

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



NAMA : JERI
NIM : 193030503055
KELAS : A
MODUL : III (POLIMERFISME)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2020

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II



Nama : JERI
NIM : 193030503055
Kelas : A
Modul : POLIMERFISME

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	8
BAB II Pembahasan	60	48
BAB III Kesimpulan	20	13
Daftar Pustaka	5	5
Lampiran	5	5
Jumlah	100	

Penilai
Asisten Praktikum

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

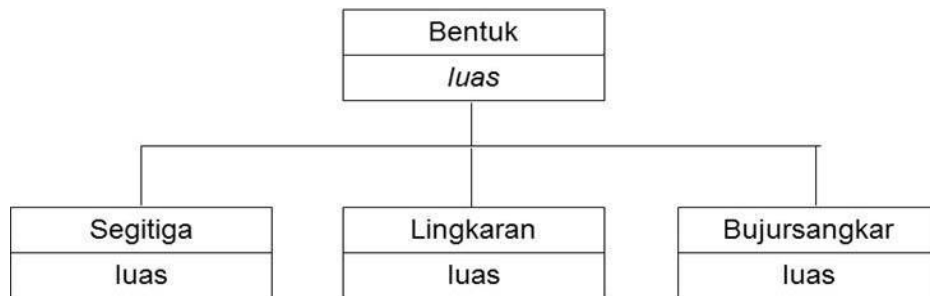
1.1. TUJUAN

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

Polimorfisme Memiliki arti “banyak bentuk”, melakukan hal yang sama untuk berbagai data yang berbeda, mengirimkan pesan yang sama ke berbagai objek yang berbeda karena tiap objek memberi respons dengan cara yang berbeda. Berikut ini merupakan contoh polimorfisme.



Polimorfisme memiliki syarat-syarat sebagai berikut:

- Ada hirarki pewarisan
- Kelas dalam hirarki pewarisan harus memiliki fungsi virtual (virtual method) dengan signature yang sama
- Menggunakan pointer atau rujukan ke kelas induk. Pointer digunakan untuk memanggil fungsi virtual

Polimorfisme dapat diimplementasikan dengan menggunakan dasar function overriding (melakukan redefinisi suatu fungsi di kelas anak,

fungsi yang di-override memiliki signature sama, signature sama : tipe balik, nama fungsi, parameter sama) dan pewarisan.

Suatu kelas disebut abstrak apabila memiliki minimal satu fungsi abstrak. Fungsi abstrak merupakan fungsi yang tidak memiliki definisi (hanya deklarasi fungsi)/menggunakan fungsi virtual (pure virtual).

```
virtual balikan namaFungsi (parameter) = 0
```

overloading : penggunaan satu nama untuk beberapa method yang berbeda (berbeda parameter), jadi dalam satu class terdapat lebih dari satu method yang memiliki nama yang sama tetapi memiliki parameter yang berbeda.

overriding : menyatakan keadaan dimana method pada subclass menolak method pada superclass-nya. Ciri dari Overriding adalah nama method, daftar parameter, dan return type yang harus sama.

Berikut ini adalah contoh program overloading (sederhana)

```
#include <iostream.h>
#include <conio.h>
class balok {
double p,l,t;
public : double Luaspermukaan() {
return(2*p*l)+(2*p*t)+(2*l*t);
}

public : double volume() {
return(p*l*t);
}
public : double volume(double t) {
return(p*l*t);
```

```

    }
    public : double volume(double t, double l) {
    return(p*l*t);
    }
    public : double volume(double t, double l, double p) {
    return(p*l*t);
    }
    void setpanjang(double pan){
    p = pan;
    }
    void setlebar(double leb){
    l = leb;
    }
    void settinggi(double tin){
    t = tin;
    }
};

void main(){
int pa,le,ti,pp,ll,tt;
balok tes;
cout<<"Balok"<<endl;
cout<<"Input Panjang : "; cin>>pa;
cout<<"Input Lebar : "; cin>>le;
cout<<"Input Tinggi : "; cin>>ti;
tes.setpanjang(pa);
tes.setlebar(le);
tes.settinggi(ti);


cout<<"Luas Permukaan Balok = "<<tes.Luaspermukaan()<<endl;
cout<<"Volume Balok = "<<tes.volume()<<endl;
cout<<"Tinggi baru = "; cin>>tt;

```

```

cout<<"Volume Balok 1 parameter = "<<tes.volume(tt)<<endl;
cout<<"Lebar baru = "; cin>>ll;
cout<<"Volume Balok 2 parameter = "<<tes.volume(tt, ll)<<endl;
cout<<"Panjang baru = "; cin>>pp;
cout<<"Volume Balok 3 parameter = "<<tes.volume(tt, ll, pp)<<endl;

getch();
}

```

Contoh Program Overriding (sederhana)

```

#include <constream.h>
#include <string.h>

class person {
public : virtual void speak();
};

class man : public person{
public : void speak();
};

void person :: speak(){
cout << "Person Speak "<<endl;
}

void man :: speak(){
cout << "Man Speak "<<endl;
}

void main(){
person p;

```

```
man m;  
p.speak();  
m.speak();
```

```
person *ptr = new man;  
ptr->speak();  
getch();  
}
```

BAB II

PEMBAHASAN

2.1. LANGKAH KERJA

Buatlah 1 program dengan tema buah bagi yang nomor ganjil dan jika nomor genap maka tema hewan dengan menggunakan prinsip PEWARISAN.

2.2. PEMBAHASAN

2.2.1. Program Hewan Pertama

Input Program

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

class hewan{
    protected :
        char nama [50];
        char sifathewan[50];
        char suara [50];
    public :
        virtual void informasi ()=0;
};

class kuda : public hewan{
    public :
        kuda (char *bo, char *jo, char *so){
            strcpy(nama, bo);
            strcpy(sifathewan, jo);
```



```

        strcpy(suara, so);
    }
    void informasi(){
        cout<<endl;
        cout<<"INFORMASI HEWAN"<<endl;
        cout<<"====="<<endl;
        cout<<"Nama      :kuda"<<endl;
        cout<<"Sifat Hewan :berlari dengan kencang"<<endl;
        cout<<"suara     :ihiii"<<endl;
    }
};

class kucing : public hewan{
    public :
        kucing (char *bo, char *jo, char *so){
            strcpy(nama, bo);
            strcpy(sifat hewan, jo);
            strcpy(suara, so);
        }
    void informasi(){
        cout<<endl;
        cout<<"INFORMASI HEWAN"<<endl;
        cout<<"====="<<endl;
        cout<<"Nama      :kucing"<<endl;
        cout<<"Sifat Hewan :berburu tikus tikus nakal"<<endl;
        cout<<"suara     :MEAW"<<endl;
    }
};

class Elang : public hewan{
    public :

```

```

Elang (char *bo, char *jo, char *so){
    strcpy(nama, bo);
    strcpy(sifathewan, jo);
    strcpy(suara, so);
}

void informasi(){
    cout<<endl;
    cout<<"INFORMASI HEWAN"<<endl;
    cout<<"====="<<endl;
    cout<<"Nama      :Elang"<<endl;
    cout<<"Sifat Hewan  :Terbang Tinggi Di
Angkasa"<<endl;
    cout<<"Suara      :ngeak"<<endl;
}

};

int main(){
    hewan *bin;
    kuda da ("kuda", "berlari dengan kencang","ihiiii");
    kucing cing ("kucing", "berburu tikus tikus nakal","MEAW");
    Elang el ("Elang", "Terbang Tinggi Di Angkasa","ngeak");
    int pilihan;
    menu:
    system ("cls");
    cout<<"MENU PROGRAM HEWAN"<<endl;
    cout<<"====="<<endl;
    cout<<"1. KUDA" <<endl;
        cout<<"2. KUCING" <<endl;
        cout<<"3. ELANG"<<endl;
        cout<<"4. keluar"<<endl;
    cout<<"masukan pilihan : ";cin>>pilihan;
    switch(pilihan){

```

```

        case 1 :
            bin = &da;
            bin->informasi();
            getch();
        case 2 :
            bin = &cing;
            bin->informasi();
            getch();
        case 3 :
            bin = &el;
            bin->informasi();
            getch();
        case 4 :
            cout<< endl;
            cout<<"Terimakasih"<<endl;
            return 0 ;

    }

}

```

Penjelasan :

```

class hewan{
    protected :
        char nama [50];
        char sifathewan[50];
        char suara [50];
    public :
        virtual void informasi ()=0;
};

```

Class hewan untuk mendeklarasikan hewan sebagai kelasnya. Qwtey. Dan public untuk mendeklarasikan bahwa hak aksesnya dapat diakses

secara umum. Virtual void informasi ()=0 berfungsi untuk mendukung adanya polimerfisme function.

```
class kuda : public hewan{
    public :
        kuda (char *bo, char *jo, char *so){
            strcpy(nama, bo);
            strcpy(sifathewan, jo);
            strcpy(suara, so);
        }
    void informasi(){
        cout<<endl;
        cout<<"INFORMASI HEWAN"<<endl;
        cout<<"====="<<endl;
        cout<<"Nama      :kuda"<<endl;
        cout<<"Sifat Hewan :berlari dengan kencang"<<endl;
        cout<<"suara     :ihiii"<<endl;
    }
}
```

class kuda : public hewan untuk mendeklarasikan bahwa kelas kuda dapat mengakses kelas hewan karena kelas kuda adalah turunannya. Public berfungsi mendeklarasi bahwa hak aksesnya dapat di akses secara umum. Void informasi merupakan fungsi dengan nama informasi yang akan menampilkan informasi hewan berupa nama, sifat hewan, dan suara. Begitu pula pada kelas kucing dan elang.

```

int main(){
    hewan *bin;
    kuda da ("kuda", "berlari dengan kencang","ihiiii");
    kucing cing ("kucing", "berburu tikus tikus nakal","MEAW");
    Elang el ("Elang", "Terbang Tinggi Di Angkasa","ngeak");
    int pilihan;
    menu:
        system ("cls");
        cout<<"MENU PROGRAM HEWAN"<<endl;
        cout<<"===== "<<endl;
        cout<<"1. KUDA" <<endl;
        cout<<"2. KUCING" <<endl;
        cout<<"3. ELANG"<<endl;
        cout<<"4. keluar"<<endl;
        cout<<"masukan pilihan : ";cin>>pilihan;
        switch(pilihan){
            case 1 :
                bin = &da;
                bin->informasi();
                getch();
            case 2 :
                bin = &cing;
                bin->informasi();
                getch();
            case 3 :
                bin = &el;
                bin->informasi();
                getch();

```

```

        case 4 :

                cout<< endl;
                cout<<"Terimakasih"<<endl;
                return 0 ;

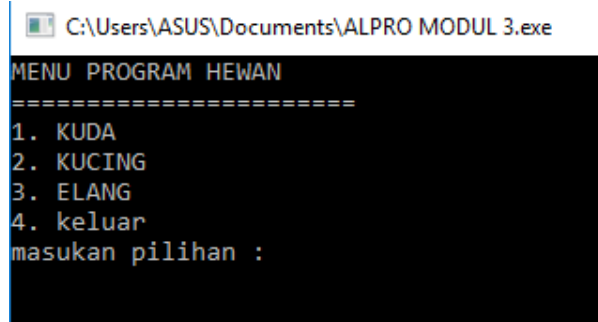
        }

}

```

Int main berfungsi sebagai fungsi utama program. Dan disini terdapat menu yang memiliki empat pilihan yang pertama adalah kuda maka akan menampilkan nama : kuda, sifat hewan : berlari dengan kencang dan suara : ihiii begitu pula pada menu kucing dan elang maka akan menampilkan informasi sesuai int main yang kita masukan tadi. Contohnya : `bin = &da; bin->informasi(); getch();` untuk mendeklarasi binatang yang merupakan inisialisasi dari kelas kuda yang dideklarasikan dengan variabel da, dikeluarkan dengan fungsi informasi yang terdapat pada kelas kuda.

Output program :



```

C:\Users\ASUS\Documents\ALPRO MODUL 3.exe
MENU PROGRAM HEWAN
=====
1. KUDA
2. KUCING
3. ELANG
4. keluar
masukan pilihan :

```

Gambar 2.1. Output menu program hewan

```
C:\Users\ASUS\Documents\ALPRO MODUL 3.exe
MENU PROGRAM HEWAN
=====
1. KUDA
2. KUCING
3. ELANG
4. keluar
masukan pilihan : 1

INFORMASI HEWAN
=====
Nama          :kuda
Sifat Hewan   :berlari dengan kencang
suara         :ihiii
```

Gambar 2.2. Menampilkan informasi hewan

```
C:\Users\ASUS\Documents\ALPRO MODUL 3.exe
=====
1. KUDA
2. KUCING
3. ELANG
4. keluar
masukan pilihan : 1

INFORMASI HEWAN
=====
Nama          :kuda
Sifat Hewan   :berlari dengan kencang
suara         :ihiii

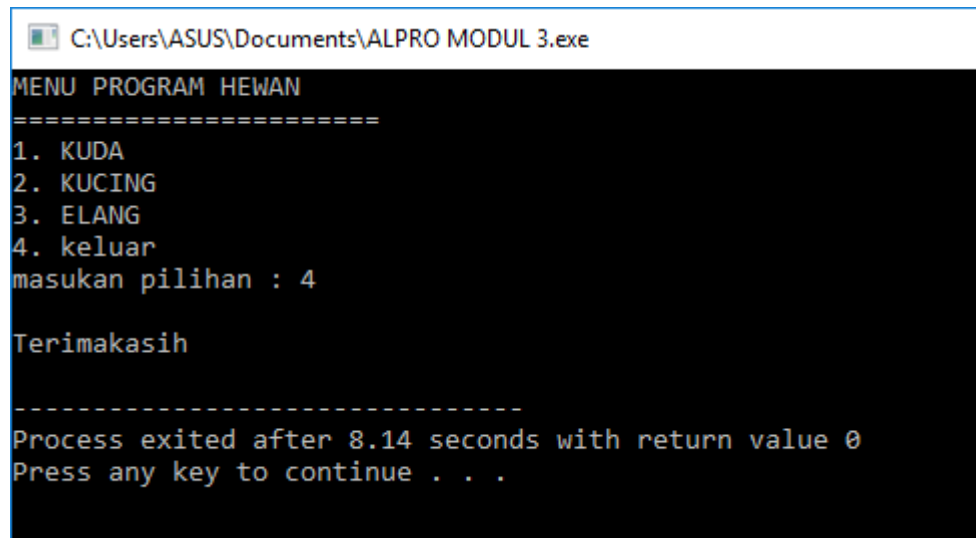
INFORMASI HEWAN
=====
Nama          :kucing
Sifat Hewan   :berburu tikus tikus nakal
suara         :MEAW

INFORMASI HEWAN
=====
Nama          :Elang
Sifat Hewan   :Terbang Tinggi Di Angkasa
Suara         :ngeak

Terimakasih

-----
Process exited after 47.09 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.3. output keseluruhan



```
C:\Users\ASUS\Documents\ALPRO MODUL 3.exe
MENU PROGRAM HEWAN
=====
1. KUDA
2. KUCING
3. ELANG
4. keluar
masukan pilihan : 4

Terimakasih

-----
Process exited after 8.14 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.4. menu keluar

BAB III

KESIMPULAN

Polimorfisme artinya “memiliki banyak bentuk” maksudnya satu hal yang sama dapat memiliki beberapa bentuk yang berbeda. Terdapat dua tipe polimerfisme, yaitu saat pengkompilasian (compile time) dan saat runtime. Polimorfisme saat pengkompilasian (compile time) dalam hal ini adalah overloading pada fungsi dan overloading pada operator yang pernah saya bahas di postingan sebelumnya. Sedangkan untuk polimorfisme saat runtime fungsi pada kelas dasar di deklarasikan untuk umum. Polimorfisme merupakan suatu konsep yang menyatakan bahwa sesuatu yang sama dapat memiliki berbagai bentuk dan perilaku yang berbeda. Dalam hal ini polimorfisme merupakan suatu sifat menyandarkan pada kesamaan nama dalam program. Pengenal data, instans, dan metode, bahkan nama fungsi dapat dibuat dengan nama yang sama untuk kegunaan yang berbeda.

DAFTAR PUSTAKA

Diana. Asisten Praktikum Teknik Informatika. Algoritma dan Pemrograman II. 2020. *Cover laporan hasil dan modul Algoritma dan pemrograman II*. Universitas Palangka Raya. Fakultas Teknik. Jurusan Teknik Informatika.

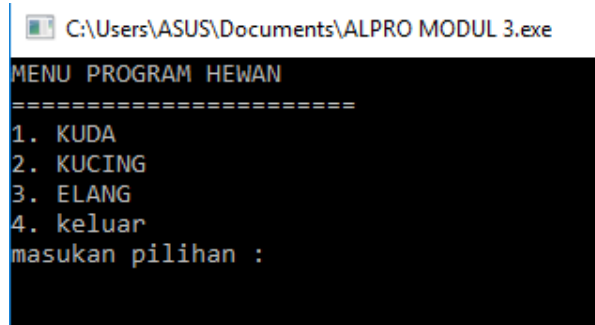
Risky Aditya. *Tutorial pemrograman berorientasi objek dengan c polimorfisme studi kasus*.

<https://adityarizki.net/tutorial-pemrograman-berorientasi-obyek-dengan-c-polimorfisme-studi-kasus/>

Nblognlife. 2017. *Polimerfisme pada c*.

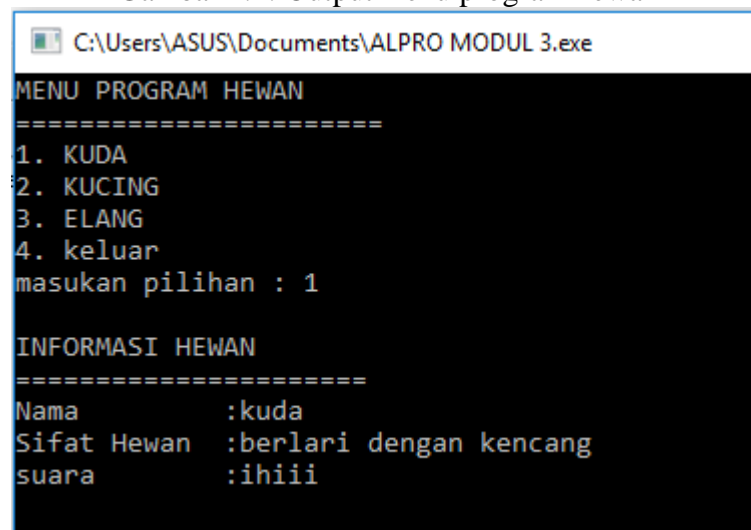
<http://www.nblognlife.com/2017/07/polimorfisme-pada-c.html>

LAMPIRAN



```
C:\Users\ASUS\Documents\ALPRO MODUL 3.exe
MENU PROGRAM HEWAN
=====
1. KUDA
2. KUCING
3. ELANG
4. keluar
masukan pilihan :
```

Gambar 2.1. Output menu program hewan



```
C:\Users\ASUS\Documents\ALPRO MODUL 3.exe
MENU PROGRAM HEWAN
=====
1. KUDA
2. KUCING
3. ELANG
4. keluar
masukan pilihan : 1

INFORMASI HEWAN
=====
Nama          :kuda
Sifat Hewan   :berlari dengan kencang
suara         :ihiii
```

Gambar 2.2. Menampilkan informasi hewan

```
C:\Users\ASUS\Documents\ALPRO MODUL 3.exe

=====
1. KUDA
2. KUCING
3. ELANG
4. keluar
masukan pilihan : 1

INFORMASI HEWAN
=====
Nama          :kuda
Sifat Hewan   :berlari dengan kencang
suara         :ihiii

INFORMASI HEWAN
=====
Nama          :kucing
Sifat Hewan   :berburu tikus tikus nakal
suara         :MEAW

INFORMASI HEWAN
=====
Nama          :Elang
Sifat Hewan   :Terbang Tinggi Di Angkasa
Suara         :ngeak

Terimakasih

-----
Process exited after 47.09 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.3. output keseluruhan

```
C:\Users\ASUS\Documents\ALPRO MODUL 3.exe

MENU PROGRAM HEWAN
=====
1. KUDA
2. KUCING
3. ELANG
4. keluar
masukan pilihan : 4

Terimakasih

-----
Process exited after 8.14 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.4. menu keluar

