

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II



NAMA : ERVINA AYU ROSAMADA
NIM : 193020503020
KELAS : A
MODUL : II (PEWARISAN)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA

2020

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II



NAMA : ERVINA AYU ROSAMADA

NIM : 193020503020

KELAS : A

MODUL : II (PEWARISAN)

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	7
BAB II Pembahasan	60	50
BAB III Kesimpulan	20	13
Daftar Pustaka	5	4
Lampiran	5	5
Jumlah	100	79

Penilai
Asisten Praktikum

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

I. TUJUAN

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

II. DASAR 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. □ Mobil adalah (“is-a”) kendaraan bermotor. □ □ Truk adalah (“is-a”) kendaraan bermotor □ □ 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 diakses.

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.

Tipe-tipe Inheritance / 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 :

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.

BAB II

PEMBAHASAN

2.5

Program I

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

using namespace std;

class hewan
{
    protected:
        char nama[20];
        char jenis[20];

    public:
        void info()
};

class sapi : public hewan
{
    protected:
        char kaki[8];

    public:

        sapi(const char nma[5], const char jns[8], const char kki[6])
        {
```

```

        strcpy(nama , nma);
        strcpy(jenis, jns);
        strcpy(kaki, kki);
    }
    void info()
    {
        cout<<"masukan nama hewan : "<<nama<<"      |"<<endl;
        cout<<"_____|"<<endl;
        cout<<nama<<" termasuk hewan jenis "<<jenis<<"      |"<<endl;
        cout<<"jumlah kaki "<<nama<<" ada "<<kaki<<"      |"<<endl;

    }
};

int main()

{

    sapi s("sapi", "mamalia", "empat");
    cout<<"*=====KLASIFIKASI HEWAN=====*"<<endl;
    cout<<"*****"<<endl;
    s.info();
    cout<<"_____|"<<endl;
return 0;
}

```

```
#include<iostream>
```

Tanda # disebut sebagai preprocessor directive, #include adalah sebuah professor pengarah yang menunjukkan kepada compiler untuk meletakkan kode dari header file iostream.

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

header yang berisi fungsi – fungsi makro dan tipe yang digunakan untuk pengoperasian string.

```
using namespace std;
```

Menyatakan bahwa program akan menggunakan namespace yang bernama std, sehingga tidak perlu untuk menyertakan identitas std:: pada anggota dari namespace std salah satunya adalah cout.

```
class hewan
```

Class adalah rancangan yang mendefinisikan variabel dan method-method pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program yang akan dijalankan suatu program. Class juga adalah sebuah konsep, konsep tersebut ada dua yaitu : method dan attribute.

```
{
```

Blok untuk membuka program.

```
protected:
```

Protected menyatakan bahwa semua anggota yang berada dalam lingkungan protected hanya dapat diakses oleh kelas yang mendefinisikannya dan kelas turunannya (subclass).

```
char nama[20];
```

Char adalah tipe data yang hanya bisa diisi dengan 1 karakter saja. Karakter ini harus ditulis diantara tanda kutip. Karena hanya bisa diisi dengan satu karakter saja, tipe data char hanya membutuhkan 1 byte memori (1 byte = 8 bit).


```
char jenis[20];
```

Char adalah tipe data yang hanya bisa diisi dengan 1 karakter saja. Karakter ini harus ditulis diantara tanda kutip. Karena hanya bisa diisi dengan satu karakter saja, tipe data char hanya membutuhkan 1 byte memori (1 byte = 8 bit).

```
public:
```

Public berfungsi untuk memberikan hak akses secara umum (public) kepada kelas-kelas turunannya maupun terhadap lingkungan luar didalam program. Bagian ini biasanya berisi fungsi-fungsi yang akan dijadikan sebagai penghubung (interface).

```
void info()
```

Fungsi pada tipe void tidak akan mengembalikan nilai apapun, void biasanya digunakan untuk menjalankan rutin program.

```
};
```

Blok untuk mengakhiri program.

```
class sapi : public hewan
```

Class adalah rancangan yang mendefinisikan variabel dan method-method pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program yang akan dijalankan suatu program. Class juga adalah sebuah konsep, konsep tersebut ada dua yaitu : method dan attribute. Public berfungsi untuk memberikan hak akses secara umum (public) kepada kelas-kelas turunannya maupun terhadap lingkungan luar didalam program. Bagian ini biasanya berisi fungsi-fungsi yang akan dijadikan sebagai penghubung (interface).

```
{
```

Blok untuk memulai program.

```
protected:
```

Protected menyatakan bahwa semua anggota yang berada dalam lingkungan protected hanya dapat diakses oleh kelas yang mendefinisikannya dan kelas turunannya (subclass).

```
char kaki[8];
```

Char adalah tipe data yang hanya bisa diisi dengan 1 karakter saja. Karakter ini harus ditulis diantara tanda kutip. Karena hanya bisa diisi dengan satu karakter saja, tipe data char hanya membutuhkan 1 byte memori (1 byte = 8 bit).

```
public:
```

Public berfungsi untuk memberikan hak akses secara umum (public) kepada kelas-kelas turunannya maupun terhadap lingkungan luar didalam program. Bagian ini biasanya berisi fungsi-fungsi yang akan dijadikan sebagai penghubung (interface).

```
sapi(const char nma[5], const char jns[8], const char kki[6])
```

Char adalah tipe data yang hanya bisa diisi dengan 1 karakter saja. Karakter ini harus ditulis diantara tanda kutip. Karena hanya bisa diisi dengan satu karakter saja, tipe data char hanya membutuhkan 1 byte memori (1 byte = 8 bit).

```
{
```

Blok untuk mengawali program.

```
strcpy(nama , nma);
```

Perintah strcpy bisa dipakai untuk mengcopy sebuah string ke dalam string lain, atau sebuah string ke dalam variabel. Strcpy adalah sebuah fungsi atau function.

```
strcpy(jenis, jns);
```

Perintah strcpy bisa dipakai untuk mengcopy sebuah string ke dalam string lain, atau sebuah string ke dalam variabel. Strcpy adalah sebuah fungsi atau function.

```
strcpy(kaki, kki);
```

Perintah strcpy bisa dipakai untuk mengcopy sebuah string ke dalam string lain, atau sebuah string ke dalam variabel. Strcpy adalah sebuah fungsi atau function.

```
}
```

Blok untuk mengakhiri program.

```
void info()
```

Fungsi pada tipe void tidak akan mengembalikan nilai apapun, void biasanya digunakan untuk menjalankan rutin program.

```
{
```

Blok untuk mengawali program.

```
cout<<"masukan nama hewan : "<<nama<<" |"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari iostream dan masuk sebagai anggota namespace std yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"_____|"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari iostream dan masuk sebagai anggota namespace std

yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<nama<<" termasuk hewan jenis "<<jenis<<"  |"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari iostream dan masuk sebagai anggota namespace std yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"jumlah kaki "<<nama<<" ada "<<kaki<<"  |"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari iostream dan masuk sebagai anggota namespace std yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
}
```

Blok untuk mengakhiri program.

```
};
```

Blok untuk mengakhiri program.

```
int main()
```

Int main() merupakan fungsi (function) utama, fungsi yang akan dibaca pertama kali oleh kompilator secara otomatis, tidak akan function yang dibaca otomatis oleh kompilator kecuali function utama int main(), fungsi utama adalah program yang mengatur arah kompilator. Int dalam fungsi utama adalah sebuah return type integer. Tanda () merupakan tempat mengisi parameter untuk function, tetapi dalam fungsi utama, hal itu tidak dibutuhkan dan dibiarkan dalam keadaan kosong.

```
{
```

Blok untuk mengawali program.

```
sapi s("sapi", "mamalia", "empat");
```

Bagian ini berfungsi untuk menampilkan di bagian void data, dengan inisial s bisa dengan mudah untuk memanggil sebuah fungsi.

```
cout<<"*=====KLASIFIKASI HEWAN=====*"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari iostream dan masuk sebagai anggota namespace std yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"*****"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk

menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
s.info();
```

Berfungsi untuk memanggil fungsi yaitu void data.

```
cout<<"_____|"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
Return 0;
```

Return 0; merupakan pernyataan pengembalian, untuk memberi tahu kepada system operasi bahwa program telah berakhir secara normal dengan bernilai 0 yang berarti tanpa kesalahan.

```
}
```

Blok untuk mengakhiri program.

Gambar berikut adalah gambar output dari program I

```
C:\Users\Acer\Documents\xxxxxap2.exe
*****KLASIFIKASI HEWAN*****
masukan nama hewan : sapi
sapi termasuk hewan jenis mamalia
jumlah kaki sapi ada empat

Process exited after 0.2874 seconds with return value 0
Press any key to continue . . .
```

Gambar 1.1 output program I

Program II

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

using namespace std;

class hewan
{
    protected:
        char nama[20];
        char jenis[20];

    public:
        void info()
};
```

```

class kupukupu : public hewan
{
    protected:
        char cirikhas[8];
    public:
        kupukupu(const char nma[5], const char jns[8], const char ck[6])
        {
            strcpy(nama , nma);
            strcpy(jenis, jns);
            strcpy(cirikhas, ck);
        }
        void info()
        {
            cout<<"masukan nama hewan : "<<nama<<"
|"<<endl;

            cout<<"_____
_____|"<<endl;

            cout<<"jenis "<<nama<<" "<<jenis<<"
|"<<endl;

            cout<<"ciri khas dari "<<nama<<" ini adalah "<<cirikhas<<"|"<<endl;

        }
};

int main()

{

```



```

    kupukupu k("kupu-kupu", "sayap bersisik", "bermotif dan warna yang menarik");
    cout<<"*=====KLASIFIKASI
HEWAN=====*"<<endl;
    cout<<"*****
*****"<<endl;
    k.info();
    cout<<"_____
_____|"<<endl;
return 0;
}

```

```
#include<iostream>
```

Tanda # disebut sebagai preprocessor directive, #include adalah sebuah professor pengarah yang menunjukkan kepada compiler untuk meletakkan kode dari header file iostream.

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

header yang berisi fungsi – fungsi makro dan tipe yang digunakan untuk pengoperasian string.

```
using namespace std;
```

Menyatakan bahwa program akan menggunakan namespace yang bernama std, sehingga tidak perlu untuk menyertakan identitas std:: pada anggota dari namespace std salah satunya adalah cout.

```
class hewan
```

Class adalah rancangan yang mendefinisikan variabel dan method-method pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program yang

akan dijalankan suatu program. Class juga adalah sebuah konsep, konsep tersebut ada dua yaitu : method dan attribute.

```
{
```

Blok untuk membuka program.

```
protected:
```

Protected menyatakan bahwa semua anggota yang berada di dalam lingkungan protected hanya dapat diakses oleh kelas yang mendefinisikannya dan kelas turunannya (subclass).

```
char nama[20];
```

Char adalah tipe data yang hanya bisa diisi dengan 1 karakter saja. Karakter ini harus ditulis diantara tanda kutip. Karena hanya bisa diisi dengan satu karakter saja, tipe data char hanya membutuhkan 1 byte memori (1 byte = 8 bit).

```
char jenis[20];
```

Char adalah tipe data yang hanya bisa diisi dengan 1 karakter saja. Karakter ini harus ditulis diantara tanda kutip. Karena hanya bisa diisi dengan satu karakter saja, tipe data char hanya membutuhkan 1 byte memori (1 byte = 8 bit).

```
public:
```

Public berfungsi untuk memberikan hak akses secara umum (public) kepada kelas-kelas turunannya maupun terhadap lingkungan luar didalam program. Bagian ini biasanya berisi fungsi-fungsi yang akan dijadikan sebagai penghubung (interface).

```
void info()
```

Fungsi pada tipe void tidak akan mengembalikan nilai apapun, void biasanya digunakan untuk menjalankan rutin program.

```
};
```

Blok untuk mengakhiri program.

```
class sapi : public hewan
```

Class adalah rancangan yang mendefinisikan variabel dan method-method pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program yang akan dijalankan suatu program. Class juga adalah sebuah konsep, konsep tersebut ada dua yaitu : method dan attribute. Public berfungsi untuk memberikan hak akses secara umum (public) kepada kelas-kelas turunannya maupun terhadap lingkungan luar didalam program. Bagian ini biasanya berisi fungsi-fungsi yang akan dijadikan sebagai penghubung (interface).

```
{
```

Blok untuk memulai program.

```
protected:
```

Protected menyatakan bahwa semua anggota yang berada dalam lingkungan protected hanya dapat diakses oleh kelas yang mendefinisikannya dan kelas turunannya (subclass).

```
char cirikhas[8];
```

Char adalah tipe data yang hanya bisa diisi dengan 1 karakter saja. Karakter ini harus ditulis diantara tanda kutip. Karena hanya bisa diisi dengan satu karakter saja, tipe data char hanya membutuhkan 1 byte memori (1 byte = 8 bit).

```
public:
```

Public berfungsi untuk memberikan hak akses secara umum (public) kepada kelas-kelas turunannya maupun terhadap lingkungan luar didalam program. Bagian ini biasanya berisi fungsi-fungsi yang akan dijadikan sebagai penghubung (interface).

```
kupukupu(const char nma[5], const char jns[8], const char ck[6])
```

Char adalah tipe data yang hanya bisa diisi dengan 1 karakter saja. Karakter ini harus ditulis diantara tanda kutip. Karena hanya bisa diisi dengan satu karakter saja, tipe data char hanya membutuhkan 1 byte memori (1 byte = 8 bit).

```
{
```

Blok untuk mengawali program.

```
strcpy(nama , nma);
```

Perintah strcpy bisa dipakai untuk mengcopy sebuah string ke dalam string lain, atau sebuah string ke dalam variabel. Strcpy adalah sebuah fungsi atau function.

```
strcpy(jenis, jns);
```

Perintah strcpy bisa dipakai untuk mengcopy sebuah string ke dalam string lain, atau sebuah string ke dalam variabel. Strcpy adalah sebuah fungsi atau function.

```
strcpy(cirikhas, ck);
```

Perintah strcpy bisa dipakai untuk mengcopy sebuah string ke dalam string lain, atau sebuah string ke dalam variabel. Strcpy adalah sebuah fungsi atau function.

```
}
```

Blok untuk mengakhiri program.

```
void info()
```

Fungsi pada tipe void tidak akan mengembalikan nilai apapun, void biasanya digunakan untuk menjalankan rutin program.

```
{
```

Blok untuk mengawali program.

```
cout<<"masukan nama hewan : "<<nama<<"      |"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari iostream dan masuk sebagai anggota namespace std

yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"_____|"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<nama<<" termasuk hewan jenis "<<jenis<<" |"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"ciri khas dari "<<nama<<" ini adalah "<<cirikhas<<"|"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std*

yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
}
```

Blok untuk mengakhiri program.

```
};
```

Blok untuk mengakhiri program.

```
int main()
```

Int main() merupakan fungsi (function) utama, fungsi yang akan dibaca pertama kali oleh kompilator secara otomatis, tidak akan function yang dibaca otomatis oleh kompilator kecuali function utama int main(), fungsi utama adalah program yang mengatur arah kompilator. Int dalam fungsi utama adalah sebuah return type integer. Tanda () merupakan tempat mengisi parameter untuk function, tetapi dalam fungsi utama, hal itu tidak dibutuhkan dan dibiarkan dalam keadaan kosong.

```
{
```

Blok untuk mengawali program.

```
kupukupu k("kupu-kupu", "sayap bersisik", "bermotif dan warna yang menarik");
```

Bagian ini berfungsi untuk menampilkan di bagian void data, dengan inisial s bisa dengan mudah untuk memanggil sebuah fungsi.

```
cout<<"*=====KLASIFIKASI HEWAN=====*"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor.

Endl merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"*****"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
k.info();
```

Berfungsi untuk memanggil fungsi yaitu void data.

```
cout<<"_____|"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

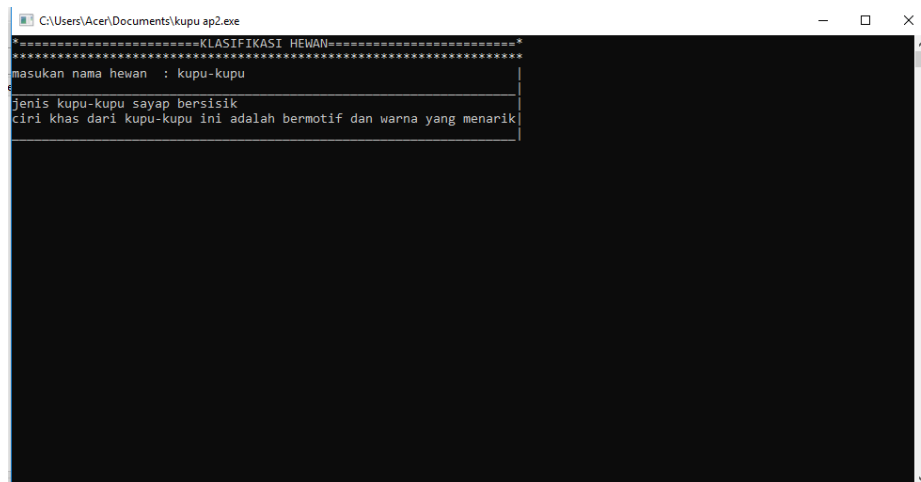
```
Return 0;
```

Return 0; merupakan pernyataan pengembalian, untuk memberi tahu kepada system operasi bahwa program telah berakhir secara normal dengan bernilai 0 yang berarti tanpa kesalahan.

```
}
```

Blok untuk mengakhiri program.

Gambar berikut adalah gambar output dari program II



```
C:\Users\Acer\Documents\kupu ap2.exe
====KLASIFIKASI HEWAN====
masukan nama hewan : kupu-kupu
jenis kupu-kupu sayap bersisik
ciri khas dari kupu-kupu ini adalah bermotif dan warna yang menarik
```

Gambar 1.2 output program II

BAB III

KESIMPULAN

Pewarisan adalah keuntungan besar dalam pemrograman berbasis object karena suatu sifat atau method didefinisikan dalam superclass, sifat ini secara otomatis diwariskan dari semua subclasses. Jadi, Kita dapat menuliskan kode method hanya sekali dan mereka dapat digunakan oleh semua subclass. Subclass hanya perlu mengimplementasikannya perbedaan sendiri dari induknya.

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.

DAFTAR PUSTAKA

Judul : konsep dasar pemrograman berorientasi objek

Penulis : Rizki Aditya

Tanggal tayang: 06/2016

Waktu akses : 13/04/2020

URL : <http://www.adityarizki.net/2012/06/konsep-dasar-pemograman-berorientasi-objek/>

Judul : kelas dan objek dalam C

Penulis : Irwansyah Dio

Tanggal tayang: 18/03/2009

Waktu akses : 13/04/2020

URL : <http://dioirwansyah.wordpress.com/2009/03/18/kelas-dan-objek-dalam-c-/>.

Judul : tutorial cpp inheritance

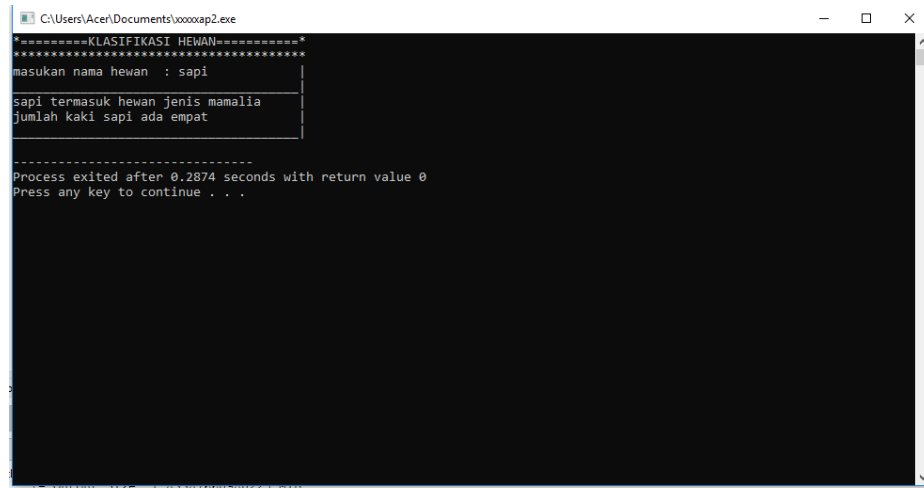
Penulis : Fajar

Tanggal tayang: 20/05/2019

Waktu akses : 15/04/2020

URL : <https://www.belajarcpp.com/tutorial/cpp/inheritance/2019/05/20>

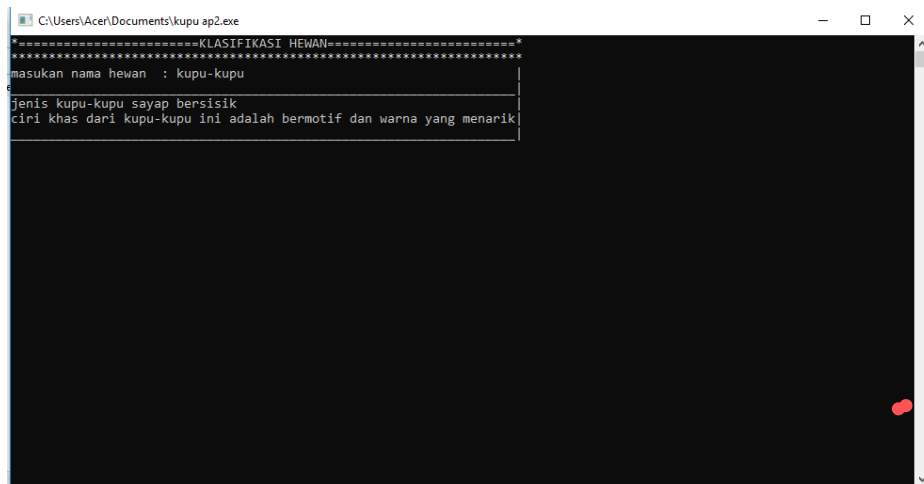
LAMPIRAN



```
C:\Users\Acer\Documents\xxxxap2.exe
====KLASIFIKASI HEWAN====
masukan nama hewan : sapi
sapi termasuk hewan jenis mamalia
jumlah kaki sapi ada empat

Process exited after 0.2874 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.1 output program I



```
C:\Users\Acer\Documents\kupu ap2.exe
====KLASIFIKASI HEWAN====
masukan nama hewan : kupu-kupu
jenis kupu-kupu sayap bersisik
ciri khas dari kupu-kupu ini adalah bermotif dan warna yang menarik
```

Gambar 2.2 output program II