

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



NAMA : OKTA ARYO
NIM : 193020503035
KELAS : A
MODUL : II (PEWARISAN)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2020

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



Nama : OKTA ARYO
NIM : 193020503035
Kelas : A
Modul : II (PEWARISAN)

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	6
BAB II Pembahasan	60	45
BAB III Kesimpulan	20	10
Daftar Pustaka	5	4
Lampiran	5	5
Jumlah	100	70

Penilai
Asisten Praktikum

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

1.1 TUJUAN

1. mahasiswa diharapkan mampu membuat kelas baru dari kelas yang sudah ada dengan pewarisan.

1.2 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 katagorisasi 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 subkatagori 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-katagori merupakan anggota dari kategori yang lebih umum. Contohnya adalah seperti di bawah ini.

- a. Mobil adalah (“is-a”) kendaraan bermotor. ☐ ☐ ☐ Truk adalah (“is-a”) kendaraan bermotor ☐
- b. 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

2.1. Program Pertama

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

using namespace std;

//kelas dalam buahan
class buahan
{
protected:
    char nama[8];
    char jenis[10];
    char namabuah[8];
    char jenisbuah[10];
    char rasabuah[8];
public:
    void buahmantan()
    {
        cout<<endl;
        cout<<"data dari buahan"<<endl;
    }
};

    class nanas: public buahan
    {
protected:
        char rasabuah[8];
```

```

public:
    //konstruktor
    nanas(char nma[8],char jns[10],char rsa_b[8])
    {
        strcpy(namabuah , nma);
        strcpy(jenisbuah, jns);
        strcpy(rasabuah, rsa_b);
    }
    void buahmantan()
    {
        cout<<"nama buahan : "<<namabuah<<endl;
        cout<<"jenis buahan : "<<jenisbuah<<endl;
        cout<<"rasa buahan : "<<rasabuah<<endl;
    }
};

int main()
{
    //menciptakan objek "nns" sekaligus melakukan inisialisasi
    //melalui Konstruktor dari Kelas nanas
    nanas nns("nanas", "bersisik", "hambar");
    cout<<"*=====HASIL DATA DARI
    BUAHAN=====*"<<endl;
    cout<<"_____ "<
    <endl;
    nns.buahmantan();

    getch();
    return 0;
}

```

Penjelasan :

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

using namespace std;

//kelas dalam buahan
class buahan
```

Batang program diatas adalah awalan dari program pewarisan buahan. Dan dengan menggunakan class buahan maka untuk pendeklarasiannya adalah menggunakan (//) dari kelas buahan.

```
protected:
    char nama[8];
    char jenis[10];
    char namabuah[8];
    char jenisbuah[10];
    char rasabuah[8];
public:
void buahmantan()
{
    cout<<endl;
    cout<<"data dari buahan"<<endl;
}
};
```

Code diatas dimana untuk pendeklarasian dari variable adalah protected bukan private. Hal ini dikarenakan private sama sekali tidak bisa diwariskan dan sedangkan protected mampu dan bisa untuk diwariskan.

Dan juga untuk variable data dari program diatas menggunakan char karena bersifat ke huruf-huruf yang bersifat terbatas sesuai dengan jumlah huruf yang dimasukkan didalam variable data char.

Dalam public program digunakan untuk membaca dan mengeksekusi void dari program supaya terbaca saat pengambilan hasil infutan dan output.

```
class nanas: public buahan
{
protected:
    char rasabuah[8];
public:
    //konstruktor
    nanas(char nma[8],char jns[10],char rsa_b[8])
    {
strcpy(namabuah , nma);
strcpy(jenisbuah, jns);
strcpy(rasabuah, rsa_b);
    }
}
```

Code diatas adalah program utama dimana kelas dari program adalah nanas. Dan untuk konstruktor untuk menentukan jumlah huruf dari variable data dari nanas yang terdiri dari nama, jenis dan rasa. Sedangkan untuk (strcpy) untuk menginfutkan langsung data dari buah nanas tanpa perlu menginfutkan Kembali ke dalam program karena data sudah terbaca dan tersimpan dialam program.

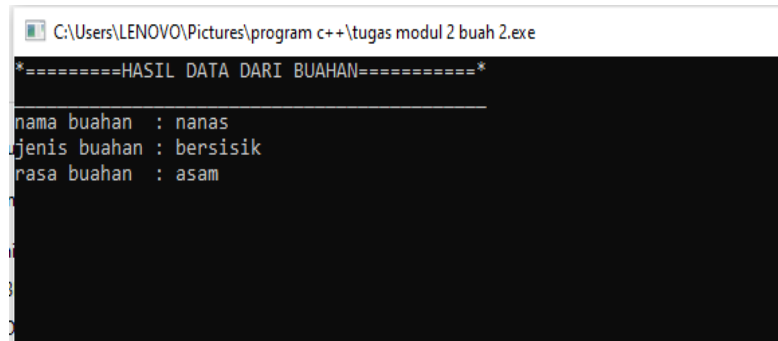

```

void buahmantan()
{
    cout<<"nama buahan : "<<namabuah<<endl;
    cout<<"jenis buahan : "<<jenisbuah<<endl;
    cout<<"rasa buahan : "<<rasabuah<<endl;
}
};

int main()
{
    //menciptakan objek "nns" sekaligus melakukan inisialisasi
    //melalui Konstruktor dari Kelas naga
    nanas nns("nanas", "bersisik", "hambar");
    cout<<"*=====HASIL DATA DARI
    BUAHAN=====*"<<endl;
    cout<<"_____ "<
    <endl;
    nns.buahmantan();
    getch();
    return 0;
}

```

Code utama dari program diatas adalah bagian akhir dari batang program dimana kita harus memasukkan data-data dari program kelas buah nanas agar terbaca langsung dari strcpy dan langsung tampil di outputnya nanti.

A screenshot of a Windows command prompt window. The title bar reads "C:\Users\LENOVO\Pictures\program c++\tugas modul 2 buah 2.exe". The output text is as follows:

```
*=====HASIL DATA DARI BUAHAN=====*  
nama buahan : nanas  
jenis buahan : bersisik  
rasa buahan : asam
```

Gambar 2.1 Hasil output Program pertama

Dari hasil output diatas kita tidak perlu menginfutkan lagi data-data dalam program dimana hasil data akan langsung ditampilkan saat program dijalankan.

2.2. Program kedua

```
#include<iostream>  
#include<conio.h>  
#include<string.h>  
  
using namespace std;  
  
//kelas dalam buahan  
class buahan  
{  
protected:  
char nama[5];  
char jenis[10];  
char namabuah[5];  
char jenisbuah[10];  
char rasabuah[8];  
  
public:
```

```

void buahmantan()
{
cout<<endl;
cout<<"data dari buahan"<<endl;
}
};

class naga : public buahan
{
protected:
char rasabuah[8];
public:
//konstruktor
naga(char nma[5],char jns[10],char rsa_b[8])
{
strcpy(namabuah , nma);
strcpy(jenisbuah, jns);
strcpy(rasabuah, rsa_b);
}
void buahmantan()
{
cout<<"nama buahan : "<<namabuah<<endl;
cout<<"jenis buahan : "<<jenisbuah<<endl;
cout<<"rasa buahan : "<<rasabuah<<endl;

}
};

int main()

{
//menciptakan objek "nga" sekaligus melakukan inisialisasi

```

```

//melalui Konstruktor dari Kelas naga

    naga nga("naga", "bersisik", "hambar");
cout<<"*=====HASIL          DATA          DARI
BUAHAN=====*"<<endl;
cout<<"_____ "<<endl;

nga.buahmantan();

getch();
return 0;
}

```

Penjelasan :

```

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

using namespace std;

//kelas dalam buahan
class buahan

```

Code program diatas sama dengan program pertama dimana Batang program diatas adalah awalan dari program pewarisan buahan. Dan dengan menggunakan class buahan maka untuk pendeklarasiannya adalah menggunakan (//) dari kelas buahan.

```

{
protected:

```

```
char nama[5];
char jenis[10];
char namabuah[5];
char jenisbuah[10];
char rasabuah[8];

public:
void buahmantan()
{
cout<<endl;
cout<<"data dari buahan"<<endl;
}
};
```

Batang program diatas hampir sama dengan program yang pertama dimana untuk pendeklarasian dari variable adalah protected bukan private. Hal ini dikarenakan private sama sekali tidak bisa diwariskan dan sedangkan protected mampu dan bisa untuk diwariskan. Dan juga untuk variable data dari program diatas menggunakan char karena bersifat ke huruf-huruf yang bersifat terbatas sesuai dengan jumlah huruf yang dimasukkan didalam variable data char.

Dalam public program digunakan untuk membaca dan mengeksekusi void dari program supaya terbaca saat pengambilan hasil infutan dan output. Hanya saja yang membedakan antara batang program pertama dan kedua adalah jumlah dari data variable char.

```
class naga : public buahan
{
protected:
char rasabuah[8];
```

```

public:
//konstruktor
naga(char nma[5],char jns[10],char rsa_b[8])
{
strcpy(namabuah , nma);
strcpy(jenisbuah, jns);
strcpy(rasabuah, rsa_b);
}
void buahmantan()
{
cout<<"nama buahan : "<<namabuah<<endl;
cout<<"jenis buahan : "<<jenisbuah<<endl;
cout<<"rasa buahan : "<<rasabuah<<endl;

}
};

```

Code diatas adalah program utama dimana kelas dari program adalah naga. Dan untuk konstruktor untuk menentukan jumlah huruf dari variable data dari naga yang terdiri dari nama, jenis dan rasa.

Sedangkan untuk (strcpy) untuk menginfutkan langsung data dari buah naga tanpa perlu menginfutkan Kembali ke dalam program karena data sudah terbaca dan tersimpan dialam program. Dimana variable char dari nama, jenis dan rasa dari buah naga untuk Batasan hurufnya dibatasi sesuai dengan jumlah angka yang terdapat didalam variabel char.

```

int main()

{
//menciptakan objek "nga" sekaligus melakukan inisialisasi

```

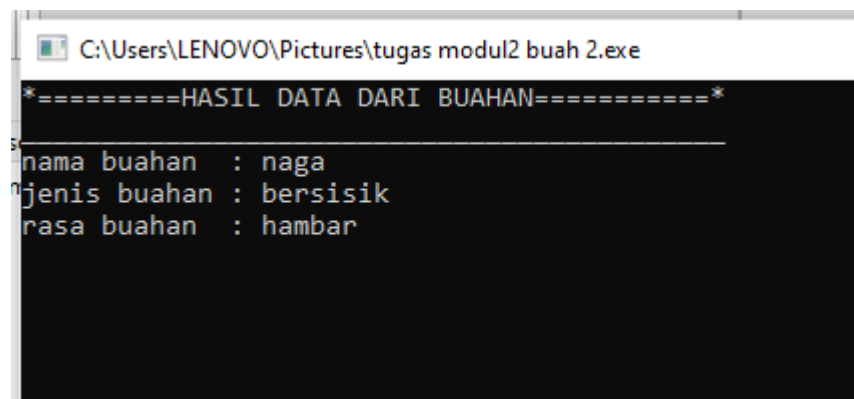
```
//melalui Konstruktor dari Kelas naga

naga nga("naga", "bersisik", "hambar");
cout<<"*=====HASIL          DATA          DARI
BUAHAN=====*"<<endl;
cout<<"_____ "<<endl;

nga.buahmantan();

getch();
return 0;
}
```

Batang program utama kedua ini sama dengan batang program pertama dimana kita harus memasukkan data-data dari program kelas buah nanas agar terbaca langsung dari strepy dan langsung tampil di outputnya nanti.



```
C:\Users\LENOVO\Pictures\tugas modul2 buah 2.exe
*=====HASIL DATA DARI BUAHAN=====*
nama buahan : naga
jenis buahan : bersisik
rasa buahan : hambar
```

Gambar 2.2 hasil output program kedua

Dari hasil output diatas kita tidak perlu menginfutkan lagi data-data dalam program dimana hasil data akan langsung ditampilkan saat program dijalankan.

BAB III

KESIMPULAN

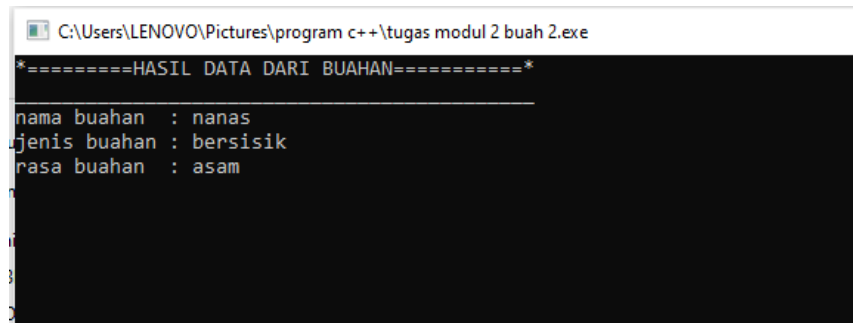
Pewarisan dalam pemrograman berorientasi objek sangat banyak didalam kehidupan sehari-hari dimana didalam pewarisan terdapat kelas-kelas dan sub kelas yang mempunyai objek-objek yang berwujud 2 dimensi maupun 3 dimensi. Pewarisan sendiri memiliki hierarki dalam penerapannya dimana hierarki tersebut memiliki turunan-turunan dari suatu kelas, contohnya saja manusia yang memiliki mata, hidung, telinga, dan anggota tubuh lainnya, begitupun sebaliknya dengan objek-objek lainnya yang bersifat benda maupun makhluk hidup.

DAFTAR PUSTAKA

Dosen Teknik Informatika. Algoritma dan Pemrograman II. 2020. Modul Praktikum Algoritma dan Pemrograman II. Universitas Palangka Raya. Fakultas Teknik. Jurusan Teknik Informatika.

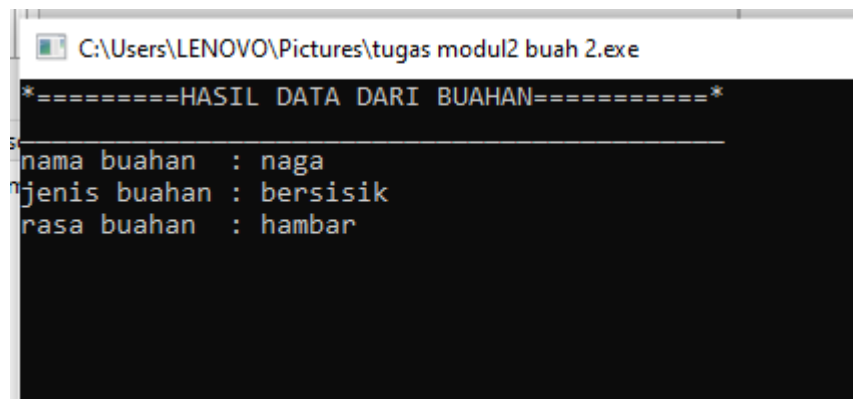
ditambah Lagi

LAMPIRAN



```
C:\Users\LENOVO\Pictures\program c++\tugas modul 2 buah 2.exe
*=====HASIL DATA DARI BUAHAN=====*
nama buahan : nanas
jenis buahan : bersisik
rasa buahan : asam
```

Gambar 2.1 hasil output program pertama



```
C:\Users\LENOVO\Pictures\tugas modul2 buah 2.exe
*=====HASIL DATA DARI BUAHAN=====*
nama buahan : naga
jenis buahan : bersisik
rasa buahan : hambar
```

Gambar 2.2 hasil output program kedua