

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



NAMA : ELNATAN KENINGATKO
NIM : 193020503038
KELAS : A
MODUL : III (POLIMORFISME)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2020

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



NAMA : ELNATAN KENINGATKO
NIM : 193020503038
KELAS : A
MODUL : III (POLIMORFISME)

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	7
BAB II Pembahasan	60	50
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 membuat polimorfisme.

1.2. Landasan Teori

1.2.1. Pengertian Polimorfisme

Polymorphism adalah kemampuan untuk menggunakan operator atau fungsi dalam berbagai cara. Polimorfisme memberikan arti yang berbeda atau fungsinya kepada operator atau fungsi. Poly, merujuk ke banyak, menandakan banyak kegunaan dari operator dan fungsi ini. Fungsi tunggal penggunaan atau operator berfungsi dalam banyak cara bisa disebut polimorfisme. Polimorfisme mengacu pada kode, operasi atau benda yang berperilaku berbeda dalam konteks yang berbeda.

Polimorfisme adalah fitur yang kuat dari bahasa pemrograman berorientasi obyek C++. Sebuah operator + berperilaku berbeda dalam konteks yang berbeda, seperti integer, float atau string yang mengacu konsep polimorfisme. Konsep di atas mengarah ke operator overloading. Konsep overloading juga merupakan cabang dari polimorfisme. Ketika keluar operator atau fungsi yang beroperasi pada tipe data baru itu kelebihan beban. Polimorfisme fitur ini mengarah pada konsep metode virtual.

Polimorfisme mengacu pada kemampuan untuk memanggil fungsi-fungsi yang berbeda dengan menggunakan hanya satu jenis fungsi panggil. Misalkan seorang programmer ingin kode kendaraan dari berbagai bentuk-bentuk seperti lingkaran, bujur sangkar, persegi panjang, dll Salah satu cara untuk mendefinisikan masing-masing dari kelas-kelas ini adalah memiliki fungsi anggota untuk masing-masing yang membuat kendaraan dari masing-masing bentuk. Pendekatan

nyaman lain pemrogram dapat mengambil adalah untuk mendefinisikan sebuah basis kelas bernama Shape dan kemudian membuat sebuah instance dari kelas tersebut. Programmer dapat memiliki array yang menyimpan pointer ke semua objek yang berbeda dari kendaraan diikuti oleh struktur loop sederhana untuk membuat kendaraan, sesuai dengan bentuk yang dikehendaki, dengan memasukkan pointer ke array yang didefinisikan. Pendekatan ini mengarah kepada fungsi yang berbeda dieksekusi oleh panggilan fungsi yang sama. Polimorfisme digunakan untuk memberikan makna yang berbeda untuk konsep yang sama. Ini adalah dasar untuk pelaksanaan fungsi virtual. Bentuk penulisan fungsi virtual :

```
virtual <tipe> <nama_fungsi>() {  
...  
}
```

Dalam polimorfisme, satu fungsi atau operator berfungsi dalam banyak hal tergantung pada penggunaan berfungsi dengan benar. Agar hal ini terjadi, kondisi berikut harus berlaku :

- a. Semua kelas-kelas yang berbeda harus berasal dari satu kelas dasar. Dalam contoh di atas, bentuk-bentuk kendaraan (lingkaran, segitiga, persegi panjang) yang dari satu kelas dasar yang disebut Shape.
- b. Fungsi anggota harus dinyatakan virtual di kelas dasar. Dalam contoh di atas, fungsi anggota untuk membuat kendaraan harus dibuat sebagai virtual untuk kelas dasar.

BAB II PEMBAHASAN

2.1. Program Pertama

```
//buah.h

#ifndef BUAH_H
#define BUAH_H

class Buah {
    public:
        virtual void tampil()=0;
};

class Fruit:public Buah {
    protected:
        char nama[10], rasa[20], warna[20], cirikhusus[20];
    public:
        Fruit(char *a,char *b,char *c, char *d);
        void tampil();
};
#endif
```

Blok code di atas merupakan header baru yang dibuat dan diberi nama “buah”. Header tersebut memiliki dua preprocessor , yaitu *ifndef* dan *define*. *ifndef* digunakan untuk memberikan bagian dari program yang akan dikompilasi, hal ini dapat dilakukan jika konstanta sudah didefinisikan pada bagian *define*. Jika nama konstanta terdefinisi, maka pernyataan akan dijalankan, namun jika nama konstanta tidak terdefinisi, maka pernyataan diabaikan. Sedangkan *define* digunakan untuk melaksanakan substitusi makro dari satu lembar/halaman teks dengan lembar/halaman teks yang lain melalui suatu file dimana teks tersebut digunakan.

Pada class induk yang diberi nama “Buah” dan memiliki akses *public* dengan *virtual void tampil()=0* sebagai fungsi *virtual*. Kemudian class turunan yang diberi nama “Fruit” dihubungkan dengan class induk agar class turunan dapat mengakses class induk. Class *Fruit* memiliki akses *protected* dengan atribut *nama[10]*, *rasa[20]*, *warna[20]* dan *cirikhusus[20]* yang

bertipe data *char*, akses *public* dengan atribut *Fruit(char *a, char *b, char *c, char *d)* sebagai konstruktor class turunan dan fungsi *void tampil()* yang juga digunakan pada class induk. Di akhir blok code tersebut terdapat *#endif* yang digunakan untuk mengkompilasi bahwa kondisi pernyataan pada *#if* telah benar, jika tidak maka akan diabaikan. Pernyataan kondisi berupa ekspresi konstanta yang dideklarasikan dengan *#define*.

```
//buah.cpp

#include <iostream>
#include <string.h>
#include "buah.h"
using namespace std;

Fruit::Fruit(char *a,char *b, char *c, char *d) {
    strcpy(nama,a);
    strcpy(rasa,b);
    strcpy(warna,c);
    strcpy(cirikhusus,d);
}

void Fruit::tampil() {
    cout<<"Nama      : "<<nama<<endl;
    cout<<"Rasa      : "<<rasa<<endl;
    cout<<"Warna      : "<<warna<<endl;
    cout<<"Ciri Khusus : "<<cirikhusus<<endl<<endl;
}
```

Blok code di atas merupakan source file yang diberi nama “buah”. Pada source file ini terdapat tiga header, yaitu *iostream* yang digunakan untuk memanggil fungsi *input* dan *output*, *string.h* yang digunakan untuk memanggil fungsi yang berhubungan dengan *string* maupun *substring* dan *buah.h* yang merupakan header baru yang dibuat sebelumnya yang digunakan untuk mengakses class, kemudian *using namespace std* yang digunakan untuk menangani setiap fungsi atau bagian *namespace std* yang bersangkutan.

Konstruktor Fruit memiliki variable *a*, *b*, *c*, dan *d* yang diberi pointer, gunanya sebagai penampung alamat yang nantinya data tersebut akan disalin ke variable *nama*, *rasa*, *warna* dan *cirikhusus* yang terdapat pada *protected* class turunan menggunakan *strcpy*. Kemudian class turunan dihubungkan dengan fungsi *tampil* yang berfungsi untuk mengikat nama fungsi ke nama class dan mengidentifikasi fungsi dari class turunan. Fungsi *tampil* tersebut menampilkan output berupa “Nama : ”, “Rasa : ”, “Warna : ” dan “Ciri Khusus : ” dengan memanggil variable yang berisikan data dari masing-masing output tersebut.

```
//Mbuah.cpp

#include <iostream>
#include "buah.h"
using namespace std;

int main() {
    Buah* b;
    int pilih;
    do{
        system("cls");
        cout<<"Buah-Buahan\n\n"
            <<"[1] Buah Naga\n"
            <<"[2] Apel\n"
            <<"[3] Mangga\n"
            <<"[4] Keluar\n\n"
            <<"Masukan pilihan anda : "; cin>>pilih;
        cout<<endl;

        switch(pilih) {
            case 1:{
                b=new Fruit("Buah naga", "Manis dan asam", "Merah
                keunguan", "Akar serabut");
                b->tampil();
                system("pause");
                break;
            }
        }
    }
}
```

```

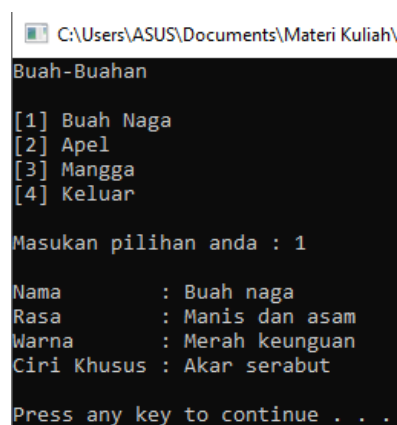
        case 2:{
            b=new Fruit("Apel","Manis","Merah atau hijau","Akar
            serabut");
            b->tampil();
            system("pause");
            break;
        }
        case 3:{
            b=new Fruit("Mangga","Manis dan asam","Hijau","Akar
            tunggang");
            b->tampil();
            system("pause");
            break;
        }
        case 4:{
            system("pause");
            break;
        }
        default:{
            cout<<"Pilihan tidak tersedia!\n"
                <<"Silahkan memilih sesuai nomor yang
                tersedia.\n";
            system("pause");
            break;
        }
    }
}
while(pilih!=4);
delete b;
return 0;
}

```

Blok code di atas merupakan main program dengan deklarasi *int main()* dan menggunakan *switch case*. Kemudian class *Buah* diberi pointer dan dibuat objek baru dengan nama “b”, objek tersebut mengakses class *Buah*. Variable *pilih* digunakan untuk menampung data yang digunakan untuk menentukan case nomor berapa yang akan dieksekusi. Variable tersebut

bertipe data *integer* (bilangan bulat). Pada setiap case terdapat operasi *new* yang fungsinya untuk mengalokasikan memori baru, sehingga parameter *Fruit* dialokasikan ke dalam objek *b* dan setiap case memiliki parameter yang berbeda-beda. Operasi *new* selalu diakhiri dengan *delete* yang berfungsi untuk menghapus memori yang telah dibuat karena data yang tersimpan pada heap tidak akan terhapus secara otomatis, sehingga bisa mengakibatkan kesalahan memori yang disebut “memory leak”. Heap sendiri merupakan bagian memori program yang tidak terpakai dan dapat digunakan untuk melakukan alokasi memori secara dinamis saat program sedang berjalan.

Pada saat user memilih nomor satu, maka case pertama akan dieksekusi. Case pertama memiliki parameter “Buah Naga”, “Manis dan asam”, “Merah keunguan” dan “Akar serabut”, parameter tersebut telah disimpan ke dalam objek *b*. Kemudian fungsi *tampil* dipanggil dengan menghubungkannya ke objek *b* menggunakan tanda “->”. Begitu seterusnya sampai pada user memilih case ketiga. Case keempat tidak memiliki parameter karena hanya digunakan untuk mengakhiri program dan tidak ada statement yang dieksekusi. Berikut ini hasil output program :



```
C:\Users\ASUS\Documents\Materi Kuliah\  
Buah-Buahan  
[1] Buah Naga  
[2] Apel  
[3] Mangga  
[4] Keluar  
  
Masukan pilihan anda : 1  
  
Nama      : Buah naga  
Rasa      : Manis dan asam  
Warna     : Merah keunguan  
Ciri Khusus : Akar serabut  
  
Press any key to continue . . .
```

Gambar 2.1. Output Program Pilihan Pertama

```
C:\Users\ASUS\Documents\Materi Kuliah\Alg
Buah-Buahan
[1] Buah Naga
[2] Apel
[3] Mangga
[4] Keluar

Masukan pilihan anda : 2

Nama      : Apel
Rasa      : Manis
Warna     : Merah atau hijau
Ciri Khusus : Akar serabut

Press any key to continue . . .
```

Gambar 2.2. Output Program Pilihan Kedua

```
C:\Users\ASUS\Documents\Materi Kuliah\
Buah-Buahan
[1] Buah Naga
[2] Apel
[3] Mangga
[4] Keluar

Masukan pilihan anda : 3

Nama      : Mangga
Rasa      : Manis dan asam
Warna     : Hijau
Ciri Khusus : Akar tunggang

Press any key to continue . . .
```

Gambar 2.3. Output Program Pilihan Ketiga

```
C:\Users\ASUS\Documents\Materi Kuliah\
Buah-Buahan
[1] Buah Naga
[2] Apel
[3] Mangga
[4] Keluar

Masukan pilihan anda : 4

Press any key to continue . . .
```

Gambar 2.4. Output Program Pilihan Keempat

BAB III

KESIMPULAN

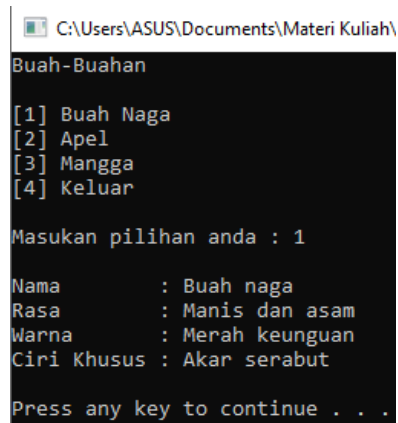
Polimorfisme memiliki arti “banyak bentuk”, melakukan hal yang sama untuk berbagai data yang berbeda, mengirimkan pesan yang sama ke berbagai objek yang berbeda karena tiap objek memberi respons dengan cara yang berbeda.

Di dalam polimorfisme harus memiliki kondisi dimana semua kelas-kelas yang berbeda harus berasal dari satu kelas dasar dan fungsi anggota harus dinyatakan virtual di kelas dasar.

DAFTAR PUSTAKA

- Anonim. 2017. *Polymorphism dan pengertiannya pada C++*, diakses dari <https://catataninformatikaku.blogspot.com/2017/03/polymorphism-dan-pengertiannya-pada-c.html>, pada 22 April 2020.
- Tim Penyusun : Dosen Teknik Informatika. 2020. *Modul Praktikum Algoritma dan Pemrograman II*. Palangka Raya : Jurusan Teknik Informatika.

LAMPIRAN



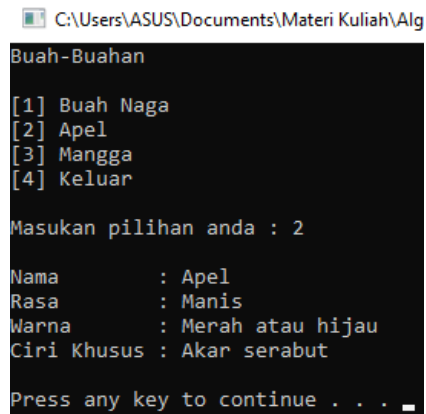
```
C:\Users\ASUS\Documents\Materi Kuliah\
Buah-Buahan
[1] Buah Naga
[2] Apel
[3] Mangga
[4] Keluar

Masukan pilihan anda : 1

Nama      : Buah naga
Rasa      : Manis dan asam
Warna     : Merah keunguan
Ciri Khusus : Akar serabut

Press any key to continue . . .
```

Gambar 2.1. Output Program Pilihan Pertama



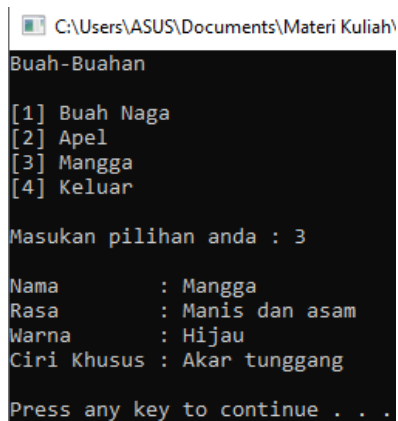
```
C:\Users\ASUS\Documents\Materi Kuliah\Alg
Buah-Buahan
[1] Buah Naga
[2] Apel
[3] Mangga
[4] Keluar

Masukan pilihan anda : 2

Nama      : Apel
Rasa      : Manis
Warna     : Merah atau hijau
Ciri Khusus : Akar serabut

Press any key to continue . . .
```

Gambar 2.2. Output Program Pilihan Kedua



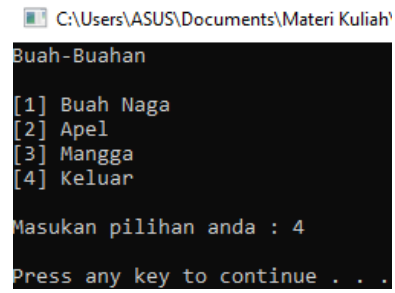
```
C:\Users\ASUS\Documents\Materi Kuliah\
Buah-Buahan
[1] Buah Naga
[2] Apel
[3] Mangga
[4] Keluar

Masukan pilihan anda : 3

Nama      : Mangga
Rasa      : Manis dan asam
Warna     : Hijau
Ciri Khusus : Akar tunggang

Press any key to continue . . .
```

Gambar 2.3. Output Program Pilihan Ketiga



```
C:\Users\ASUS\Documents\Materi Kuliah>
Buah-Buahan
[1] Buah Naga
[2] Apel
[3] Mangga
[4] Keluar

Masukan pilihan anda : 4

Press any key to continue . . .
```

Gambar 2.4. Output Program Pilihan Keempat