

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : ABIGAEL GERRY PRATAMA**  
**NIM : 193020503021**  
**KELAS : A**  
**MODUL : II (PEWARISAN)**

**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 : II (PEWARISAN)

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	8
BAB II Pembahasan	60	50
BAB III Kesimpulan	20	13
Daftar Pustaka	5	5
Lampiran	5	5
Jumlah	100	81

Penilai

Asisten Praktikum

Diana

# **BAB I**

## **TUJUAN DAN LANDASAN TEORI**

### **1.1. TUJUAN**

Setelah menyelesaikan modul ini, mahasiswa diharapkan mampu membuat kelas baru dari kelas yang sudah ada dengan pewarisan.

### **1.2. LANDASAN TEORI**

Inheritance atau Pewarisan adalah salah satu konsep pada object oriented programming, yang mengadopsi konsep pewarisan yang dimiliki oleh object pada dunia nyata. Konsep Pewarisan ini memiliki model relasi “adalah”, dimana sebuah class akan dimungkinkan untuk mengambil isi dari class lain sebagai isi dari class tersebut.

Konsep pewarisan akan mempermudah kita dalam membuat dan perawatan sebuah program, dengan menghilangkan “penggunaan kode berulang” saat pembuatan banyak class yang serupa.

Untuk mendirikan sebuah base class, sama seperti layaknya kita membuat class biasa. Untuk mendirikan derived class dan mewarisi informasi dari base class, mendirikan class sama seperti kita mendirikan class seperti biasanya dan akan dibutuhkan tambahan berupa daftar class penurunan (base class) yang ditulis setelah tanda “:”.

Dalam PBO, kita mengambil realita kehidupan sehari-hari. Kita melakukan pengamatan bahwa manusia secara alami sering melakukan pengelompokan atas objek atau benda. Sejauh ini kita mengetahui cara untuk melakukan pengelompokan – pengelompokan atas objek-objek yang serupa (menjadi kelas objek).

Pemberian access specifier pada daftar base class dalam mendirikan derived class dapat berupa public, protected atau private, masing-masing akan memberikan efek berbeda pada derived class tersebut. Dan efek tersebut adalah:

**Public Inheritance:** Jika mendaftarkan base class pada derived class menggunakan access specifier public maka akan membuat member dari base class yang memiliki sifat private, protected, dan public akan menjadi diri mereka sendiri pada derived class tersebut.

**Protected Inheritance:** jika mendaftarkan base class pada derived class menggunakan access specifier protected maka akan membuat member dari base class yang bersifat protected dan public menjadi bersifat protected pada derived class.

**Private Inheritance:** jika mendaftarkan base class pada derived class menggunakan access specifier private maka akan membuat member dari base class yang bersifat protected dan public menjadi bersifat private pada derived class.

Selain melakukan kategorisasi terhadap objek yang memiliki sekumpulan atribut dan perilaku yang sama, manusia sering melakukan pengelompokan terhadap objek yang memiliki kesamaan atas beberapa (tidak semua) atribut/perilaku. Contoh: Pengelompokan atas kendaraan bermotor, kemudian menggrupkannya berdasarkan suatu tipe atau jenis (mobil, truk, sepeda motor, dll.). Setiap subkategori ini merupakan kelas atas objek-objek yang serupa.

Dalam bahasa pemrograman C++ dimungkinkan untuk mendaftarkan banyak base class pada satu derived class. Cara penulisannya sama seperti kita

mendaftarkan satu base class pada satu derived class, hanya daftar dari base class akan ditulis secara sejajar dan dipisahkan dengan tanda koma “,”.

Ada beberapa karakteristik yang di-share oleh semua kelompok. Relasi antar kelas-kelas ini disebut dengan relasi “is-a”. Dalam setiap kasus, objek yang dikelompokkan bersama dalam satu sub-kategori merupakan anggota dari kategori yang lebih umum.

Objek yang dikelompokkan dalam satu kelas men-share sekumpulan atribut dan perilaku. Jadi, seluruh objek kendaraan bermotor memiliki sekumpulan atribut dan perilaku yang juga dimiliki oleh objek dari mobil. Keterkaitan antar kelas dalam relasi “is-a” berasal dari kenyataan bahwa sub kelas memiliki atribut dan perilaku yang dimiliki kelas induk, ditambah atribut dan perilaku yang dimiliki oleh sub kelas tersebut.

Superclass (“kelas dasar” atau “kelas induk”) merupakan kelas yang lebih general dalam relasi “is-a”. Subclass (“kelas turunan” atau “kelas anak”) merupakan kelas yang lebih spesifik dalam relasi “is-a”. Objek yang dikelompokkan dalam sub kelas memiliki atribut dan perilaku kelas induk, dan juga atribut dan perilaku tambahan. (Jadi, kumpulan atribut dan perilaku sub kelas lebih besar dari super kelas-nya). Relasi “is-a” antar superclass dan subclasses-nya disebut dengan pewarisan atau inheritance.

Subclass “mewarisi” suatu superclass (atau juga bisa dikatakan sebuah subclass “turunan dari” suatu superclass) karena reusabilitas Perangkat Lunak, membuat kelas baru (kelas turunan) dari kelas yang sudah ada (kelas dasar), kelas turunan mewarisi kelas induk yang mendapatkan data dan perilaku, merupakan bentuk spesial dari kelas induk, dan diperluas dengan perilaku tambahan,

Pewarisan ada dua jenis yaitu pewarisan tunggal dan pewarisan jamak. Pada protected access, protected members dapat diakses oleh member kelas dasar, friend kelas dasar, member kelas turunan, dan friend kelas turunan. Kelas turunan dapat merujuk/mengakses langsung public dan protected data member kelas induk dengan menggunakan nama atribut yang diakses.

## BAB II

### PEMBAHASAN

#### 2.1. Program pertama

```
/*  
    Nama      : Abigael Gerry Pratama  
    NIM       : 193020503021  
*/  
  
#include <iostream>  
#include <conio.h>  
#include <cstring>  
  
using namespace std;  
  
//Kelas Dasar "Buah"  
class Buah  
{  
protected:  
    char nama[20];  
    char jenis[20];  
  
public:  
    //konsruktor  
    //di sini letak modifikasinya  
    Buah(char *nm, char *jns)  
    {  
        strcpy (nama, nm);  
        strcpy (jenis, jns);  
    }  
}
```

```

//di sini letak modifikasinya
void infoBuah()
{
    cout << endl;
    cout << " Nama\t\t: " << nama << endl;
    cout << " Jenis\t\t: " << jenis << endl;
}
};

class Jenis : public Buah
{
protected:
    char ciri_khusus[30];

public:
    //konstruktor
    //di sini letak modifikasinya
    Jenis(char* nm, char* jns, char* cr_k) : Buah(nm, jns)
    {
        strcpy (ciri_khusus, cr_k);
    }

    //di sini letak modifikasinya
    void infoJenis()
    {
        cout << " Ciri Khusus\t: " << ciri_khusus << endl;
        cout << endl;
    }
};

int main()

```



```

{
    //menciptakan objek "j1" sekaligus melakukan inisialisasi melalui
    Konstruktor dari Kelas Jenis dan Konstruktor dari Kelas Buah
    Jenis j1 ("Rambutan", "Antalagi", "Besar dan Manis");
    cout << "\n INFO SALAH SATU BUAH RAMBUTAN " << endl;

    j1.infoBuah();
    j1.infoJenis();

    system("pause");

    return 0;
}

```

## Penjelasan

```

include <iostream>
#include <conio.h>
#include <cstring>

using namespace std;

//Kelas Dasar "Buah"
class Buah
{
protected:
    char nama[20];
    char jenis[20];

public:
    //konstruktor
    //di sini letak modifikasinya

```

```

    Buah(char *nm, char *jns)
    {
        strcpy (nama, nm);
        strcpy (jenis, jns);
    }

    //di sini letak modifikasinya
    void infoBuah()
    {
        cout << endl;
        cout << " Nama\t\t: " << nama << endl;
        cout << " Jenis\t\t: " << jenis << endl;
    }
};

```

Pernyataan yang diawali dengan tanda (#) merupakan pernyataan untuk menyertakan preprocessor. Pernyataan seperti `#include <iostream>` berarti memerintahkan kompiler untuk menyertakan file header `iostream`. Dalam file header ini, terdapat beberapa fungsi standar yang dipakai dalam proses input dan output seperti perintah `cout`, `cin`, dan `endl`. Selain itu, program di atas juga menggunakan preprocessor `#include <conio.h>` untuk membaca perintah `system` (“`cls`”), dan juga `#include <cstring>` untuk membaca perintah `strcpy`.

Pada kelas buah, nama buah ditulis dengan sintaks `class Buah`. Tampak bahwa terdapat dua buah perubah akses (anggota kelas) yang digunakan pada kelas ini yaitu `protected` dan `public`. Yang termasuk perubah akses `protected` yaitu variabel `nama` (bertipe data karakter (`char`) dengan panjang maksimum 20 karakter) dan `jenis` (maksimum 20). Yang termasuk perubah akses `public` yaitu metode `Buah` dan `infoBuah` (bertipe `void`). Suatu fungsi yang tidak menghasilkan suatu nilai apapun maka bagian `return type`

ini diganti dengan void. Definisi atas metode-metode tersebut akan dipaparkan di bagian bawah.

```
class Jenis : public Buah
{
protected:
    char ciri_khusus[30];

public:
    //konstruktor
    //di sini letak modifikasinya
    Jenis(char* nm, char* jns, char* cr_k) : Buah(nm, jns)
    {
        strcpy (ciri_khusus, cr_k);
    }

    //di sini letak modifikasinya
    void infoJenis()
    {
        cout << " Ciri Khusus\t: " << ciri_khusus << endl;
        cout << endl;
    }
};
```

Pada kelas Jenis di atas, nama kelas ditulis dengan sintaks class Jenis, yang menunjukkan bahwa Jenis adalah nama suatu kelas. Pernyataan class Jenis : public Buah menunjukkan bahwa pada kelas ini akan digunakan data dan fungsi yang dapat diakses secara public pada kelas buah (kelas induk). Tampak bahwa terdapat dua buah perubah akses (anggota kelas) yang digunakan pada kelas ini yaitu protected dan public. Yang termasuk perubah akses protected yaitu variabel bagian ciri\_khusus (bertipe data char). Yang termasuk perubah akses public yaitu metode Jenis dan infoJenis

(bertipe void). Suatu fungsi yang tidak menghasilkan suatu nilai apapun maka bagian return type ini diganti dengan void. Definisi atas metode-metode tersebut akan dipaparkan di bagian bawah.

```
int main()
{
    //menciptakan objek "j1" sekaligus melakukan inisialisasi melalui
    Kontruktor dari Kelas Jenis dan Kontruktor dari Kelas Buah
    Jenis j1 ("Rambutan", "Antalagi", "Besar dan Manis");
    cout << "\n INFO SALAH SATU BUAH RAMBUTAN " << endl;

    j1.infoBuah();
    j1.infoJenis();

    system("pause");

    return 0;
}
```

Penulisan listing program untuk bagian program utama selalu diawali dengan fungsi int main (). Fungsi dari program utama di atas yaitu untuk menentukan prosedur dan langkah eksekusi terhadap masing-masing kelas yang telah dibuat. Pada program utama di atas ternyata prosedur yang digunakan untuk menampilkan output ke layar yaitu tanpa meminta input dari user, artinya obyek yang dimasukkan sudah ditentukan terlebih dahulu dalam program utama. Berikut ini adalah hasil output dari program yang telah dibuat.

```
D:\Abigael Gerry\Teknik Informatika UPR_PRAKTIKUM\Se

INFO SALAH SATU BUAH RAMBUTAN

Nama      : Rambutan
Jenis     : Antalagi
Ciri Khusus : Besar dan Manis

Press any key to continue . . .
```

*JR. tnguh*

Gambar 2.1 Program Satu

Ini adalah contoh output yang keluar setelah program dijalankan.

## 2.2. Program kedua

```
/*
    Nama      : Abigael Gerry Pratama
    NIM       : 193020503021
*/

#include <iostream>
#include <conio.h>
#include <string>

using namespace std;

class Buah{
protected:
    string nama, harga;
public:
    void infoBuah(){
        cout << endl;
        cout << " Info salah satu jenis buah\n" << endl;
    }
};
```

```
class Mangga : public Buah{
private:
    string tipeBuah;
public:
    Mangga(string nm, string hg, string tp){
        nama = nm;
        tipeBuah = tp;
        harga = hg;
    }
    void infoMangga()
    {
        cout << " Nama\t\t: " << nama << endl;
        cout << " Tipe buah\t: " << tipeBuah << endl;
        cout << " Harga\t\t: Rp" << harga << endl << endl;
    }
};
```

```
class Apple : public Buah{
private:
    string tipeBuah;
public:
    Apple(string nm, string hg, string tp)
    {
        nama = nm;
        tipeBuah = tp;
        harga = hg;
    }
    void infoApple()
    {
        cout << " Nama\t\t: " << nama << endl;
```

```

        cout << " Tipe buah\t: " << tipeBuah << endl;
        cout << " Harga\t\t: Rp" << harga << endl << endl;
    }
};

int main()
{
    Mangga    mangga1("Mangga",    "15.000",    "Manalagi"),
mangga2("Mangga" ,"50.000", "Golek");
    Apple    apel1("Apel","55.000",  "Anna"),    apel2("Apel","65.000",
"Fuji");
    mangga1.infoBuah();
    mangga1.infoMangga();
    mangga2.infoMangga();
    apel1.infoApple();
    apel2.infoApple();

        system("pause");

    return 0;
}

```

## Penjelasan

```

#include <iostream>
#include <conio.h>
#include <string>

using namespace std;

class Buah{

```

```
protected:
    string nama, harga;
public:
    void infoBuah(){
        cout << endl;
        cout << " Info salah satu jenis buah\n" << endl;
    }
};
```

Program ini menggunakan pernyataan `#include <iostream>` untuk menggunakan perintah `cout`, `cin`, dan `endl`. Lalu pernyataan `#include <conio.h>` digunakan untuk perintah `system("cls")`. Dan pernyataan yang terakhir yaitu `#include <string>` yang digunakan untuk menggunakan tipe data `string`.

Kelas utama pada program ini yaitu class `Buah`, yang terdapat dua anggota kelas yaitu `protected` dan `public`. Nama dan harga (bertipe data `string`) termasuk akses `protected`, dan metode `infoBuah` merupakan akses `public` (bertipe `void`).

```
class Mangga : public Buah{
private:
    string tipeBuah;
public:
    Mangga(string nm, string hg, string tp){
        nama = nm;
        tipeBuah = tp;
        harga = hg;
    }
    void infoMangga()
    {
```



```

        cout << " Nama\t\t: " << nama << endl;
        cout << " Tipe buah\t: " << tipeBuah << endl;
        cout << " Harga\t\t: Rp" << harga << endl << endl;
    }
};

```

Ini adalah class Mangga : public Buah yang merupakan kelas ini akan digunakan data dan fungsi yang akan diakses secara public dari kelas utama (class Buah). Anggota kelas ini yaitu private dan public. Yang termasuk akses private yaitu tipeBuah (bertipe data string), dan yang termasuk akses public yaitu Mangga dan infoMangga (bertipe void). Pernyataan `cout << " Nama\t\t: " << nama << endl;` berfungsi untuk menampilkan nama dari data yang dimasukan untuk kelas mangga.

```

class Apple : public Buah{
private:
    string tipeBuah;
public:
    Apple(string nm, string hg, string tp)
    {
        nama = nm;
        tipeBuah = tp;
        harga = hg;
    }
    void infoApple()
    {
        cout << " Nama\t\t: " << nama << endl;
        cout << " Tipe buah\t: " << tipeBuah << endl;
        cout << " Harga\t\t: Rp" << harga << endl << endl;
    }
};

```

Pada class Apple : public Buah hampir sama dengan class Mangga, yang merupakan kelas ini akan digunakan data dan fungsi yang akan diakses secara public dari kelas utama (class Buah). Anggota kelas ini yaitu private dan public. Yang termasuk akses private yaitu tipeBuah (bertipe data string), dan yang termasuk akses public yaitu Apple dan infoApple (bertipe void). Pernyataan cout << " Nama\t\t: " << nama << endl; berfungsi untuk menampilkan nama dari data yang dimasukan untuk kelas apple.

```
int main()
{
    Mangga    mangga1("Mangga",    "15.000",    "Manalagi"),
mangga2("Mangga" ,"50.000", "Golek");
    Apple  apel1("Apel","55.000", "Anna"),  apel2("Apel","65.000",
"Fuji");
    mangga1.infoBuah();
    mangga1.infoMangga();
    mangga2.infoMangga();
    apel1.infoApple();
    apel2.infoApple();

    system("pause");

    return 0;
}
```

Fungsi dari program utama di atas yaitu untuk menentukan prosedur dan langkah eksekusi terhadap masing-masing kelas yang telah dibuat. Pada program utama di atas ternyata prosedur yang digunakan untuk menampilkan output ke layar yaitu tanpa meminta input dari user, artinya obyek yang dimasukkan sudah ditentukan terlebih dahulu dalam program utama. Tampak bahwa program utama tersebut akan menampilkan obyek-obyek yang memuat metode-metode dalam kelas Buah. Berikut adalah hasil output setelah program dibuat.

```
D:\Abigael Gerry\Teknik Informatika UPR\PRAKTIKUM

Info salah satu jenis buah

Nama      : Mangga
Tipe buah : Manalagi
Harga     : Rp15.000

Nama      : Mangga
Tipe buah : Golek
Harga     : Rp50.000

Nama      : Apel
Tipe buah : Anna
Harga     : Rp55.000

Nama      : Apel
Tipe buah : Fuji
Harga     : Rp65.000
```

## Program 2.2 Program Dua

Berikut outputan sesuai dengan program yang dijalankan.

### **BAB III**

### **KESIMPULAN**

Pada modul ini kita diajarkan bahwa pewarisan adalah salah satu konsep pada object oriented programming, yang mengadopsi konsep pewarisan yang dimiliki oleh object pada dunia nyata. Dan juga agar bisa menggunakan konsep pewarisan agar mempermudah kita dalam membuat dan perawatan sebuah program, dengan menghilangkan “penggunaan kode berulang” saat pembuatan banyak class yang serupa. Pemberian access specifier pada daftar base class dalam mendirikan derived class dapat berupa public, protected atau private, masing-masing akan memberikan efek berbeda pada derived class tersebut.



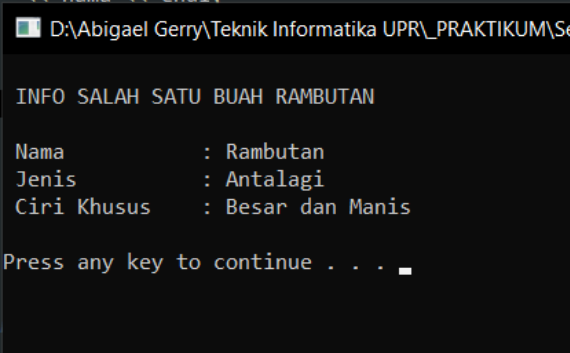
## DAFTAR PUSTAKA

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

Sinauarduino. 2016. *Inheritance Pada Classes C++*. [https://www.sinauarduino.co  
m/artikel/inheritance-pada-classes-cpp/](https://www.sinauarduino.com/artikel/inheritance-pada-classes-cpp/) (diakses pada Minggu, 12 April  
2020)

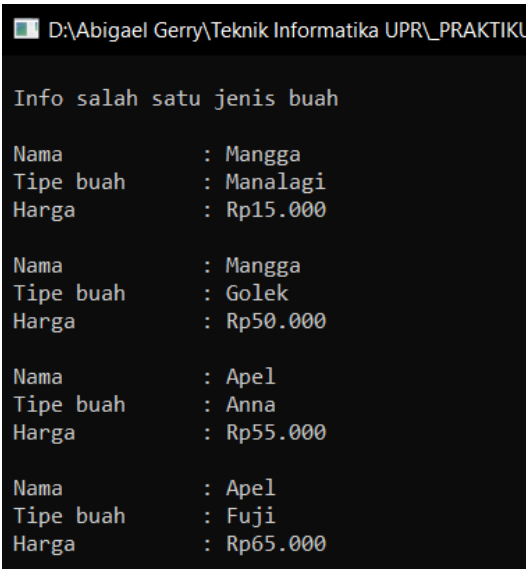
PT Proweb Indonesia. 2015. *Inheritance pada C++*. [https://www.proweb.co  
.id/articles/mobile\\_development/inheritance\\_cplusplus.html](https://www.proweb.co.id/articles/mobile_development/inheritance_cplusplus.html) (diakses pada  
Minggu, 12 April 2020)

## LAMPIRAN



```
D:\Abigael Gerry\Teknik Informatika UPR_PRAKTIKUM\Se  
INFO SALAH SATU BUAH RAMBUTAN  
Nama      : Rambutan  
Jenis     : Antalagi  
Ciri Khusus : Besar dan Manis  
Press any key to continue . . .
```

Gambar 2.1 Program Satu



```
D:\Abigael Gerry\Teknik Informatika UPR_PRAKTIKU  
Info salah satu jenis buah  
Nama      : Mangga  
Tipe buah : Manalagi  
Harga     : Rp15.000  
  
Nama      : Mangga  
Tipe buah : Golek  
Harga     : Rp50.000  
  
Nama      : Apel  
Tipe buah : Anna  
Harga     : Rp55.000  
  
Nama      : Apel  
Tipe buah : Fuji  
Harga     : Rp65.000
```

Gambar 2.2 Program Dua