

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : ELNATAN KENINGATKO
NIM : 193020503038
KELAS : A
MODUL : II (PEWARISAN)**

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

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



NAMA : ELNATAN KENINGATKO
NIM : 193020503038
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

1.2.1. Pengertian Pewarisan

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.

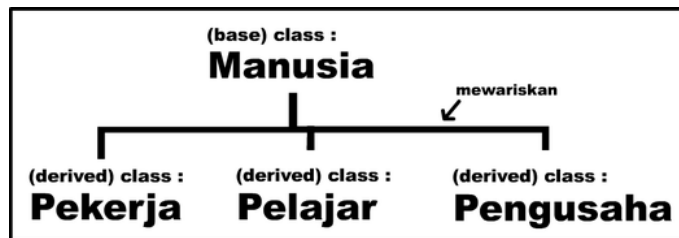
Konsep pewarisan akan mempermudah kita dalam membuat dan perawatan sebuah program, dengan menghilangkan “penggunaan kode berulang” saat pembuatan banyak class yang serupa.

1.2.2. Konsep Pewarisan

Contoh kasus yang tepat untuk penggunaan konsep pewarisan adalah ketika kita sedang membuat class macam-macam orang, dimana pertama-tama anda akan membuat sebuah class orang sebagai base class yang memiliki attribute dan method layaknya sebagai manusia pada dunia nyata, lalu base class tersebut isinya dapat diwarisi ke derived classes, dimana derived merupakan macam-macam golongan dari aktivitas manusia, misalnya class pekerja, class pelajar, class pengusaha dan lain sebagainya.

Dalam kasus tersebut, class dari macam-macam orang (pelajar, pengusaha, pekerja) harus mewarisi isi dari class orang, karena beberapa class tersebut memang membutuhkan identitas (attribute) dan sifat (method) dasar menjadikannya layaknya orang, dan hal itu

dimiliki oleh class orang. Dengan hal ini kita tidak perlu mengetikan kode berulang kali yang mirip seperti isi dari class orang ke beberapa class yang membutuhkan isi tersebut. Perhatikan Gambar 1.1. berikut.



Gambar 1.1. Ilustrasi Pewarisan

1.2.3. Pewarisan Tunggal

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 “:”. Berikut bentuk penulisannya.

```
class nama_class : access_speifier base_class {
    //...
};
```

Class nama_class, adalah tempat dimana anda membuat sebuah class dengan nama yang anda butuhkan yang diikuti sebuah keyword class. Tanda “:”, adalah tanda untuk memisahkan derived class dan class turunannya. Access_specifier base_class, adalah tempat dimana kita mendaftarkan base class yang ingin kita turunkan informasinya, dengan di ikuti access specifier yang akan menentukan sifat dari base class tersebut dalam derivered class itu.

1.2.4. Pewarisan Jamak

Pewarisan dimana satu kelas diturunkan lebih dari satu kelas yang berbeda disebut pewarisan jamak. Dalam pewarisan ini jumlah kelas dasarnya lebih dari satu, dan perlu dicatat bahwa kelas dasarnya bukan merupakan turunan dari satu kelas. Kelas turunan mewarisi seluruh sifat dari kelas dasarnya, sehingga sifat dari beberapa kelas dasar dan sifat khas dirinya. Perhatikan syntax dari pewarisan jamak berikut ini.

```
class A {  
    //...  
};  
  
class B {  
    //...  
}  
class C: public A, public B {  
    //...  
}
```

Pada bentuk tersebut terdapat dua kelas dasar yaitu kelas A dan kelas B yang menurunkan kelas C. Kelas C akan mewarisi sifat dari kelas A maupun sifat dari kelas B, tetapi tidak berlaku sebaliknya.

1.2.5. Tipe-Tipe Akses Pewarisan

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 :

1. Public

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.

2. Protected

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.

3. Private

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.

BAB II PEMBAHASAN

2.1. Program Pertama

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

class B {
    public:
        B(char *a,char *b);
        void cetakinfo();
    protected:
        char nama[10],rasa[20];
};

B::B(char *a,char *b){
    strcpy(nama,a);
    strcpy(rasa,b);
}

void B::cetakinfo(){
    cout<<"Informasi salah satu buah\n";
    cout<<"-----\n\n";
}

class buahnaga:public B {
    protected:
        char warna[20];
    public:
        buahnaga(char *a,char *b,char *w);
```

```

        void cetakinfo();
    };

    buahnaga::buahnaga(char *a,char *b, char *w):B(a,b) {
        strcpy(warna,w);
    }

    void buahnaga::cetakinfo() {
        B::cetakinfo();
        cout<<"Nama : "<<nama<<endl
            <<"Rasa : "<<rasa<<endl
            <<"Warna : "<<warna<<endl<<endl;
    }

    int main() {
        buahnaga bn("Buah naga","Manis dan asam","Merah keunguan");

        bn.cetakinfo();

        system("pause");
        return 0;
    }

```

Penjelasan :

Source code di atas merupakan sebuah program yang menampilkan informasi salah satu buah dengan menggunakan metode pewarisan tunggal..

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

String.h merupakan header yang berisi fungsi-fungsi, digunakan untuk menangani *string* ataupun *substring*. Header tersebut digunakan untuk

pengoperasian string dan *array*. Fungsi-fungsi yang terdapat di dalam header *string.h* antara lain : *strcpy*, *strncpy*, *strncat*, *strcmpt*, *strncmp* dan *strlen*.

```
class B {  
    public:  
        B(char *a,char *b);  
        void cetakinfo();  
    protected:  
        char nama[10],rasa[20];  
};
```

Penggalan code di atas merupakan kelas B yang juga merupakan kelas induk dari program. Kelas tersebut memiliki akses public dengan konstruktor “B(char *a, char *b)” dan *member function* “void cetakinfo()”, akses protected dengan variable “char nama[10],rasa[20]”. Kedua akses tersebut akan digunakan oleh kelas turunan dengan menggunakan nama atribut yang diakses.

```
B::B(char *a,char *b){  
    strcpy(nama,a);  
    strcpy(rasa,b);  
}
```

Penggalan code tersebut merupakan konstruktor dari kelas induk, dengan menggunakan type data *char*. Variable a dan b diberi pointer, gunanya yaitu sebagai variable asal yang datanya akan disalin ke variable nama dan rasa pada akses protected. Bentuk pendeklarasiannya terdapat pada code “strcpy(nama,a)” dan “strcpy(rasa,b)”.

```
void B::cetakinfo(){
    cout<<"Informasi salah satu buah\n";
    cout<<"-----\n\n";
}
```

Penggalan code di atas merupakan member functions “cetakinfo” yang terdapat di kelas induk. Dengan menggunakan nilai balikan *void*, kelas B dihubungkan dengan member function “cetakinfo” menggunakan tanda hubung “::”. Ketika member function tersebut di panggil, maka akan menampilkan output “Informasi salah satu buah” dan “-----”.

```
class buahnaga:public B {
    protected:
        char warna[20];
    public:
        buahnaga(char *a,char *b,char *w);
        void cetakinfo();
};
```

Penggalan code di atas merupakan kelas turunan dengan nama “buahnaga” yang dihubungkan ke public kelas B menggunakan tanda “:”. Kelas turunan tersebut memiliki akses protected dengan variable “char warna[20]”, akses public dengan konstruktor “buahnaga(char *a, char *b, char *w)” dan member function “void cetakinfo()”.

```
buahnaga::buahnaga(char *a,char *b, char *w):B(a,b) {
    strcpy(warna,w);
}
```

Penggalan code di atas merupakan konstruktor kelas turunan yang dihubungkan dengan konstruktor kelas induk menggunakan tanda “:” yang

berisi variable dengan akses public. Fungsinya yaitu agar kelas turunan dapat mengakses variable yang terdapat di kelas induk. Konstruktor tersebut memiliki tambahan variable w yang diberi tanda pointer sebagai variable asal yang datanya akan disalin ke variable warna pada akses protected. Bentuk pendeklarasiannya terdapat pada code “strcpy(warna,w)”.

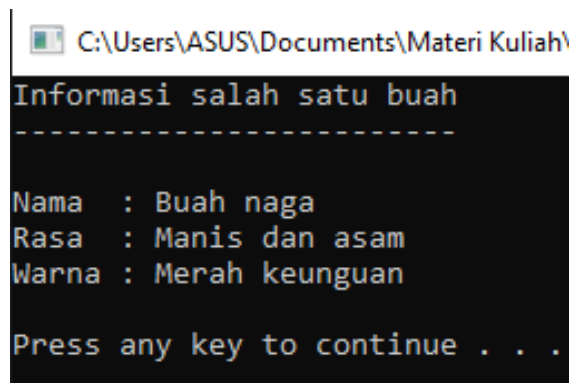
```
void buahnaga::cetakinfo() {  
    B::cetakinfo();  
    cout<<"Nama : "<<nama<<endl  
        <<"Rasa : "<<rasa<<endl  
        <<"Warna : "<<warna<<endl<<endl;  
}
```

Penggalan code di atas merupakan member function “cetakinfo” yang terdapat di kelas induk B yang diakses oleh kelas turunan buahnaga dengan nilai balikan void. Dengan memanggil member function “B::cetakinfo()”, maka otomatis data yang dibuat di member function kelas turunan buahnaga tersebut akan masuk ke member function kelas induk B karena telah diberi akses public. Member function buahnaga tersebut akan menampilkan output “Nama : “ dengan memanggil variable nama, kemudiann “Rasa : “ dengan memanggil variable rasa di baris berikutnya dan “Warna : “ dengan memanggil variable warna di baris berikutnya.

```
int main() {  
    buahnaga bn("Buah naga","Manis dan asam","Merah keunguan");  
  
    bn.cetakinfo();  
  
    system("pause");  
    return 0;
```

```
}
```

Penggalan code di atas merupakan badan utama program. Dengan memanggil kelas turunan buahnaga dan membuat objek baru dengan nama "bn". Objek bn berisikan data "Buah naga", "Manis dan asam" dan "Merah keunguan". Data tersebut akan otomatis masuk ke variable nama, rasa dan warna sesuai urutannya. Kemudian member function cetakinfo dipanggil dengan menghubungkannya ke objek bn menggunakan tanda ".". Dan diakhiri dengan "system("pause")" yang kegunaannya untuk menghentikan proses eksekusi program dengan menampilkan output "Press any key to continue..." yang artinya user diminta untuk menekan sembarang tombol keyboard agar eksekusi program dilanjutkan. Berikut ini output program :



```
C:\Users\ASUS\Documents\Materi Kuliah\'
Informasi salah satu buah
-----
Nama   : Buah naga
Rasa   : Manis dan asam
Warna  : Merah keunguan
Press any key to continue . . .
```

Gambar 2.1. Output Program Pertama

2.2. Program Kedua

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

class B {
    public:
        B(char *a,char *b);
```

```

        void cetakinfo();
    protected:
        char nama[10], rasa[20];
};

B::B(char *a, char *b) {
    strcpy(nama, a);
    strcpy(rasa, b);
}

void B::cetakinfo() {
    cout<<"Informasi salah satu buah\n";
    cout<<"-----\n\n";
}

class buahnaga: public B {
    protected:
        char warna[20];
    public:
        buahnaga(char *a, char *b, char *w);
        void cetakinfo();
};

buahnaga::buahnaga(char *a, char *b, char *w): B(a, b) {
    strcpy(warna, w);
}

void buahnaga::cetakinfo() {
    B::cetakinfo();
    cout<<"Nama : "<<nama<<endl;
    cout<<"Rasa : "<<rasa<<endl;
}

```

```

        cout<<"Warna : "<<warna<<endl<<endl;
    }

int main() {
    int pilih;
    do{
        ulang:
        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:{
                buahnaga bn("Buah naga","Manis dan
                    asam","Merah keunguan");
                bn.cetakinfo();
                system("pause");
                break;
            }
            case 2:{
                buahnaga bn("Apel","Manis","Merah atau hijau");
                bn.cetakinfo();
                system("pause");
                break;
            }
            case 3:{

```

```

        buahnaga bn("Mangga","Manis dan
        asam","Hijau");
        bn.cetakinfo();
        system("pause");
        break;
    }
    case 4:{
        system("pause");
        break;
    }
    default:{
        cout<<"Pilihan tidak tersedia!\n"
            <<"Silahkan memilih sesuai nomor yang
            tersedia.\n";
        system("pause");
        goto ulang;
        break;
    }

    }

}

while(pilih!=4);
return 0;
}

```

Penjelasan :

Program kedua ini merupakan program yang strukturnya sama seperti program pertama, hanya perbedaanya terdapat pada bagian badan utama program.

```
int main() {
    int pilih;
    do{
        ulang:
        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:{
                buahnaga bn("Buah naga","Manis dan
                    asam","Merah keunguan");
                bn.cetakinfo();
                system("pause");
                break;
            }
            case 2:{
                buahnaga bn("Apel","Manis","Merah atau hijau");
                bn.cetakinfo();
                system("pause");
                break;
            }
            case 3:{
                buahnaga bn("Mangga","Manis dan
                    asam","Hijau");
                bn.cetakinfo();
            }
        }
    } while (pilih != 4);
}
```



```

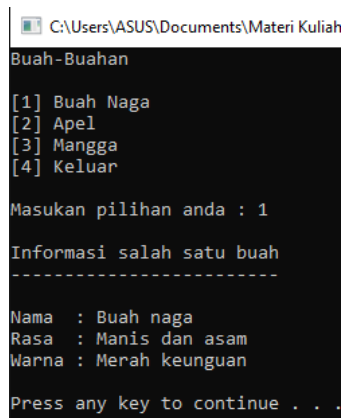
        system("pause");
        break;
    }
    case 4:{
        system("pause");
        break;
    }
    default:{
        cout<<"Pilihan tidak tersedia!\n"
            <<"Silahkan memilih sesuai nomor yang
            tersedia.\n";
        system("pause");
        goto ulang;
        break;
    }

    }
}
while(pilih!=4);
return 0;
}

```

Penggalan code di atas merupakan perbedaan yang terdapat pada program kedua. Program tersebut menggunakan perintah *switch case*, yaitu perintah yang membandingkan isi sebuah variable dengan beberapa nilai. Jika proses perbandingan tersebut menghasilkan nilai true, maka block code case akan dieksekusi dan di setiap block code case diakhiri perintah “break” agar eksekusi case berhenti pada saat kondisi terpenuhi. Pada saat user memasukan angka satu, maka program akan menampilkan output “Nama : “ dengan memanggil variable yang berisi data “Buah naga”, “Rasa : “ dengan memanggil variable yang berisi data “Manis dan asam” dan “Warna : “

dengan memanggil variable yang berisi data “Merah keunguan”. Begitu seterusnya sampai user memasukkan angka tiga yang masing-masing menampilkan data yang berbeda. Apabila user memasukkan pilihan angka yang bukan dari angka satu sampai empat, maka akan muncul output “Pilihan tidak tersedia” dan “Silahkan memilih sesuai no yang tersedia!” dan kemudian program akan dijalankan ulang. Hal tersebut dikarenakan adanya perintah “default” dan “goto”, perintah default akan dieksekusi apabila user memasukan data pada case yang nilainya tidak tertera di dalam case. Sedangkan perintah goto akan dieksekusi karena berada di dalam default, kegunaannya yaitu goto akan mengarahkan eksekusi program ke pernyataan yang diberi label sebelumnya. Sehingga, selama user tidak memasukkan data sesuai nomor yang tertera di case, maka perintah default akan dieksekusi dan mengulang program dari baris pernyataan yang diberi label oleh perintah goto. Berikut ini adalah hasil output program :



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

Gambar 2.2. Output Program Kedua, Pilihan Pertama

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

Masukan pilihan anda : 2

Informasi salah satu buah
-----
Nama : Apel
Rasa : Manis
Warna : Merah atau hijau
Press any key to continue . . .
```

Gambar 2.3. Output Program Kedua, Pilihan Kedua

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

Masukan pilihan anda : 3

Informasi salah satu buah
-----
Nama : Mangga
Rasa : Manis dan asam
Warna : Hijau
Press any key to continue . . .
```

Gambar 2.4. Output Program Kedua, 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 2.5. Output Program Kedua, Pilihan Keempat

BAB III

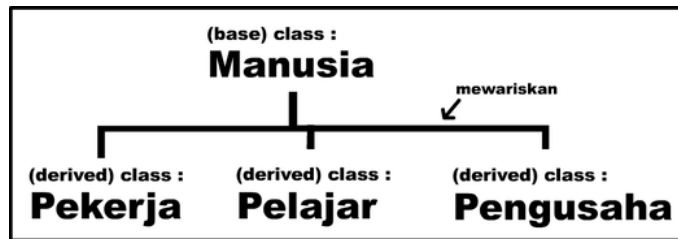
KESIMPULAN

Pewarisan adalah penurunan sifat yang ada pada suatu kelas induk kepada kelas baru yang menjadi kelas turunannya. Konsep pewarisan memiliki model relasi “adalah”, dimana class turunan akan dimungkinkan untuk mengambil isi dari class induk sebagai isi dari class tersebut. Ketentuan tersebut tidak berlaku sebaliknya, artinya sifat yang ada pada class turunan tidak akan diwariskan pada class induk. Pewarisan dibagi menjadi dua bagian, yaitu pewarisan tunggal dan pewarisan jamak.

DAFTAR PUSTAKA

- Anonim. 2011. *Belajar C++ Part 11*, diakses dari <http://pemogramanajip.blogspot.com/2011/12/belajar-c-part-11.html>, pada 15 April 2020.
- Laksono, Fajar. 2019. Penjelasan dan Tipe-tipe Pewarisan (Inheritance), diakses dari <https://www.belajarcpp.com/tutorial/cpp/inheritance/>, pada 15 April 2020.
- Tim Penyusun : Dosen Teknik Informatika. 2020. Modul Praktikum Algoritma dan Pemrograman II. Palangka Raya : Jurusan Teknik Informatika.

LAMPIRAN



Gambar 1.1. Ilustrasi Pewarisan

```
C:\Users\ASUS\Documents\Materi Kuliah\
Informasi salah satu buah
-----
Nama : Buah naga
Rasa : Manis dan asam
Warna : Merah keunguan
Press any key to continue . . .
```

Gambar 2.1. Output Program Pertama

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

Gambar 2.2. Output Program Kedua, Pilihan Pertama

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

Masukan pilihan anda : 2

Informasi salah satu buah
-----
Nama : Apel
Rasa : Manis
Warna : Merah atau hijau

Press any key to continue . . .
```

Gambar 2.3. Output Program Kedua, Pilihan Kedua

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

Masukan pilihan anda : 3

Informasi salah satu buah
-----
Nama : Mangga
Rasa : Manis dan asam
Warna : Hijau

Press any key to continue . . .
```

Gambar 2.4. Output Program Kedua, 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 2.5. Output Program Kedua, Pilihan Keempat