

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : THOMAS NURHUDA  
NIM : 193010503002  
KELAS : A  
MODUL : II (PEWARISAN)**

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

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



Nama : Thomas Nurhuda  
NIM : 193010503002  
Kelas : A  
Modul : II (Pewarisan)

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	6
BAB II Pembahasan	60	50
BAB III Kesimpulan	20	13
Daftar Pustaka	5	5
Lampiran	5	4
Jumlah	100	78

Penilai  
Asisten Praktikum

Diana

## **BAB I**

### **TUJUAN DAN LANDASAN TEORI**

#### **A. Tujuan**

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

#### **B. Landasan Teori**

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). 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.

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. Contohnya adalah seperti di bawah ini.

1. Mobil adalah (“is-a”) kendaraan bermotor.
2. Truk adalah (“is-a”) kendaraan bermotor.
3. Sepeda Motor adalah (“is-a”) kendaraan bermotor.

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 yang diakses.

## BAB II

### PEMBAHASAN

#### 1. Program Class Buah Durian

Perintah pertama yang diberikan kali ini adalah membuat sebuah program kelas buah yang mampu mewariskan kelas dari kelas induk sebelumnya menggunakan sifat dari pewarisan (inheritance). Berikut adalah pendeklarasian program yang akan dibuat.

*Tabel 2.1.1. Program Class Durian*

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

class durian{
    public:
        durian(char *k,char *r,char *b);
        void info();

    protected:
        char kulit[30];
        char rasa [30];
        char biji [30];
};

durian::durian(char *k, char *r, char *b){
    strcpy(kulit,k);
    strcpy(rasa,r);
    strcpy(biji,b);
}
```

```

void durian::info() {
    cout<<"Class Buah Dijalankan :\\n"
    <<"=====\\n\\n";
}

class harga:public durian {
    protected:
        char status[30];

    public:
        harga(char *k, char *r, char *b, char *s);//konstruktor
        void info();
};

harga::harga(char *k,char *r, char *b, char *s):durian(k,r,b) {
    strcpy(status,s);
}

void harga::info() {
    durian::info();
    cout<<"Kulit Buah Durian : "<<kulit<<endl;
    cout<<"Rasa Buah Durian : "<<rasa<<endl;
    cout<<"Jenis Biji Durian : "<<biji<<endl;
    cout<<"Harga Buah Durian : Rp."<<status<<endl<<endl;
}

int main(){
    harga buah1("Berduri", "Manis", "Dikotil", "30000");

    buah1.info();

    system("pause");
    return 0;
}

```

Dalam membuat sebuah program, seorang programmer harus membuat File Header terlebih dahulu, dimana nantinya File Header ini digunakan untuk memanggil library-library yang ada sehingga suatu fungsi dapat digunakan secara baik dan benar. `#include` digunakan untuk mendeklarasikan File Header didalam bahasa pemrograman C++. File Header yang digunakan pada program diatas adalah `<iostream>` yang digunakan untuk menampilkan perintah `cin`, `cout` dan `endl`. Kemudian `<string.h>` digunakan untuk menampilkan perintah `strcpy`, `strcmp`, dan lain-lain. Dan kode `using namespace std;` berfungsi untuk memanggil Class/Object/Fungsi yang terdapat didalam namespace tersebut.

Pada program diatas, setelah pendeklarasian file header, kode program selanjutnya adalah pendeklarasian kelas dari Durian sebagai berikut.

*Tabel 2.1.2 Deklarasi Class Durian*

```
class durian{  
    public:  
        durian(char *k,char *r,char *b);  
        void info();  
  
    protected:  
        char kulit[30];  
        char rasa [30];  
        char biji [30];  
};
```

Sebagai pemrograman yang berbasis objek, sebuah program tentunya memiliki fitur yang dinamakan Class atau kelas. Sebuah kelas bisa dikatakan sebagai pengembangan dari struktur data. Perbedaan yang mencolok antara struktur data dan kelas adalah pada kelas, terdapat member-member berupa fungsi dan variabel, sedangkan pada struktur data hanya dapat menampung variabel saja. Didalam sebuah kelas terdapat anggota yang disebut member dan untuk mengakses member ini, maka diperlukan penentu akses.

Penentu akses yang digunakan terbagi menjadi 3 kategori, yaitu Private, Public serta Protected. Pada program diatas penentu akses yang digunakan adalah Public dan Protected. Public merupakan hak akses maksimum, dimana fungsi dan kelas apapun selama masih dalam satu program dapat mengakses member yang bernaung di bawah label Public. Fungsi yang terdapat pada akses Public ini adalah fungsi “durian(char \*k, char \*r, char \*b);” yang berguna sebagai Konstruktor yaitu fungsi khusus sebuah kelas yang bertanggung jawab dalam membuat objek baru. Selain itu, konstruktor juga berfungsi menyediakan untuk menginisialisasi object yang dibuat dan digunakan untuk memastikan data member yang bernilai benar. Konstruktor ini memiliki ciri khusus yaitu tidak mempunyai tipe data dan void. Dan fungsi yang kedua pada akses Public ini adalah “void info();” yang nantinya akan digunakan untuk menampilkan data pada output program. Protected merupakan hak akses yang dimana member-member dibawah label ini dapat diakses oleh kelas turunan dari kelas tersebut. Member-member yang ada pada akses Protected ini adalah “char kulit[30];”, “char rasa[30];” dan “char biji[30]. Semua member pada akses Protected adalah Array dengan tipe data char yang masing-masing memiliki nilai maksimum 30.

Selanjutnya terdapat kode peogram :

*Tabel 2.1.3 Deklarasi Variabel Konstruktor Class Durian*

```
durian::durian(char *k, char *r, char *b){
    strcpy(kulit,k);
    strcpy(rasa,r);
    strcpy(biji,b);
}
```

Kode program diatas adalah pendeklarasian data dari konstruktor durian pada kelas Durian yang nanti nya digunakan pada saat ingin memasukkan data didalam program sehingga data bisa terbaca oleh program nantinya. Pada bagian ini lah masing-masing tipe data diberi variabel sebagai alamat pemanggilan data ketika ingin menginputkan data. Salah satu contohnya adalah “strcpy(kulit,k);”.



“strcpy” digunakan untuk menyalin string asal ke variabel tujuan dengan syarat string tujuan harus mempunyai tipe data dan ukuran yang sama dengan string asal. “(kulit,k);” inilah yang merupakan variabel dari pointer k yang berfungsi sebagai alamat atau memori pemanggilan saat kita menginputkan data. Program berikutnya adalah fungsi void info sebagai berikut.

*Tabel 2.1.4. Info Class Durian*

```
void durian::info() {  
    cout<<"Class Buah Dijalankan : \n"  
    <<"===== \n \n";  
}
```

Fungsi kode program diatas adalah untuk menampilkan data dari kelas durian pada output program. Selanjutnya adalah pendeklarasian kelas turunan dari kelas induk. Kelas turunan adalah kelas yang mewarisi kelas sebelumnya (kelas induk) dimana kelas ini dapat menggunakan property-property dari parentnya. Pada program diatas pendeklarasiannya adalah sebagai berikut.

*Tabel 2.1.5. Deklarasi Class Turunan*

```
class harga:public durian {  
    protected:  
        char status[30];  
  
    public:  
        harga(char *k, char *r, char *b, char *s);//konstruktor  
        void info();  
};
```

Pada tabel diatas, kelas turunan dari kelas induk bernama Class Harga. Dimana kelas ini mewarisi akses public dari Class Durian. Kelas turunan juga memiliki hak akses, karena kelas diatas menurunkan hak akses public dari kelas

induknya, maka member pada akses public kelas induk harus dimiliki oleh kelas turunan. Kelas turunan dapat menambahkan variabel data pada hak akses public, contohnya “char \*s” yang dapat dilihat pada konstruktor kelas turunan diatas. Untuk hak akses Protected pada kelas turunan hanyalah memuat variabel data tambahan seperti “char status[30]” karena variabel sebelumnya bisa diakses dari protected Class Durian.

Kode program berikutnya adalah pendeklarasian data dari konstruktor harga pada Class Harga yang nanti nya digunakan pada saat ingin memasukkan data didalam program sehingga data bisa terbaca oleh program nantinya. Pada bagian ini lah masing-masing tipe data diberi variabel sebagai alamat pemanggilan data ketika ingin menginputkan data. Karena ini merupakan kelas turunan, maka yang dideklarasikan hanyalah variabel tambahannya saja. Pendeklarasiannya adalah sebagai berikut.

*Tabel 2.1.6. Deklarasi Variabel Konstruktor Class Turunan*

```
harga::harga(char *k,char *r, char *b, char *s):durian(k,r,b) {  
    strcpy(status,s);  
}
```

Kemudian untuk menampilkan info pada output program kelas turunan dapat dilakukan dengan mendeklarasikan kode program sebagai berikut.

*Tabel 2.1.7. Info Class Turunan*

```
void harga::info() {  
    durian::info();  
    cout<<"Kulit Buah Durian : "<<kulit<<endl;  
    cout<<"Rasa Buah Durian : "<<rasa<<endl;  
    cout<<"Jenis Biji Durian : "<<biji<<endl;  
    cout<<"Harga Buah Durian : Rp."<<status<<endl<<endl;  
}
```

Data yang akan ditampilkan pada output program nantinya adalah Kulit, Rasa, Jenis Biji serta Harga Buah Durian. Ambil salah satu kode program yaitu `cout<<"Kulit Buah Durian: "<<kulit<<endl;`. "kulit" pada program tersebut adalah variabel yang digunakan untuk menampilkan data dari memori variabel kulit yang dideklarasikan sebelumnya.

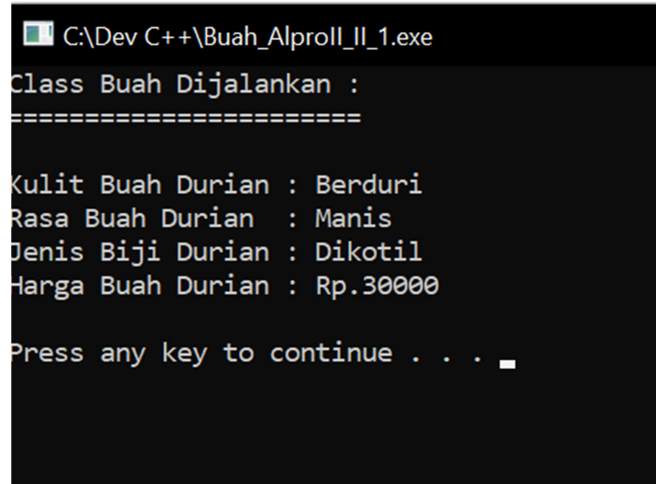
Kode program berikutnya adalah Fungsi main atau `Int main()` merupakan fungsi utama dalam program. Fungsi ini akan dieksekusi pertama kali saat program dijalankan. Oleh karena itu, kita harus menuliskan logika program di dalam fungsi ini. Maksud dari kata `Int` didepan main adalah tipe data yang akan dikembalikan. Maka didalam fungsi `main()`, wajib kita sertakan kode `return 0` yang berarti fungsi main akan mengembalikan nilai 0 setelah dieksekusi. Pada program C++ terdapat Blok kode yang merupakan kumpulan statement atau ekspresi yang dibungkus dengan kurung kurawal atau `{...}`.

*Tabel 2.1.8. Fungsi Utama*

```
int main(){  
    harga buah1("Berduri", "Manis", "Dikotil", "30000");  
  
    buah1.info();  
  
    system("pause");  
    return 0;  
}
```

Fungsi main diatas akan menampilkan data dari kelas turunan (Class Harga). Objek pada program diatas adalah buah1. Data yang terdapat pada objek ini akan ditampilkan pada output berdasarkan alamat memori yang telah dideklarasikan sebelumnya. Agar data bisa ditampilkan pada output program, maka diperlukan kode "buah1.info();". Kode system ("pause") untuk menghentikan proses eksekusi sampai user menekan sembarang tombol. Dan kode `return 0;` digunakan untuk mengakhiri sebuah program.

Ketika program diatas dijalankan, maka output yang akan ditampilkan adalah sebagai berikut.

A screenshot of a Windows command prompt window. The title bar shows the file path 'C:\Dev C++\Buah\_AlproII\_II\_1.exe'. The output text is as follows:

```
Class Buah Dijalankan :  
=====  
  
Kulit Buah Durian : Berduri  
Rasa Buah Durian : Manis  
Denis Biji Durian : Dikotil  
Harga Buah Durian : Rp.30000  
  
Press any key to continue . . .
```

*Gambar 2.1. Output Class Buah Durian*

## 2. Program Class Buah Pisang

Perintah kedua yang diberikan kali ini hamper sama dengan perintah sebelumnya, yaitu membuat sebuah program kelas buah yang mampu mewariskan kelas dari kelas induk sebelumnya menggunakan sifat dari pewarisan (inheritance). Berikut adalah pendeklarasian program yang akan dibuat.

*Tabel 2.2.1. Program Class Pisang*

```
#include<iostream>  
#include<string.h>  
using namespace std;  
  
class Pisang{  
    public:  
        Pisang(char *k,char *r,char *b);  
        void info();  
    protected:
```

```

        char kulit[30];
        char rasa [30];
        char biji [30];
};

Pisang::Pisang(char *k, char *r, char *b){
    strcpy(kulit,k);
    strcpy(rasa,r);
    strcpy(biji,b);
}

void Pisang::info() {
    cout<<"Class Buah Dijalankan : \n"
    <<"===== \n \n";
}

class harga:public Pisang {
    public:
        harga(char *k, char *r, char *b, char *s);//konstruktor
        void info();

    protected:
        char status[30];
};

harga::harga(char *k,char *r, char *b, char *s):Pisang(k,r,b) {
    strcpy(status,s);
}

void harga::info() {
    Pisang::info();
}

```

```

        cout<<"Kulit Buah Pisang : "<<kulit<<endl;
        cout<<"Rasa Buah Pisang : "<<rasa<<endl;
        cout<<"Jenis Biji Pisang : "<<biji<<endl;
        cout<<"Harga Buah Pisang : Rp."<<status<<endl<<endl;
    }

    int main(){
        harga buah1("Berwarna Kuning", "Manis", "Monokotil", "15000");

        buah1.info();

        system("pause");
        return 0;
    }

```

Program diatas sebenarnya sama seperti program pertama, dimana untuk membuat program kita harus membuat file header terlebih dahulu. File Header ini digunakan untuk memanggil library-library yang ada sehingga suatu fungsi dapat digunakan secara baik dan benar. #include digunakan untuk mendeklarasikan File Header didalam bahasa pemrograman C++. Program kedua ini juga menggunakan File Header <iostream> yang digunakan untuk menampilkan perintah cin,cout dan endl. Kemudian <string.h> digunakan untuk menampilkan perintah strcpy,strcmp,dan lain-lain. Dan kode using namespace std; berfungsi untuk memanggil Class/Object/Fungsi yang terdapat didalam namespace tersebut.

Pada program diatas, setelah pendeklarasian file header, kode program selanjutnya adalah pendeklarasian kelas dari Pisang sebagai berikut.

*Tabel 2.2.2 Deklarasi Class Pisang*

```
class Pisang{  
    public:  
        Pisang(char *k,char *r,char *b);  
        void info();  
  
    protected:  
        char kulit[30];  
        char rasa [30];  
        char biji [30];  
};
```

Sebagai pemrograman yang berbasis objek, sebuah program tentunya memiliki fitur yang dinamakan Class atau kelas. Sebuah kelas bisa dikatakan sebagai pengembangan dari struktur data. Perbedaan yang mencolok antara struktur data dan kelas adalah pada kelas, terdapat member-member berupa fungsi dan variabel, sedangkan pada struktur data hanya dapat menampung variabel saja. Didalam sebuah kelas terdapat anggota yang disebut member dan untuk mengakses member ini, maka diperlukan penentu akses.

Penentu akses yang digunakan terbagi menjadi 3 kategori, yaitu Private, Public serta Protected. Pada program diatas penentu akses yang digunakan adalah Public dan Protected. Public merupakan hak akses maksimum, dimana fungsi dan kelas apapun selama masih dalam satu program dapat mengakses member yang bernaung di bawah label Public. Fungsi yang terdapat pada akses Public ini adalah fungsi “Pisang(char \*k, char \*r, char \*b);” yang berguna sebagai Konstruktor yaitu fungsi khusus sebuah kelas yang bertanggung jawab dalam membuat objek baru. Selain itu, konstruktor juga berfungsi menyediakan untuk menginisialisasi object yang dibuat dan digunakan untuk memastikan data member yang bernilai benar. Konstruktor ini memiliki ciri khusus yaitu tidak mempunyai tipe data dan void. Dan fungsi yang kedua pada akses Public ini adalah “void info();” yang nantinya akan digunakan untuk menampilkan data pada output program.

Protected merupakan hak akses yang dimana member-member dibawah label ini dapat diakses oleh kelas turunan dari kelas tersebut. Member-member yang ada pada akses Protected ini adalah “char kulit[30];”, “char rasa[30];” dan “char biji[30]. Semua member pada akses Protected adalah Array dengan tipe data char yang masing-masing memiliki nilai maksimum 30.

Selanjutnya terdapat kode program :

*Tabel 2.2.3 Deklarasi Variabel Konstruktor Class Pisang*

```
Pisang::Pisang(char *k, char *r, char *b){  
    strcpy(kulit,k);  
    strcpy(rasa,r);  
    strcpy(biji,b);  
}
```

Kode program diatas adalah pendeklarasian data dari konstruktor durian pada Class Pisang yang nanti nya digunakan pada saat ingin memasukkan data didalam program sehingga data bisa terbaca oleh program nantinya. Pada bagian ini lah masing-masing tipe data diberi variabel sebagai alamat pemanggilan data ketika ingin menginputkan data. Salah satu contohnya adalah “strcpy(kulit,k);”. “strcpy” digunakan untuk menyalin string asal ke variabel tujuan dengan syarat string tujuan harus mempunyai tipe data dan ukuran yang sama dengan string asal. “(kulit,k);” inilah yang merupakan variabel dari pointer k yang berfungsi sebagai alamat atau memori pemanggilan saat kita menginputkan data. Program berikutnya adalah fungsi void info sebagai berikut.

*Tabel 2.2.4. Info Class Pisang*

```
void Pisang::info() {  
    cout<<"Class Buah Dijalankan : \n"  
    <<"===== \n \n";  
}
```



Fungsi kode program diatas adalah untuk menampilkan data dari kelas durian pada output program. Selanjutnya adalah pendeklarasian kelas turunan dari kelas induk. Kelas turunan adalah kelas yang mewarisi kelas sebelumnya (kelas induk) dimana kelas ini dapat menggunakan property-property dari parentnya. Pada program diatas pendeklarasiannya adalah sebagai berikut.

*Tabel 2.2.5. Deklarasi Class Turunan*

```
class harga:public Pisang {  
    public:  
        harga(char *k, char *r, char *b, char *s); //konstruktor  
        void info();  
  
    protected:  
        char status[30];  
};
```

Pada tabel diatas, kelas turunan dari kelas induk bernama Class Harga. Dimana kelas ini mewarisi akses public dari Class Pisang. Kelas turunan dapat menambahkan variabel data pada hak akses public, contohnya “char \*s” yang dapat dilihat pada konstruktor kelas turunan diatas. Untuk hak akses Protected pada kelas turunan hanyalah memuat variabel data tambahan seperti “char status[30]” karena variabel sebelumnya bisa diakses dari protected Class Pisang.

Kode program berikutnya adalah pendeklarasian data dari konstruktor harga pada Class Harga yang nanti nya digunakan pada saat ingin memasukkan data didalam program sehingga data bisa terbaca oleh program nantinya. Pada bagian ini lah masing-masing tipe data diberi variabel sebagai alamat pemanggilan data ketika ingin menginputkan data. Karena ini merupakan kelas turunan, maka yang dideklarasikan hanyalah variabel tambahannya saja. Pendeklarasiannya adalah sebagai berikut.

*Tabel 2.2.6. Deklarasi Variabel Konstruktur Class Turunan*

```
harga::harga(char *k, char *r, char *b, char *s):Pisang(k,r,b) {  
    strcpy(status,s);  
}
```

Kemudian untuk menampilkan info pada output program kelas turunan dapat dilakukan dengan mendeklarasikan kode program sebagai berikut.

*Tabel 2.2.7. Info Class Turunan*

```
void harga::info() {  
    Pisang::info();  
    cout<<"Kulit Buah Pisang : "<<kulit<<endl;  
    cout<<"Rasa Buah Pisang : "<<rasa<<endl;  
    cout<<"Jenis Biji Pisang : "<<biji<<endl;  
    cout<<"Harga Buah Pisang : Rp."<<status<<endl<<endl;  
}
```

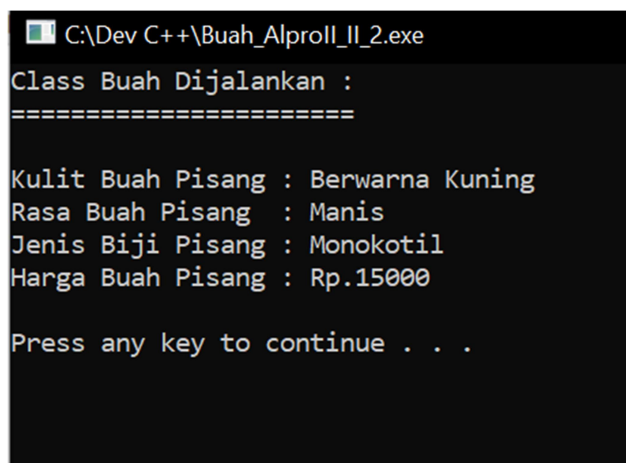
Data yang akan ditampilkan pada output program nantinya adalah Kulit, Rasa, Jenis Biji serta Harga Buah Pisang. Ambil salah satu kode program yaitu `cout<<"Kulit Buah Pisang : "<<kulit<<endl;`. "kulit" pada program tersebut adalah variabel yang digunakan untuk menampilkan data dari memori variabel kulit yang dideklarasikan sebelumnya.

Kode program berikutnya adalah Fungsi main atau `Int main()` merupakan fungsi utama dalam program. Fungsi ini akan dieksekusi pertama kali saat program dijalankan. Oleh karena itu, kita harus menuliskan logika program di dalam fungsi ini. Maksud dari kata `Int` didepan `main` adalah tipe data yang akan dikembalikan. Maka didalam fungsi `main()`, wajib kita sertakan kode `return 0` yang berarti fungsi `main` akan mengembalikan nilai 0 setelah dieksekusi. Pada program C++ terdapat Blok kode yang merupakan kumpulan statement atau ekspresi yang dibungkus dengan kurung kurawal atau `{...}`.

*Tabel 2.2.8. Fungsi Utama*

```
int main(){  
    harga buah1("Berwarna Kuning", "Manis", "Monokotil",  
    "15000");  
  
    buah1.info();  
  
    system("pause");  
    return 0;  
}
```

Fungsi main diatas akan menampilkan data dari kelas turunan (Class Harga). Objek pada program diatas adalah buah1. Data yang terdapat pada objek ini akan ditampilkan pada output berdasarkan alamat memori yang telah dideklarasikan sebelumnya. Agar data bisa ditampilkan pada output program, maka diperlukan kode “buah1.info();”. Kode system (“pause”) untuk menghentikan proses eksekusi sampai user menekan sembarang tombol. Dan kode return 0; digunakan untuk mengakhiri sebuah program. Output yang ditampilkan ketika program dijalankan adalah:



```
C:\Dev C++\Buah_AlproII_2.exe  
Class Buah Dijalankan :  
=====  
  
Kulit Buah Pisang : Berwarna Kuning  
Rasa Buah Pisang : Manis  
Jenis Biji Pisang : Monokotil  
Harga Buah Pisang : Rp.15000  
  
Press any key to continue . . .
```

*Gambar 2.2. Output Class Buah Pisang*

### **BAB III**

### **KESIMPULAN**

Sebagai pemrograman yang berbasis objek, sebuah program tentunya memiliki fitur yang dinamakan Class atau kelas. Sebuah kelas bisa dikatakan sebagai pengembangan dari struktur data. Perbedaan yang mencolok antara struktur data dan kelas adalah pada kelas, terdapat member-member berupa fungsi dan variabel, sedangkan pada struktur data hanya dapat menampung variabel saja. Didalam sebuah kelas terdapat anggota yang disebut member dan untuk mengakses member ini, maka diperlukan penentu akses. Penentu akses yang digunakan terbagi menjadi 3 kategori, yaitu Private, Public serta Protected.

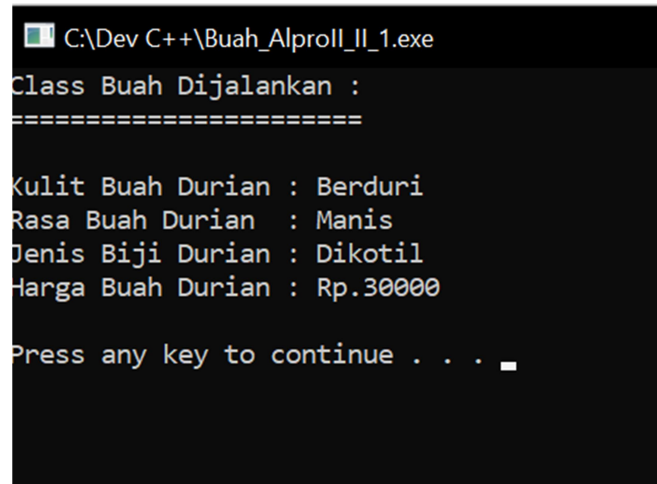
Inheritance atau Pewarisan adalah konsep pemrograman dimana sebuah Class dapat menurunkan Property dan method yang dimilikinya kepada Class lain. Konsep inheritance digunakan untuk memanfaatkan fitur code reuse untuk menghindari duplikat program. Konsep inheritance membuat struktur atau hierarchy class dalam kode program. Class yang akan diturunkan disebut Class Induk(Parent Class), super class atau base class. Sedangkan Class yang menerima penurunan disebut Class Anak atau Class Turunan. Tidak semua Property dan Method yang diturunkan dari Class Induk. Property dan method yang bisa diakses oleh kelas turunan hanyalah Protected dan Public, sedangkan hak akses Private tidak diturunkan kepada Class Turunan.

## DAFTAR PUSTAKA

- Ahmad muahrdian. 2019. *Fungsi Int Main*. <https://www.petanikode.com/c-syntak/>. (Diakses pada 12 April 2020, pukul 21.15 WIB)
- Dosen Teknik Informatika. 2020. *Modul Praktikum Algoritma Pemrograman II*. Print Pdf.
- Sinauarduino. 2016. *Pengertian Classes pada Pemrograman C++*. <https://www.sinauarduino.com/artikel/pengertian-classes-pada-pemrograman-cpp/> (Diakses pada 15 April 2020, pukul 21.00 WIB)
- Unknown. 2015. *Class Constructor pada C++*. <https://www.materipelajaran.web.id/2015/07/class-constructor-pada-cplusplus.html> (Diakses pada 15 April 2020, pukul 20.15 WIB)
- Yanuardi. 2018. *Macam-Macam Perintah atau Header pada C++*. <https://www.forumkomputer.com/2018/10/perintah-atau-header-pada-c.html> (Diakses pada 11 April 2020, pukul 14.20 WIB)

## LAMPIRAN

### 1. ~~Output Program Class Buah Durian~~



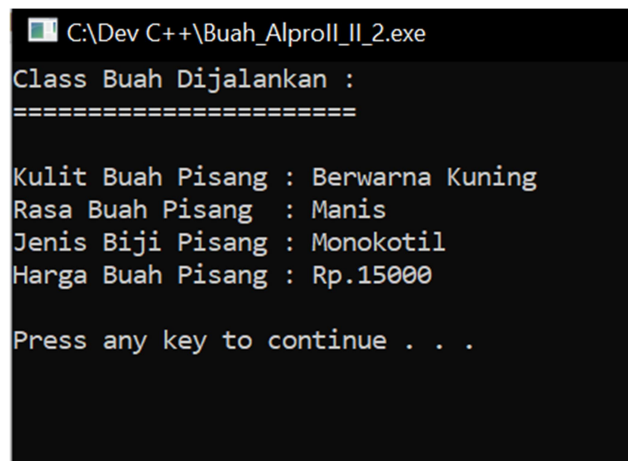
```
C:\Dev C++\Buah_Alproll_II_1.exe
Class Buah Dijalankan :
=====

Kulit Buah Durian : Berduri
Rasa Buah Durian   : Manis
Jenis Biji Durian  : Dikotil
Harga Buah Durian  : Rp.30000

Press any key to continue . . .
```

*Gambar 1. Output Class Buah Durian*

### 2. ~~Output Program Class Buah Pisang~~



```
C:\Dev C++\Buah_Alproll_II_2.exe
Class Buah Dijalankan :
=====

Kulit Buah Pisang : Berwarna Kuning
Rasa Buah Pisang   : Manis
Jenis Biji Pisang  : Monokotil
Harga Buah Pisang  : Rp.15000

Press any key to continue . . .
```

*Gambar 2. Output Class Buah Pisang*