

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



NAMA : ABIGAEL GERRY PRATAMA
NIM : 193020503021
KELAS : A
MODUL : III (POLIMORFISME)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA

2020

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



Nama : ABIGael GERRY PRATAMA
NIM : 193020503021
Kelas : A
Modul : III (POLIMORFISME)

| Komposisi | MAX | Nilai |
|---------------------------------|-----|-------|
| BAB I Tujuan dan Landasan Teori | 10 | 7 |
| BAB II Pembahasan | 60 | 48 |
| BAB III Kesimpulan | 20 | 10 |
| Daftar Pustaka | 5 | 5 |
| Lampiran | 5 | 5 |
| Jumlah | 100 | |

Penilai

Asisten Praktikum

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

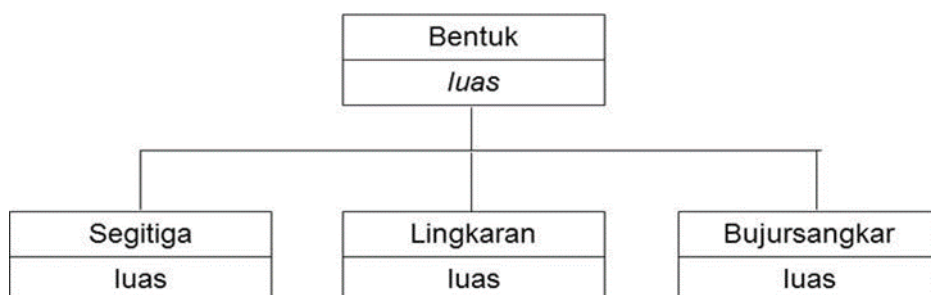
1.1. TUJUAN

Setelah menyelesaikan modul ini, mahasiswa diharapkan mampu membuat polimorfisme.

1.2. LANDASAN TEORI

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). 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 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.



Gambar 1.2 Polimorfisme

Polimorfisme memiliki syarat-syarat sebagai berikut:

1. Ada hirarki pewarisan
2. Kelas dalam hirarki pewarisan harus memiliki fungsi virtual (virtual method) dengan signature yang sama
3. 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.

BAB II

PEMBAHASAN

2.1. Pembahasan Program

Program ini disusun dengan 3 buah file source code dengan 1 buah project dan dijalankan menjadi satu compiler. Judul dari project ini diberi nama BUAH MODUL III. Dan file source code yang pertama berjudul Mbuah.cpp, yang kedua buah.cpp, dan yang ketiga buah.h. Berikut adalah pembahasan dari masing-masing source code.

```
#ifndef BUAH_H
#define BUAH_H
using namespace std;
class Buah {
    public:
        virtual void Hello();
};
class Mangga : public Buah {
    public:
        void Hello();
};
class Apel : public Buah {
    public:
        void Hello();
};
#endif
```

Program source file yang pertama yaitu Mbuah.cpp, yang menjadi program utama di project ini. #define sendiri digunakan untuk mendefinisikan suatu nilai tertentu kepada suatu nama konstanta. umumnya, pembuatan file_header menggunakan bahasa standart yang disediakan oleh c++. Using namespace std digunakan untuk membaca perintah cout, cin,

dan endl. Lalu terdapat tiga class pada program ini yaitu class buah, class mangga, dan class apel. Setiap class memiliki fungsi yang berbeda-beda. Class buah, mangga, apel memiliki anggota class yaitu public. Class buah menjadi class utamanya.

```
#include "buah.h"
#include <iostream>
void Buah::Hello() {
    cout << " Hallo, saya suka buah ..." << endl;
}
void Mangga::Hello() {
    cout << " Hallo, saya suka mangga ..." << endl;
}
void Apel::Hello() {
    cout << " Hallo, saya suka apel ..." << endl;
}
```

Pada source code yang kedua ini berjudul buah.cpp bertujuan untuk menampilkan hasil dari pilihan yang diinputkan. Misalkan class buah maka akan menampilkan “Halo, saya suka buah ...”, begitu juga dengan yang lain. Include “buah.h” digunakan untuk mencari file dari sebelumnya, dengan begitu using namespace std tidak digunakan pada source file ini.

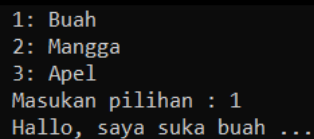
```
#include "buah.h"
#include <iostream>
#include <stdlib.h>
int main() {
    Buah* m;
    int pilihan;
do
{
```

```

        cout << " 1: Buah\n 2: Mangga\n 3: Apel\n Masukan pilihan : ";
    cin >> pilihan;
}
while ( pilihan < 1 || pilihan > 3 );
switch ( pilihan ) {
    case 1: m = new Buah; break;
    case 2: m = new Mangga; break;
    case 3: m = new Apel; break;
}
m->Hello(); delete m;
return 0;
}

```

Source file yang ketiga ini berjudul buah.h yang berfungsi untuk menu pilihan. Disini akan tersedia tiga pilihan, dan akan diminta memasukan salah satu pilihan, maka akan keluar hasil dari pilihan yang dimasukan. Berikut adalah hasil dari program yang dijalankan. Karena menggunakan pointer untuk mengakses classnya, maka diperlukan arrow operator untuk mengakses methodnya. Delete menghapus data yang telah disimpan dalam heap memory.

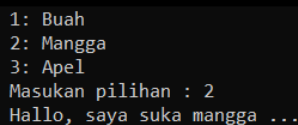


```

1: Buah
2: Mangga
3: Apel
Masukan pilihan : 1
Hallo, saya suka buah ...

```

Gambar 2.1 Pilihan 1



```

1: Buah
2: Mangga
3: Apel
Masukan pilihan : 2
Hallo, saya suka mangga ...

```

Gambar 2.2 Pilihan 2

```
1: Buah  
2: Mangga  
3: Apel  
Masukan pilihan : 3  
Hallo, saya suka apel ...
```

Gambar 2.3 Pilihan 3

BAB III

KESIMPULAN

Program ini bertujuan agar dapat memahami konsep polimorfisme dengan tepat dan mengkompail 3 source file menjadi 1 compiler, dan dibuat dalam satu project. Polimorfisme 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.

DAFTAR PUSTAKA

Belajar cpp. 2019. *Penjelasan dan Tipe-tipe Pewarisan (Inheritance)*.
<https://www.belajarcpp.com/tutorial/cpp/inheritance/> (diakses pada Selasa,
21 April 2020)

Adityarizki. 2011. *Tutorial Pemrograman Berorientasi Obyek dengan C++:
Polimorfisme (Studi Kasus)*. <https://adityarizki.net/tutorial-pemrograman-berorientasi-obyek-dengan-c-polimorfisme-studi-kasus/> (diakses pada
Selasa, 21 April 2020)

Sinauarduino. 2016. *Polymorphism dan Virtual Function Pada Classes C++*.
<https://www.sinauarduino.com/artikel/polymorphism-dan-virtual-function-pada-classes-cpp/> (diakses pada Selasa, 21 April 2020)

LAMPIRAN

```
1: Buah  
2: Mangga  
3: Apel  
Masukan pilihan : 1  
Hallo, saya suka buah ...
```

Gambar 2.1 Pilihan 1

```
1: Buah  
2: Mangga  
3: Apel  
Masukan pilihan : 2  
Hallo, saya suka mangga ...
```

Gambar 2.2 Pilihan 2

```
1: Buah  
2: Mangga  
3: Apel  
Masukan pilihan : 3  
Hallo, saya suka apel ...
```

Gambar 2.3 Pilihan 3