

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



Nama : OKKI ANDARESTA
NIM : DBC 118 068
Kelas : A
Modul : II (PEWARISAN)

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

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



Nama : Okki Andaresta
NIM : DBC 118 068
Kelas : A
Modul : PEWARISAN

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	8
BAB II Pembahasan	60	48
BAB III Kesimpulan	20	13
Daftar Pustaka	5	5
Lampiran	5	5
Jumlah	100	79

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/Penurunan* 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 duplikasi kode program.

Konsep *inheritance* membuat sebuah struktur atau ‘*hierarchy*’ *class* dalam kode program. Class yang akan ‘*diturunkan*’ bisa disebut sebagai *class induk (parent class)*, *super class*, atau *base class*. Sedangkan class yang ‘*menerima penurunan*’ bisa disebut sebagai *class anak (child class)*, *sub class*, *derived class* atau *heir class*. Tidak semua *property* dan *method* dari *class* induk akan diturunkan. *Property* dan *method* dengan hak akses *private*, tidak akan diturunkan kepada class anak. Hanya *property* dan *method* dengan hak akses *protected* dan *public* saja yang bisa diakses dari *class* anak.

Suatu class yang mempunyai class turunan dinamakan parent class atau base class. Sedangkan class turunan itu sendiri seringkali disebut subclass atau child class. Suatu subclass dapat mewarisi apa-apa yang dipunyai oleh parent class.

Karena suatu subclass dapat mewarisi apa apa yang dipunyai oleh parent class-nya, maka member dari suatu subclass adalah terdiri dari apa-apa yang ia punyai dan juga apa-apa yang ia warisi dari class parent-nya. Kesimpulannya, boleh dikatakan bahwa suatu subclass adalah tidak lain hanya memperluas (extend) parent class-nya.

Dari hirarki diatas dapat dilihat bahwa, semakin kebawah, class akan semakin bersifat spesifik. Class mamalia memiliki seluruh sifat yang dimiliki oleh hewan, demikian halnya juga macan , kucing, Paus dan Monyet memiliki seluruh sifat yang diturunkan dari class mamalia. Dengan konsep ini, karakteristik yang dimiliki oleh class binatang cukup didefinisikan dalam class binatang saja. Class mamalia tidak perlu mendefinisikan ulang apa yang telah dimiliki oleh class binatang, karena sebagai class turunannya, ia akan mendapatkan karakteristik dari class binatang secara otomatis. Demikian juga dengan class macan, kucing, Paus dan monyet, hanya perlu mendefinisikan karakteristik yang spesifik dimiliki oleh class-nya masing-masing.

Dengan memanfaatkan konsep pewarisan ini dalam pemrograman, maka hanya perlu mendefinisikan karakteristik yang lebih umum akan didapatkan dari class darimana ia diturunkan.

Keuntungan inheritance

1. Subclass menyediakan state/behaviour yang spesifik yang membedakannya dengan superclass, hal ini akan memungkinkan programmer Java untuk menggunakan ulang source code dari superclass yang telah ada.
2. Java dapat mendefinisikan superclass khusus yang bersifat generik, yang disebut abstract class, untuk mendefinisikan class dengan behaviour dan state secara umum.

Istilah Inheritance adalah sebagai berikut:

1. Extends

Keyword ini harus kita tambahkan pada definisi class yang menjadi subclass.

2. Superclass

Superclass digunakan untuk menunjukkan hirarki class yang berarti class dasar dari subclass/class anak.

3. Subclass

Subclass adalah class anak atau turunan secara hirarki dari superclass. Keyword ini digunakan untuk memanggil konstruktor dari superclass atau menjadi variabel yang mengacu pada superclass.

Method Overriding

Pendefinisian ulang method yang sama pada subclass dalam inheritance, method overriding berbeda dengan method overloading. Kalau method overriding adalah mendefinisikan kembali method yang sama, baik nama method maupun signature atau parameter yang diperlukan dalam subclass, kalau method overloading adalah mendefinisikan method yang memiliki nama yang sama, tetapi dengan signature yang berbeda dalam definisi class yang sama.

Kita baru perlu menerapkan inheritance pada saat kita jumpai ada suatu class yang dapat diperluas dari class lain. *Misal terdapat class Pegawai* `public class Pegawai { public String nama; public double gaji; }` *Misal terdapat class Manager* `public class Manajer { public String nama; public double gaji; public String departemen; }` Dari 2 buah class diatas, kita lihat class Manajer mempunyai data member yang identik sama dengan class Pegawai, hanya saja ada tambahan data member departemen. Sebenarnya yang terjadi disana adalah class Manajer merupakan perluasan dari class Pegawai dengan tambahan data member departemen.

BAB II

PEMBAHASAN

2.1 Program pertama

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

class induk{
    protected :
        int kromosom_AA,kromosom_XY;
    public:
        void input(int autosom,int gonosom);
};

void induk::input(int autosom,int gonosom){
    kromosom_AA = autosom;
    kromosom_XY = gonosom;
}

class anak : public induk{
    public :
        int total() {
            return kromosom_AA+kromosom_XY;}
};

int main(){
    int AA,XY;
    cout<<"Menghitung total Kromosom pada Domba" <<endl<<endl;
    cout<<"Masukkan Jumlah Autosom(AA) Domba : "; cin>>AA;
    cout<<"masukkan Jumlah Gonosom(XY) Domba : "; cin>>XY;
```

```

anak a;

    a.input(AA,XY);
    cout<<"Total jumlah Kromosom pada Domba adalah =
"<<a.total()<<endl;

    getch();
    return 0;

}

```

Penjelasan program pertama

```

#include <iostream>
#include <conio.h>
using namespace std;

class induk{
    protected :
        int kromosom_AA,kromosom_XY;
    public:
        void input(int autosom,int gonosom);
};

void induk::input(int autosom,int gonosom){
    kromosom_AA = autosom;
    kromosom_XY = gonosom;
}

class anak : public induk{
    public :
        int total() {
            return kromosom_AA+kromosom_XY;}
};

```

Code diatas terdapat dua buah kelas yaitu kelas induk dan kelas anak. Kelas induk bertugas dalam menerima input nilai dari user, sedangkan kelas anak berfungsi dalam proses kalkulasi nilai yang didapatkan dari input user tadi. Kelas induk memiliki member *protected* berupa kromosom_AA dan kromosom_XY serta member *public* berupa fungsi input. Fungsi input bertugas menerima input nilai melalui parameter Autosom serta Gonosom yang dimilikinya dan selanjutnya dialamatkan pada member kromosom_AA kromosom_XY. Kelas anak dideklarasikan terhadap kelas induk dengan penentu akses berupa *public*. Kelas anak ini memiliki member *public* berupa fungsi total. kelas anak mengakses member induk yaitu kromosom_AA, kromosom_XY, dimana member ini bersifat *protected*. Pada fuction return Untuk membalikkan nilai kepada pemanggil menggunakan formula penjumlahan kromosom_AA+kromosom_XY.

```
int main(){
    int AA,XY;
    cout<<"Menghitung total Kromosom pada Domba" <<endl<<endl;
    cout<<"Masukkan Jumlah Autosom(AA) Domba : "; cin>>AA;
    cout<<"masukkan Jumlah Gonosom(XY) Domba : "; cin>>XY;
    anak a;
    a.input(AA,XY);
    cout<<"Total jumlah Kromosom pada Domba adalah =
"<<a.total()<<endl;

    getch();
    return 0;
}
```

Dalam program juga terdapat **int main()**, kode ini menyatakan pembuatan suatu fungsi **main**. Setiap program C++ harus memiliki sebuah main karena main merupakan kode awal yang akan dieksekusi ketika program dijalankan. didalam

program tersebut juga terdapat fungsi **cout<<endl**, Fungsi dari **cout** yaitu untuk mencetak ke layar monitor, dan fungsi dari **endl** yaitu merupakan suatu fungsi manipulator yang digunakan untuk menyisipkan karakter Newline atau mengatur pindah baris.

Pada operasinya di fungsi utama/ main(), kelas anak membuat objek 'a'. Kemudian objek ini digunakan dalam mengakses member fungsi input milik kelas induk, serta pada tampilan layar diberikan masukkan nilai Autosom(AA) = 52, yang merupakan jumlah kromosom pada tubuh domba(Autosom dengan simbol AA). Dan diberikan masukkan nilai Gonosom(XY) = 2, yang menyatakan jumlah kromosom pada sel gamet pada domba jantan(Gonosom dengan simbol XY). Sebagai hasil operasi, objek 'a' kemudian mengakses member fungsi luas miliknya sendiri dan mencetak hasil kalkulasi penjumlahan dengan total nilai untuk menghitung jumlah seluruh gabungan(total) kromosom pada domba dengan nilai 54. Berikut ini merupakan hasil outputnya:

```
Menghitung total Kromosom pada Domba
Masukkan Jumlah Autosom(AA) Domba : 52
masukkan Jumlah Gonosom(XY) Domba : 2
Total jumlah Kromosom pada Domba adalah = 54
```

Gambar 2.1 Program Pertama

2.2. Program kedua

```
#include <iostream>
using namespace std;

class hewan{
protected:
int d,n;
public:
void input(float diploid , float pasang);
};

void hewan::input(float diploid, float pasang){
d = diploid;
n = pasang;
}

class kambing:public hewan{
public:
float total(){return 2*n;};
};

int main(){
kambing o;
o.input(2,30);
cout<<"Program total kromosom diploid kambing"<<endl;
cout<<"Nilai dari diploid(2n) kambing = "<<o.total()<<endl;
return 0;
}
```

Penjelasan program kedua

```
#include <iostream>
using namespace std;

class hewan{
protected:
int d,n;
public:
void input(float diploid , float pasang);
};

void hewan::input(float diploid, float pasang){
d = diploid;
n = pasang;
}

class kambing:public hewan{
public:
float total(){return 2*n;};
};
```

Pada code diatas terdapat dua buah kelas yaitu kelas hewan dan kelas Kambing. Kelas Hewan bertugas dalam menerima input nilai dari user, sedangkan kelas Kambing yang merupakan turunannya berfungsi dalam proses kalkulasi nilai yang didapatkan dari input user.

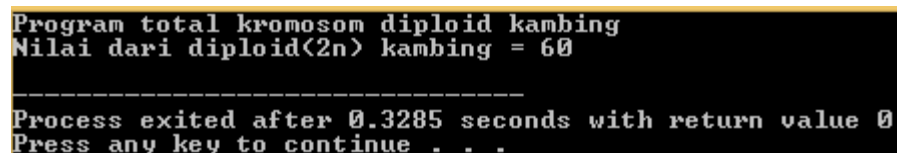
Kelas hewan memiliki member *protected* berupa **d** dan **n** serta member *public* berupa fungsi input. Fungsi input bertugas menerima input nilai melalui parameter panjang serta lebar yang dimilikinya dan selanjutnya dialamatkan pada member *protected* **d** dan **n**.

Kelas Kambing dideklarasikan terhadap kelas Hewan dengan penentu akses berupa *public*. Kelas kambing ini memiliki member *public* berupa fungsi total

dengan tipe data float , formula secara ilmiah kromosom diploid adalah $2n$, sehingga saya menggunakan formula $2*n$. Kelas kambing dideklarasikan dengan penentu akses *public*, maka member hewan *protected* akan tetap *protected* dan *public* akan tetap *public* dengan kata lain kelas kambing mengakses member hewan yaitu **d** dan **n**, dimana member ini bersifat *protected*.

```
int main(){
    kambing o;
    o.input(2,30);
    cout<<"Program total kromosom diploid kambing"<<endl;
    cout<<"Nilai dari diploid(2n) kambing = "<<o.total()<<endl;
    return 0;
}
```

Pada operasinya di fungsi utama/ main(), kelas kambing membuat objek 'o'. Kemudian objek ini digunakan dalam mengakses member fungsi input milik kelas hewan serta diberikan masukkan nilai 2 dan 30. Sebagai hasil operasi, objek 'o' kemudian mengakses member fungsi total miliknya sendiri dan mencetak hasil kalkulasi 60, yang mana diketahui sebagai jumlah keseluruhan kromosom pada kambing. Berikut ini output program yang didapatkan:



```
Program total kromosom diploid kambing
Nilai dari diploid(2n) kambing = 60
-----
Process exited after 0.3285 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.2 Program Kedua

BAB III

KESIMPULAN

Pewarisan atau inheritance adalah konsep dasar pemrograman berorientasi objek yang dapat membentuk interaksi antar class untuk mewarisi semua attribute dan method dari kelas lain sehingga menghasilkan suatu kelas baru yang menjadi turunannya. Dimana pada pewarisan ini terdapat penentu akses, yang merupakan hak akses maksimal dari member kelas induk yang diwariskan kepada kelas anak. Terdapat 3 penentu_akses yang mempengaruhi pewarisan/ *inheritance*, yaitu *Public*, *Protected* dan *Private*. Public pada member kelas induk yang berupa public akan tetap public dan yang protected tetap protected pada kelas anak. Sedangkan Protected pada member kelas induk yang berupa public akan berubah menjadi protected, dan protected tetap sebagai protected. Dan Private pada member yang diturunkan dari kelas induk pada kelas anak semua berubah menjadi private, baik itu public atau protected.

DAFTAR PUSTAKA

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

Andri Kristanto. 2009. *Algoritma & Pemrograman dengan C++*.
Yogyakarta. Graha Ilmu.

Budi Raharjo. 2010. *Pemrograman C++*. Bandung. Informatika.

sinauarduino. 2016. *Inheritance Pada Classes C++*.
<https://www.sinauarduino.com/artikel/inheritance-pada-classes-cpp/> (diakses pada : Selasa, 14 April 2020 pada pukul: 22.07)

LAMPIRAN

```
Menghitung total Kromosom pada Domba
Masukkan Jumlah Autosom(AA) Domba : 52
masukkan Jumlah Gonosom(XY) Domba : 2
Total jumlah Kromosom pada Domba adalah = 54
```

Gambar 2.1 Program Pertama

```
Program total kromosom diploid kambing
Nilai dari diploid(2n) kambing = 60
-----
Process exited after 0.3285 seconds with return value 0
Press any key to continue . . .
```

Gambar2.2 Program Kedua