

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : M. ADE SHOFY**  
**NIM : 193010503006**  
**KELAS : A**  
**MODUL : III (POLIMORFISME)**

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

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



Nama : M. Ade Shofy  
NIM : 193010503006  
Kelas : A  
Modul : POLIMORFISME

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	7
BAB II Pembahasan	60	47
BAB III Kesimpulan	20	10
Daftar Pustaka	5	5
Lampiran	5	5
Jumlah	100	

Penilai  
Asisten Praktikum

Diana

# BAB I

## TUJUAN DAN LANDASAN TEORI

### I. TUJUAN

Setelah menyelesaikan modul ini, mahasiswa diharapkan mampu membuat folimorfisme.

### II. DASAR TEORI

Polimorfisme merupakan fitur pemrograman berorientasi obyek yang penting setelah pengkapsulan (encapsulation) dan pewarisan (inheritance). Polimorfisme berasal dari bahasa Yunani, ***poly***(banyak) dan ***morphos*** (bentuk). Polimorfisme menggambarkan kemampuan kode-kode bahasa C++ berperilaku berbeda tergantung situasi pada waktu run (program berjalan).

Contoh polimorfisme yang konkrit dalam dunia nyata yaitu mobil. Mobil yang ada di pasaran terdiri atas berbagai tipe dan berbagai merk, tetapi semuanya memiliki interface kemudi yang sama, seperti: stir, tongkat transmisi, pedal gas dan rem. Jika seseorang dapat mengemudikan satu jenis mobil saja dari satu merk tertentu, maka orang itu akan dapat mengemudikan hampir semua jenis mobil yang ada, karena semua mobil tersebut menggunakan interface yang sama. Harus diperhatikan disini bahwa interface yang sama tidak berarti cara kerjanya juga sama. Misal pedal gas, jika ditekan maka kecepatan mobil akan meningkat, tetapi bagaimana proses peningkatan kecepatan ini dapat berbeda-beda untuk setiap jenis mobil.

Konstruksi polimorfisme dalam pemrograman berorientasi obyek memungkinkan untuk mengadakan **ikatan dinamis** (juga disebut ikatan tunda, atau ikatan akhir). Apabila fungsi-fungsi dari suatu kelas dasar didefinisikan ulang atau ditindih pada kelas turunan, maka obyek-obyek yang dihasilkan hirarki kelas berupa obyek polimorfik. Polimorfik artinya mempunyai banyak bentuk atau punya kemampuan untuk mendefinisi banyak bentuk.

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.

Salah satu bentuk polimorfisme pada C++ dapat digunakan pada fungsi atau operator dan dikenal sebagai istilah *overloading*. *Overloading* terhadap fungsi akan memungkinkan sebuah fungsi dapat menerima bermacam-macam tipe dan memberikan nilai balik yang bervariasi pula.

Berikut merupakan contoh sintaks sederhana mengenai polimorfisme :

```
1. class induk
2. {
3.     //member
4. };
5. class anak: public induk
6. {
7.     //member
8. };
9. int main(){
10.     induk *pointer_anak;
11.     anak objek_anak;
12.     pointer_anak = &objek_anak
13.}
```

Dari permissalan di atas, kita ketahui bahwa terdapat kelas anak yang menginduk pada kelas induk. Kemudian pada fungsi main(), terdapat deklarasi pointer bertipe kelas induk yang digunakan sebagai penunjuk objek kelas anak.

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**

## BAB II

### PEMBAHASAN

#### 1. Program pertama

```
#include <iostream>
#include <conio.h>
using namespace std;

//Kelas Dasar Mangga (Kelas Abstrak)
class Mangga
{
protected:
    char nama[20];

public:
    //fungsi virtual murni
    virtual void info_bisa() = 0;
};

//kelas Bangkok turunan dari kelas Mangga
class Bangkok : public Mangga
{
public:
    Bangkok(char* nm)
    {
        strcpy(nama, nm);
    }

    void info_bisa()
    {
        cout<<"Mangga "<<nama<<" manis besar";
    }
};

//kelas Madu turunan dari kelas Mangga
class Madu : public Mangga
{
public:
    Madu(char* nm)
    {
        strcpy(nama, nm);
    }
};
```

```

    }

    void info_bisa()
    {
        cout<<"Mangga "<<nama<<" manis seperti madu";
    }
};

int main()
{
    //deklarasi objek
    Mangga *obj_mangga;
    Bangkok bnk("Bangkok");
    Madu mda("Madu");

    cout<<"POLIMORFISME 1"<<endl;
    cout<<"-----"<<endl;

    //menunjuk ke objek dari kelas Bangkok
    obj_mangga = &bnk;
    obj_mangga->info_bisa();

    cout<<endl;

    //menunjuk ke objek dari kelas Madu
    obj_mangga = &mda;
    obj_mangga->info_bisa();

    _getche();
    return 0;
}

```

Code program 2.1

Output program

```

POLIMORFISME 1
-----
Mangga Bangkok manis besar
Mangga Madu manis seperti madu

```

Output program 2.1

Program ini merupakan program polimorfisme yang hanya menggunakan fungsi virtual murni saja dengan class utama merupakan class mangga lalu class bangkok dan class madu yang merupakan turunan dari class mangga.

Perhatikan pada potongan source code berikut.

```
//Kelas Dasar Mangga (Kelas Abstrak)
class Mangga
{
protected:
    char nama[20];

public:
    //fungsi virtual murni
    virtual void info_bisa() = 0;
};
```

Pada potongan source code ini sudah jelas bahwa fungsi yang digunakan merupakan fungsi virtual murni. Jika suatu kelas setidaknya memiliki sebuah fungsi virtual murni maka kelas tersebut disebut dengan kelas abstrak. Pada kelas abstrak tidak diperbolehkan untuk menciptakan objek dan yang diperbolehkan hanyalah pointer ke objek berkelas.

## 2. Program kedua

```
#include <iostream>
#include <conio.h>
using namespace std;

//Kelas Dasar Buah (Kelas Abstrak)
class Buah
{
protected:
    char nama_buah[20];
    char rasa_b[20];

public:
    void informasi()
    {
        cout<<"Informasi Buah"<<endl;
    }
};
```

```

//fungsi virtual murni
virtual void namaBuah() = 0;
virtual void rasa() = 0;
};

//kelas Manggis turunan dari kelas Buah
class Manggis : public Buah
{
public:
    Manggis(char* nm, char *rs)
    {
        strcpy(nama_buah, nm);
        strcpy(rasa_b, rs);
    }

    void informasi()
    {
        cout<<"Informasi Manggis"<<endl;
    }

    void namaBuah()
    {
        cout<<"Nama : "<<nama_buah<<endl;
    }

    void rasa()
    {
        cout<<"Rasa : \""<<rasa_b<<"\"<<endl;
    }
};

//kelas Jeruk turunan dari kelas Buah
class Jeruk : public Buah
{
public:
    Jeruk(char* nm, char *rs)
    {
        strcpy(nama_buah, nm);
        strcpy(rasa_b, rs);
    }

    void informasi()
    {
        cout<<"Informasi Buah"<<endl;
    }
};

```



```

    }

    void namaBuah()
    {
        cout<<"Nama : "<<nama_buah<<endl;
    }

    void rasa()
    {
        cout<<"Rasa : \"<<rasa_b<<\"<<endl;
    }
};

int main()
{
    //deklarasi objek
    Buah *obj_bah;
    Manggis mgs("Manggis", "manis");
    Jeruk jrk("Jeruk", "masam");

    cout<<"POLIMORFISME 2"<<endl;
    cout<<"-----"<<endl;

    //menunjuk ke objek dari kelas Manggis
    obj_bah = &mgs;
    obj_bah->informasi();
    obj_bah->namaBuah();
    obj_bah->rasa();

    cout<<endl;

    //menunjuk ke objek dari kelas Jeruk
    obj_bah = &jrk;
    obj_bah->informasi();
    obj_bah->namaBuah();
    obj_bah->rasa();

    _getche();
    return 0;
}

```

Code program 2.2

### Output program

```
POLIMORFISME 2
```

```
-----
```

```
Informasi Buah
```

```
Nama : Manggis
```

```
Rasa : "manis"
```

```
Informasi Buah
```

```
Nama : Jeruk
```

```
Rasa : "masam"
```

### Output program 2.2

Program di atas merupakan program polimorfisme dengan class utama adalah class buah dan class manggis dan class jeruk merupakan class turunannya.

Pada kelas dasar Buah terdapat dua fungsi yang berbeda, yaitu fungsi biasa dan fungsi virtual murni

Sekarang perhatikan output program, ketika `obj_bah->informasi();` dijalankan hasilnya adalah string “Informasi Buah”. Kenapa bukan “Informasi Manggis/Jeruk” padahal pointer sudah menunjuk pada kelas turunan Manggis/Jeruk? Hal ini dikarenakan kata “virtual” tidak ditambahkan pada fungsi `informasi()` yang ada pada kelas dasar Buah. Dan jangan lupa juga bahwa fungsi `informasi()` dari kelas dasar Buah telah diwariskan ke kelas turunannya.

### **BAB III**

### **KESIMPULAN**

Polimorfisme merupakan fitur pemrograman berorientasi obyek yang penting setelah pengkapsulan (encapsulation) dan pewarisan (inheritance). Polimorfisme berasal dari bahasa Yunani, *poly*(banyak) dan *morphos* (bentuk). Polimorfisme menggambarkan kemampuan kode-kode bahasa C++ berperilaku berbeda tergantung situasi pada waktu run (program berjalan). Contoh polimorfisme yang konkrit dalam dunia nyata yaitu mobil. Mobil yang ada di pasaran terdiri atas berbagai tipe dan berbagai merk, tetapi semuanya memiliki interface kemudi yang sama, seperti: stir, tongkat transmisi, pedal gas dan rem.

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**

## DAFTAR PUSTAKA

Adityarizki. *Tutorial PBO Dengan C Polimorfisme Studi Kasus.*

<https://adityarizki.net/tutorial-pemrograman-berorientasi-obyek-dengan-c-polimorfisme-studi-kasus/> (diakses pada : Selasa, 21 April 2020 pada pukul 18:00).

Sinuarduino. *Polimorfism Dan Virtual Function Pada Classes.*

<https://www.sinuarduino.com/artikel/polymorphism-dan-virtual-function-pada-classes-cpp/> (diakses pada : Selasa, 21 April 2020 pada pukul 18:00).

Nblognlife. 2017, *Polimorfisme Pada C .*

<http://www.nblognlife.com/2017/07/polimorfisme-pada-c.html> (diakses pada : Selasa, 21 April 2020 pada pukul 18:00).

## LAMPIRAN

### POLIMORFISME 1

-----

Mangga Bangkok manis besar  
Mangga Madu manis seperti madu

Output program 2.1

### POLIMORFISME 2

-----

Informasi Buah  
Nama : Manggis  
Rasa : "manis"

Informasi Buah  
Nama : Jeruk  
Rasa : "masam"

Output program 2.2