

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



Nama : OKKI ANDARESTA
NIM : DBC 118 068
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 : Okki Andaresta
NIM : DBC 118 068
Kelas : A
Modul : DASAR PEMROGRAMAN
BERORIENTASI OBJEK

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	5
BAB II Pembahasan	60	45
BAB III Kesimpulan	20	10
Daftar Pustaka	5	5
Lampiran	5	5
Jumlah	100	70

Penilai

Asisten Praktikum

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

1.1. TUJUAN

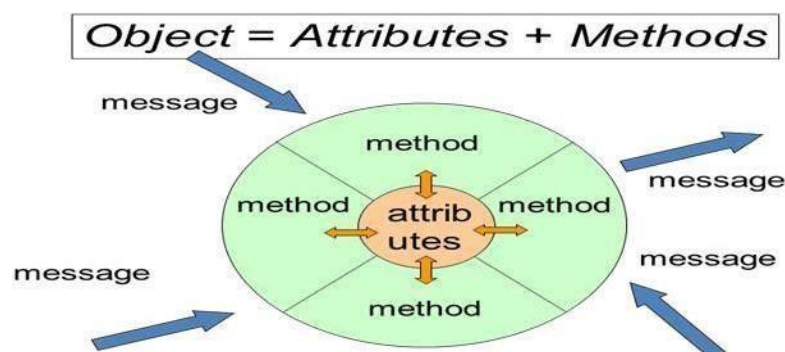
Setelah menyelesaikan modul ini, mahasiswa diharapkan mampu :

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

1.2. LANDASAN TEORI

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

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



Landasan
teori
ditambah

Kelas (Class) terdiri dari model objek yang memiliki atribut (data members) dan *Behaviors (member functions)*, dan *Member functions* yaitu *Methods* yang dipanggil sebagai response terhadap 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 Accessible fungsi dalam program, dan ***protected*** yang biasanya digunakan untuk pewarisan .

Fungsi *Constructor* merupakan *member function* 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( ){  
    ...  
}
```

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

BAB II

PEMBAHASAN

I 1 enter

2.1. Program hewan pertama

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

class domba{
    private:
        int spesies;
        int kromosom;

    public:
        domba (int);
        int jlhSpesies();
        int jlhKromosom();

};

domba :: domba(int t){
    spesies = t;
    kromosom = t;
}
```

```
domba m(7);

domba h (54);

cout<<"jumlah spesies domba setidaknya adalah : "<<m.jlhSpesies()<<endl;

cout<<"jumlah haploid kromosom domba : "<<h.jlhKromosom()<<endl;

return 0;

}
```

Penjelasan :

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

class domba{
    private:
        int spesies;
        int kromosom;

    public:
        domba (int);
        int jlhSpesies();
        int jlhKromosom();

};
```

Pada Code diatas termasuk kelas yang bernama kelas domba yang dimana memiliki jenis private dan publik. Di public terdapat methods dan terdapat contruktor yang memiliki nama yang sama dengan class domba.

```

domba :: domba(int t){
    spesies = t;
    kromosom = t;
}

int domba :: jlhSpesies(){
    return spesies;
}

int domba :: jlhKromosom(){
    return kromosom;
}

```

fungsi tersebut adalah member function yang dapat mengakses dapat diakses diluar. seperti domba dengan tipe balikan jlhSpesies dan domba jlhKromosom.

```

int main (){
    domba m(7);

    domba h (54);

    cout<<"jumlah spesies domba setidaknya adalah : "<<m.jlhSpesies()<<endl;

    cout<<"jumlah haploid kromosom domba : "<<h.jlhKromosom()<<endl;

    return 0;
}

```

Dalam program int main (){ ini terdapat variabel m bernilai 7 dan h bernilai 54, cout berguna untuk mencetak kata dalam layar dengan masing-masing variabel m untuk jlh.Spesies dan variabel h untuk jlhKromosom dan return 0 berfungsi untuk mengembalikan nilai ke sisitem operasi.

Output program :

```
jumlah spesies domba setidaknya adalah : 7
jumlah haploid kromosom domba : 54

-----
Process exited after 0.2274 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.1 program pertama

2.2 Program hewan kedua

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

class hewan{
private:
    string klasifikasi;
    string filum ;
    string kelas;

public:
    hewan(string klasifikasi, string filum, string kelas){
        hewan::klasifikasi = klasifikasi;
        hewan::filum = filum;
        hewan::kelas = kelas;
    }

    void perolehInfo(void){
        cout << "Klasifikasi : " << klasifikasi << endl;
        cout << "Filum      : " << filum      << endl;
        cout << "Kelas      : " << kelas      << endl;
        cout << endl;
    }

};

int main()
{
    hewan a("Vertebrata","Chordata","Mammalia");
    hewan b("Invertebrata","Arthropoda","insecta" );
    a.olehInfo();
    b.olehInfo();
    return 0;
}
```


Penjelasan :

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

class hewan{
private:
    string klasifikasi;
    string filum ;
    string kelas;

public:
    hewan(string klasifikasi, string filum, string kelas){
        hewan::klasifikasi = klasifikasi;
        hewan::filum = filum;
        hewan::kelas = kelas;
    }
}
```

Pada program hewan kedua ini terdapat badan **Class** program tersebut, dengan **Private** yang dapat diakses oleh program secara khusus(privat) adalah *string klasifikasi; string filum; string kelas;* yang dapat diakses secara khusus saja. Kemudian dalam badan program terdapat program **public** disinilah badan private akan diakses, akan mengisi data yang dimasukan dalam program ke tempatnya, seperti *hewanl::klasifikasi = klasifikasi;* yang mengartikan bahwa isi dari *klasifikasi* akan disisi data dari *klasifikasi* begitu juga seterusnya pada *filum* dan *kelas*. Dan bentuk *hewan(string klasifikasi, string filum, string kelas)* berguna untuk menjadikan parameter dari yang diambil dari **private**.

```

void perolehInfo(void){
    cout << "Klasifikasi : " << klasifikasi << endl;
    cout << "Filum      : " << filum      << endl;
    cout << "Kelas     : " << kelas     << endl;
    cout << endl;
}

};

int main()
{
    hewan a("Vertebrata","Chordata","Mammalia");
    hewan b("Invertebrata","Arthropoda","insecta" );
    a. perolehInfo();
    b. perolehInfo();
    return 0;
}

```

Kemudian Disana terdapat *void perolehInfo* yang menjadi konstruktornya. Dalam Badan *Class* tersebut akan menampilkan outputnya. Dalam badan program *main()* terdapat pemanggilan *hewan a* kemudian diikuti dengan nama variabel dan isi parameternya. pemanggilan tersebut bermaksud untuk menempatkan parameter tersebut kedalam masing-masing tempatnya, misalkan *hewan a*, disana parameter pertama adalah "vertebrata", dimana "vertebrata" ini akan masuk kedalam *string* dan *chordata* begitu pula urutan selanjutnya. Kemudian setelah semua isi telah masuk dalam parameternya, sekarang tinggal konstruktornya, dalam *Class hewan a dan hewan b* tadi terdapat void untuk memberi bentukan yang dikeluarkan, maka jika hewan a ingin dioutputkan, maka perlu memanggil *void perolehinfo()* dengan bentuk pemanggilan *a. perolehInfo()*; dari sana akan diproses bentuk keluarannya. Begitu juga pada pemanggilan hewan b.

Output kedua :

BAB III

KESIMPULAN

Kelas dan objek merupakan dua hal yang berbeda, jika kelas merupakan kumpulan suatu fungsi-fungsi dalam suatu unit yang terdapat atribut dan method, sedangkan objek merupakan bentuk arsitektur dari suatu kelas membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer.

DAFTAR PUSTAKA

- Dosen Teknik Informatika. Algoritma dan Pemrograman I. 2020.
Modul Praktikum Algoritma dan Pemrograman II. Universitas
Palangka Raya. Fakultas Teknik. Jurusan Teknik Informatika.
- Andri Kristanto. 2009. *Algoritma & Pemrograman dengan C++*.
Yogyakarta. Graha Ilmu.
- Budi Raharjo. 2010. *Pemrograman C++*. Bandung. Informatika.

LAMPIRAN

```
jumlah spesies domba setidaknya adalah : ?  
jumlah haploid kromosom domba : 54  
-----  
Process exited after 0.2274 seconds with return value 0  
Press any key to continue . . .
```

Gambar 2.1 program pertama

```
Klasifikasi : Vertebrata  
Filum       : Chordata  
Kelas      : Mammalia  
  
Klasifikasi : Invertebrata  
Filum       : Arthropoda  
Kelas      : insecta  
-----  
Process exited after 0.323 seconds with return value 0  
Press any key to continue . . .
```

Gambar2.2 Program Kedua

