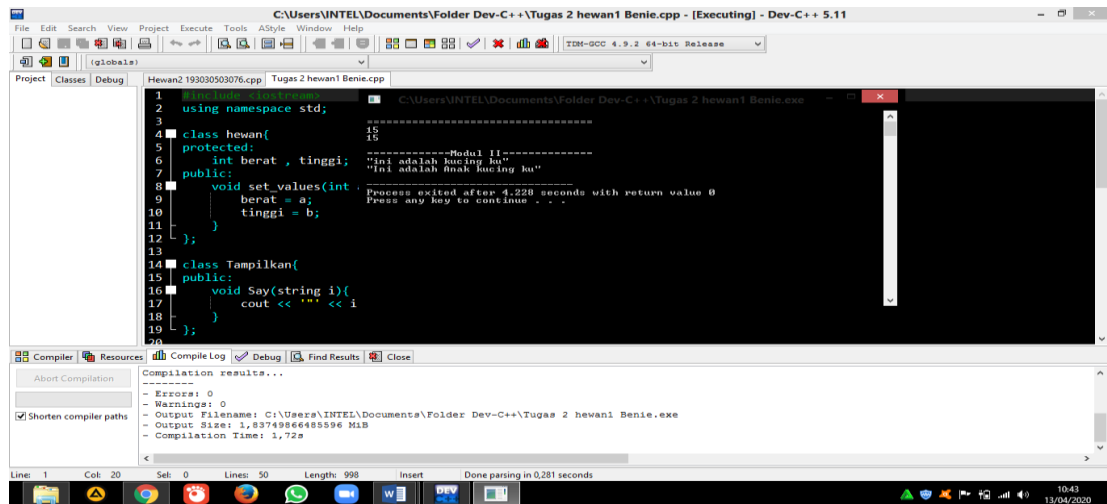
(Diakses sabtu 11 April 2020 Jam 08.55)

<https://docplayer.info/45953393-Modul-3-pewarisan-tujuan-dasar-teori.html>

(Diakses sabtu 11 April 2020 Jam 10.23)

## Lampiran :

Gambar 1.1 Program 1



Gambar 1.2 Program 2

