

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



NAMA : HUSSAIN SYACH NURALAM
NIM : 193020503029
KELAS : A
MODUL : II (PEWARISAN)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA
2020

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



Nama : Hussain Syach Nuralam
NIM : 193020503029
Kelas : A
Modul : II (Pewarisan)

Komposisi	MAX	Nilai
BAB I Tujuan dan LandasanTeori	10	8
BAB II Pembahasan	60	50
BAB III Kesimpulan	20	13
Daftar Pustaka	5	5
Lampiran	5	5
Jumlah	100	81

Penilai
AsistenPraktikum

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

1.1 Tujuan

Tujuan dari dilaksanakannya praktikum Algoritma dan Pemrograman 2, modul II, Pewarisan adalah agar mahasiswa mampu membuat kelas baru dari kelas yang sudah ada dengan menggunakan pewarisan (*Inheritance*).

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. Se jauh 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.

- a. Mobil adalah (“is-a”) kendaraan bermotor.
- b. Truk adalah (“is-a”) kendaraan bermotor
- c. Sepeda Motor adalah (“is-a”) kendaraan bermotor

Objek yang dikelompokkan dalam satu kelas membagikan (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.

Sehingga, didapatkan pengertian dari pewarisan atau inheritance. 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.

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 “:”.

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

Gambar 1.1 Bentuk penulisan base class

Dengan penjelasan sebagai berikut :

- a) class nama_class ; adalah tempat dimana anda membuat sebuah class dengan nama yang anda butuhkan yang diikuti sebuah keyword class.
- b) Tanda : ; adalah tanda untuk memisahkan derived class dan class turunannya.
- c) 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 derived class itu.

Berikut adalah contoh penulisannya pada program.

```
class pekerja : public orang{  
    //...  
};
```

Gambar 1.2 Contoh base class pada program

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.

BAB II PEMBAHASAN

2.1 Program Pertama Pewarisan Pada Program Hewan

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

using namespace std;

class Hewan { //kelas Hewan
    public:
        Hewan(char *a,char *b);
        void tampildata();

    protected:
        char nama[15],kelas[15];
};

Hewan::Hewan(char *a,char *b){ //konstruktor
    strcpy(nama, a);
    strcpy(kelas, b);
}

void Hewan::tampildata(){
    cout<<"Baik. Hewan yang anda pilih adalah:\n";
}
```

Gambar 2.1 Tugas Pewarisan pertama

```

class Kucing:public Hewan { //kelas turunan
    protected:
        char jenis[15];
    public:
        Kucing(char *a,char *b,char *c);
        void tampilkandata();
};

Kucing::Kucing(char *a,char *b, char *c):Hewan(a,b) { //konstruktor kelas
turunan
    strcpy(jenis,c);
}

void Kucing::tampilkandata() {
    Hewan::tampilkandata();
    cout<<"Nama           : "<<nama<<endl;
    cout<<"Kelas           : "<<kelas<<endl;
    cout<<"Jenis             : "<<jenis<<endl<<endl;
}

int main() { //program utama
    int pilih;
    do{
        ulang:
        system("cls");
        cout<<"--SELAMAT DATANG DI RUMAH HUSIN--\n"
        <<"-----HEWAN PELIHARAAN HUSIN-----\n\n"
        <<"HEWAN APA YANG INGIN ANDA LIHAT?\n"

```

Gambar 2.2 Tugas Pewarisan pertama

```

        <<"[1] Kucing\n"
        <<"[2] Ayam\n"
        <<"[3] Kadal\n"
        <<"[4] Keluar\n\n"
        <<"Masukan pilihan : "; cin>>pilih;
cout<<endl;

switch(pilih) {
    case 1:{
        Kucing bb("Kucing","Mamalia","Anggora");
        bb.tampildata();
        system("pause");
        break;
    }
    case 2:{
        Kucing bb("Ayam","Aves","Silkie");
        bb.tampildata();
        system("pause");
        break;
    }
    case 3:{
        Kucing bb("Kadal","Reptil","Iguana");
        bb.tampildata();
        system("pause");
        break;
    }
}

```

Gambar 2.3 Tugas Pewarisan pertama


```

        case 4:{
            system("cls");
            break;
        }
        default:{
            cout<<"Pilihan tidak ada!\n"
            <<"Silahkan memilih sesuai nomor yang ada.";
            goto ulang;
            break;
        }
    }
}
while(pilih!=4);
return 0;
}

```

Gambar 2.4 Tugas Pewarisan pertama

Penjelasan :

```

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

using namespace std;

```

Gambar 2.5 File Header

Kode di atas merupakan file header. File header adalah file yang berisi deklarasi untuk berbagai fungsi yang dibutuhkan oleh program baik itu sebagai standar Input/Output maupun sebagai syntax. Berikut adalah macam file header yang terdapat pada program diatas.

- 1) **iostream.h (iostream)** : Merupakan singkatan dari input outout stream header yang digunakan sebagai standar input output operasi yang digunakan oleh

bahasa C++. Fungsi-fungsi yang ada didalam iostream.h antara lain sebagai berikut :

- 1.1) **cout** : Merupakan fungsi keluaran pada C++ yang menampilkan data dengan tipe data apapun kelayar.
- 1.2) **cin** : Merupakan fungsi masukan pada C++ yang bisa memasukan data berupa numerik dan karakter.
- 1.3) **endl** : Merupakan suatu fungsi yang manipulator yang digunakan untuk melakukan perintah Newline atau pindah baris
- 2) **string.h** : Merupakan file header yang berfungsi untuk melakukan manipulasi string. Fungsi-fungsi yang ada di string.h antara lain sebagai berikut :
 - 2.1) **strcpy()** : fungsi ini digunakan untuk menyalin suatu string ke variabel tujuan. Bentuk umum penulisannya adalah strcpy (variabel tujuan, string).
 - 2.2) **strlen()** : fungsi ini digunakan untuk menghitung jumlah karakter yang ada dalam suatu string. Bentuk umum penulisannya adalah strlen (string).
 - 2.3) **strcmp()** : fungsi ini digunakan untuk membandingkan 2 buah string. Bentuk umum penulisannya adalah strcmp (string1,string2).
 - 2.4) **strrev()** : fungsi ini digunakan untuk membalikan urutan suatu string. Bentuk umum penulisannya adalah strrev (string).
 - 2.5) **strlwr()** : fungsi ini digunakan untuk mengubah semua huruf menjadi huruf kecil. Bentuk umum penulisannya adalah strlwr (string).
 - 2.6) **strupr()** : fungsi ini digunakan untuk mengubah semua huruf menjadi huruf kapital. Bentuk umum penulisannya adalah strupr (string).
 - 2.7) **strcat()** : fungsi ini digunakan untuk menggabungkan 2 buah string, untuk menggunakan fungsi ini juga harus menambahkan file header ctype.h. Bentuk umum penulisannya adalah strcat (variabel tujuan, string).
- 3) **conio.h** : Merupakan File Header yang berfungsi untuk menampilkan hasil antarmuka kepada pengguna. Fungsi -fungsi yang ada didalam conio.h antara lain sebagai berikut :

- 3.1) **getch()** : merupakan singkatan dari get character and echo yang digunakan untuk menahan (pause) output suatu program dan akan kembali mengeksekusi setelah kita melakukan inputan baik itu tombol enter atau tombol lainnya dan inputan tersebut tidak ditampilkan dalam window.
- 3.2) **getche()** : secara fungsi sama dengan getch() akan tetapi ketika melakukan inputan, inputan tersebut tampil dalam window.
- 3.3) **clrscr()** : merupakan singkatan dari clear screen yang digunakan untuk membersihkan layar windows .
- 4) **using namespace std** : File header ini terdiri dari 3 statements yaitu “using”, “namespace” dan “std”. Fungsi-fungsi penggunaan using namespace std adalah sebagai berikut :
- 3.1) Using adalah deklarasi arahan / panggilan / pemberitahuan kepada kompiler untuk penggunaan deklarasi namespace dan anggota namespace.
- 3.2) Namespace adalah penyedia metode untuk mencegah konflik nama dalam proyek-proyek besar. Simbol pendeklarasian di dalam blok namespace ditempatkan di lingkup bernama yang akan mencegah mereka pada keliruan pengenalan pada scope yang lainnya. Jika pendeklarasiannya seperti diatas, itu berfungsi untuk memanggil namespace yang telah dibuat.
- 3.3) Std adalah nama namespace tersebut yang sudah tersedia dan tidak perlu dibuat ulang menggunakan fungsi namespace, cukup dipanggil. Std merupakan wadah urutan yang merangkum ukuran data dan array dinamis. Di dalam std terdapat halnya seperti cout, cin, endl dan lain-lain.

```

class Hewan { //kelas Hewan
public:
    Hewan(char *a,char *b);
    void tampildata();

protected:
    char nama[15],kelas[15];
};
Hewan::Hewan(char *a,char *b){ //konstruktor
    strcpy(nama, a);
    strcpy(kelas, b);
}

```

Gambar 2.6 Class

Kode program di atas merupakan kode yang berjenis class yang bernama *Hewan*. Class adalah salah satu dari konsep OOP yang digunakan untuk membungkus data abstraksi procedural sebagai deskripsi tergeneralisir atau rancangan dari sebuah object untuk mendefinisikan atau menggambarkan isi dan tingkah laku sebagai entitas dari object. Untuk mendirikan Class kita membutuhkan keyword class yang dilanjutkan dengan pemberian nama dari deklarasi class tersebut. lalu dilanjutkan dengan meletakkan tanda { dan } untuk mengapit definisi dari class. Class termasuk sebuah pernyataan maka dari itu akhir dari deklarasi class diwajibkan untuk mengakhiri class menggunakan tanda titik-koma (;).

Di dalam kelas ini juga terdapat tipe data yakni char (character). Char adalah salah satu tipe data sembarang huruf, angka, dan simbol. Yang memungkinkan kita untuk memesan memori yang dapat menyimpan nilai dalam bentuk karakter tunggal seperti 'a', '\$' dan lain-lain.

Terdapat pula keyword public dan protected yang memiliki fungsi sebagai berikut.

1) Public

Keyword public menyatakan bahwa semua anggota yang berada dalam lingkungan public ini dapat diakses di luar kelas yang mendefinisikannya.

2) Protected

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

Terdapat pula konstruktor yakni Hewan::Hewan(char *a,char *b). Konstruktor adalah fungsi anggota dari suatu kelas yang secara otomatis dijalankan ketika suatu objek diciptakan.

Terdapat perintah strcpy yang bisa dipakai untuk menyalin sebuah string ke dalam string lain, atau sebuah string ke dalam variabel. Dalam bahasa C maupun C++ dan sebagainya, strcpy adalah sebuah fungsi atau function.

```
void Hewan::tampildata(){  
    cout<<"Baik. Hewan yang anda pilih adalah:\n";  
}
```

Gambar 2.7 Void

Kode di atas merupakan function dan function yang digunakan adalah void. Void adalah sebuah fungsi (function) yang ada dalam sebuah bahasa pemrograman C, entah itu C++ atau C#. Fungsi ini juga disebut sebagai prosedur (procedure). Fungsi ini tidak mengembalikan nilai keluaran (return output) yang didapat dari hasil proses tersebut, ini kenapa fungsi ini disebut void, secara harfiah berarti kosong.

```

class Kucing:public Hewan { //kelas turunan
    protected:
        char jenis[15];
    public:
        Kucing(char *a,char *b,char *c);
        void tampildata();
};

```

Gambar 2.8 Kelas turunan

Kode di atas merupakan kelas turunan atau subclass yang mewarisi superclass yakni class *Hewan*. Kelas turunan (subclass) akan mewarisi data field (properti atau atribut) serta metoda terakses (accessible methods) yang dimiliki superclass. Superclass akan memiliki sifat lebih umum (general), sedangkan kelas turunan akan lebih spesifik. Umumnya, kelas turunan mempunyai data field dan metoda tambahan selain data field dan metoda yang diwarisi dari superclass.

```

int main() { //program utama
    int pilih;
    do{
        ulang:
        system("cls");
        cout<<"--SELAMAT DATANG DI RUMAH HUSIN--\n"
        <<"-----HEWAN PELIHARAAN HUSIN-----\n\n"
        <<"HEWAN APA YANG INGIN ANDA LIHAT?\n"
            <<"[1] Kucing\n"
            <<"[2] Ayam\n"
            <<"[3] Kadal\n"
            <<"[4] Keluar\n\n"
            <<"Masukan pilihan : "; cin>>pilih;
        cout<<endl;
    }while(pilih!=4);
}

```

Gambar 2.9 Main program

Kode di atas merupakan int main(). Int main() artinya main program mengembalikan nilai int secara default, int main() akan mengembalikan nilai 0, dan fungsi main() tidak memiliki bagan deklarasi lokal, dan hanya memiliki sebuah pernyataan yang dapat dieksekusi, berupa fungsi output printf().

```
switch(pilih) {  
    case 1:{  
        Kucing bb("Kucing","Mamalia","Anggora");  
        bb.tampildata();  
        system("pause");  
        break;  
    }  
    case 2:{  
        Kucing bb("Ayam","Aves","Silkie");  
        bb.tampildata();  
        system("pause");  
        break;  
    }  
    case 3:{  
        Kucing bb("Kadal","Reptil","Iguana");  
        bb.tampildata();  
        system("pause");  
        break;  
    }  
    case 4:{  
        system("cls");  
        break;  
    }  
}
```

Gambar 2.10 Switch pada main program

Kode diatas merupakan pernyataan switch case yang terdapat pada main program. Pernyataan switch adalah adalah pernyataan penyeleksian untuk

memanipulasi jalanya aliran program berdasarkan conditional expression, Mengendalikan aliran program agar dapat berjalan secara fleksibel atas keadaan program dan keinginan pengguna. Pernyataan ini akan memilih pilihan yang sesuai dengan ekspresi kondisi.

Pada umumnya pernyataan switch tidak jauh berbeda seperti pernyataan penyeleksian IF, yang berbeda adalah; jika pada if, if mengevaluasi kebenaran dari sebuah ekspresi (conditional expression) seperti operasi logika, relasi dan perbandingan yang menghasilkan bilangan boolean 1 (benar) atau 0 (salah).

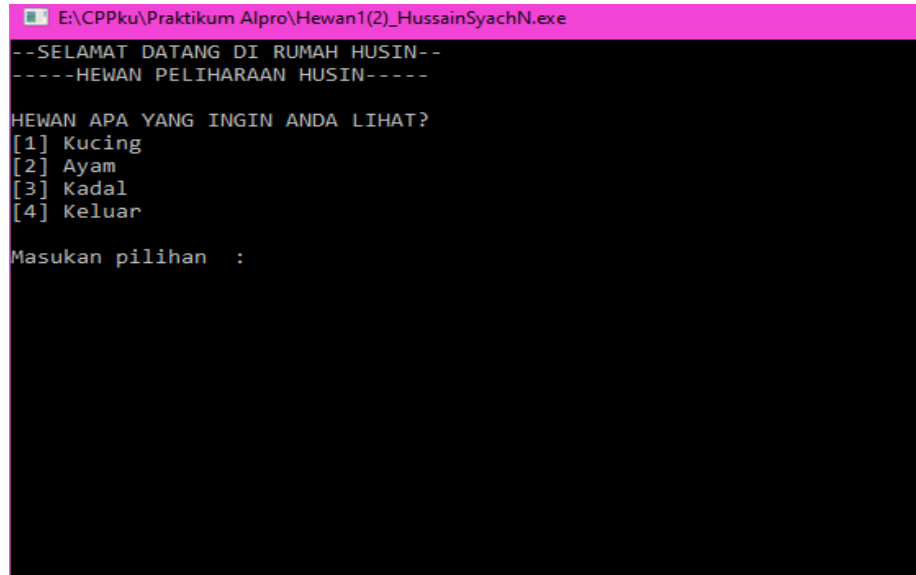
```
        default:{
            cout<<"Pilihan tidak ada!\n"
            <<"Silahkan memilih sesuai nomor yang ada.";
            goto ulang;
            break;
        }
    }
while(pilih!=4);
return 0;
}
```

Gambar 2.11 Switch dan return main program

Statemen default pada struktur switch-case berguna untuk mengeksekusi statemen alternatif, yaitu jika nilai yang kita masukkan ternyata tidak sesuai dengan nilai-nilai konstanta yang telah didefinisikan. Sedangkan statemen break pada struktur switch-case digunakan untuk menunjukkan bahwa perintah siap keluar dari struktur switch-case. Jika pernyataan break tidak ada, maka program akan diteruskan ke pilihan-pilihan berikutnya. Sehingga setiap pilihan akan di cek dan dijalankan jika syarat nilai konstanta terpenuhi, termasuk statemen default juga akan dijalankan jika semua cabang diatasnya tidak memiliki pernyataan break.

Pernyataan GOTO adalah pernyataan yang memungkinkan kita untuk mengatur arahnya aliran pengeksekusian CPU terhadap program kita. GOTO berfungsi untuk memerintahkan CPU melompat ke baris manapun berdasarkan label yang telah dibuat.

Sehingga didapat output sebagai berikut.

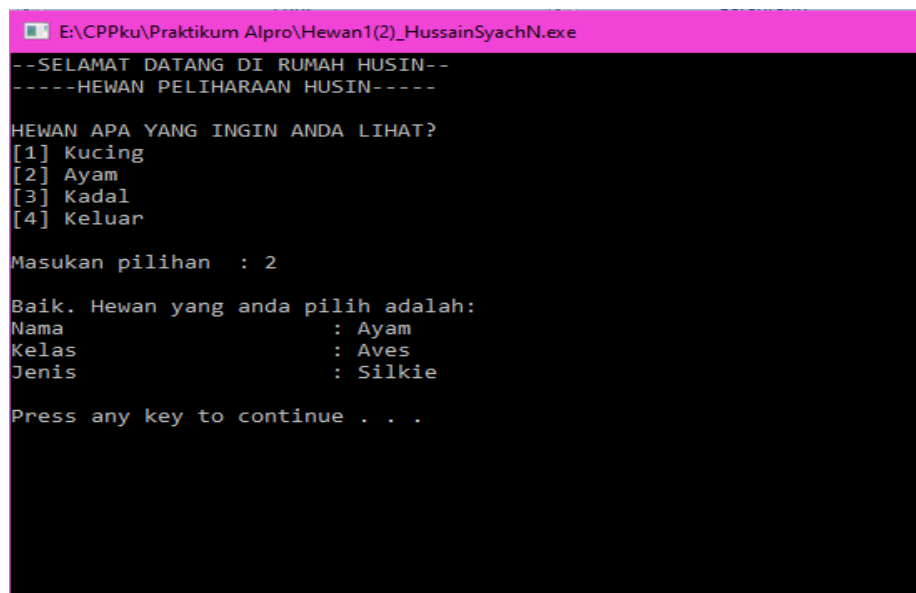


```
E:\CPPku\Praktikum Alpro\Hewan1(2)_HussainSyachN.exe
--SELAMAT DATANG DI RUMAH HUSIN--
-----HEWAN PELIHARAAN HUSIN-----

HEWAN APA YANG INGIN ANDA LIHAT?
[1] Kucing
[2] Ayam
[3] Kadal
[4] Keluar

Masukan pilihan :
```

Gambar 2.12 Output untuk masukkan data



```
E:\CPPku\Praktikum Alpro\Hewan1(2)_HussainSyachN.exe
--SELAMAT DATANG DI RUMAH HUSIN--
-----HEWAN PELIHARAAN HUSIN-----

HEWAN APA YANG INGIN ANDA LIHAT?
[1] Kucing
[2] Ayam
[3] Kadal
[4] Keluar

Masukan pilihan : 2

Baik. Hewan yang anda pilih adalah:
Nama      : Ayam
Kelas    : Aves
Jenis     : Silkie

Press any key to continue . . .
```

Gambar 2.13 Output untuk keluaran data

2.2 Program Kedua Pewarisan Pada Program Hewan

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

using namespace std;

class Hewan{ //Kelas Dasar "Hewan"
    protected:
    char nama[15];
        char jenis[15];
    char habit[15];

    public:
    void infoHewan(){
        cout<<endl;
        cout<<"Berikut adalah data dari Betty"<<endl;
    }
};

class Ayam : public Hewan
{
protected:
    char ciri_khusus[20];
```

Gambar 2.14 Program pewarisan kedua

```

public:
    //konstruktor
    Ayam(char *nm, char *jns, char *hbt, char *cr_k)
    {
        strcpy(nama, nm);
        strcpy(jenis, jns);
        strcpy(habit, hbt);
        strcpy(ciri_khusus, cr_k);
    }

    void infoAyam()
    {
        cout<<"Nama      : "<<nama<<endl;
        cout<<"Jenis      : "<<jenis<<endl;
        cout<<"Sifat       : "<<habit<<endl;
        cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
    }
};

```

Gambar 2.15 Program pewarisan kedua

```

int main()
{
    //menciptakan objek "aym" sekaligus melakukan inisialisasi melalui
    Konstruktor dari Kelas ayam

    Ayam aym("Betty", "Ayam Silkie", "Jinak", "Memiliki bulu yang
    sangat lebat ");

    cout<<"DATA AYAM SILKIE MILIK HUSIN"<<endl;
    cout<<"-----"<<endl;

    aym.infoHewan();
    aym.infoAyam();

    _getche();
    return 0;
}

```

Gambar 2.16 Program pewarisan kedua

Penjelasan :

```

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

using namespace std;

```

Gambar 2.17 File Header

Kode di atas merupakan file header. File header adalah file yang berisi deklarasi untuk berbagai fungsi yang dibutuhkan oleh program baik itu sebagai standar Input/Output maupun sebagai syntax.

```

class Hewan{ //Kelas Dasar "Hewan"
protected:
    char nama[15];
    char jenis[15];
    char habit[15];

public:
    void infoHewan(){
        cout<<endl;
        cout<<"Berikut adalah data dari Betty"<<endl;
    }
};
class Ayam : public Hewan
{
protected:
    char ciri_khusus[20];

```

Gambar 2.18 Class

Kode program di atas merupakan kode yang berjenis class yang bernama *Hewan*. Class adalah salah satu dari konsep OOP yang digunakan untuk membungkus data abstraksi procedural sebagai deskripsi tergeneralisir atau rancangan dari sebuah object untuk mendefinisikan atau menggambarkan isi dan tingkah laku sebagai entitas dari object. Untuk mendirikan Class kita membutuhkan keyword class yang dilanjutkan dengan pemberian nama dari deklarasi class tersebut. lalu dilanjutkan dengan meletakkan tanda { dan } untuk mengapit definisi dari class. Class termasuk sebuah pernyataan maka dari itu akhir dari deklarasi class diwajibkan untuk mengakhiri class menggunakan tanda titik-koma (;). Di dalam kelas ini juga terdapat tipe data yakni char (character). Char adalah salah satu tipe data sembarang huruf, angka, dan simbol. Yang

memungkinkan kita untuk memesan memori yang dapat menyimpan nilai dalam bentuk karakter tunggal seperti 'a', '\$' dan lain-lain.

Terdapat pula keyword public dan protected yang memiliki fungsi sebagai berikut.

1) Public

Keyword public menyatakan bahwa semua anggota yang berada dalam lingkungan public ini dapat diakses di luar kelas yang mendefinisikannya.

2) Protected

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

Terdapat pula kelas turunan atau subclass yang mewarisi superclass yakni class *Ayam*. Kelas turunan (subclass) akan mewarisi data field (properti atau atribut) serta metoda terakses (accessible methods) yang dimiliki superclass. Superclass akan memiliki sifat lebih umum (general), sedangkan kelas turunan akan lebih spesifik. Umumnya, kelas turunan mempunyai data field dan metoda tambahan selain data field dan metoda yang diwarisi dari superclass.

```

public:

    //konstruktor
    Ayam(char *nm, char *jns, char *hbt, char *cr_k)
    {
        strcpy(nama, nm);
        strcpy(jenis, jns);
        strcpy(habit, hbt);
        strcpy(ciri_khusus, cr_k);
    }

    void infoAyam()
    {
        cout<<"Nama      : "<<nama<<endl;
        cout<<"Jenis      : "<<jenis<<endl;
        cout<<"Sifat       : "<<habit<<endl;
        cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
    }

```

Gambar 2.19 Class, konstruktor, dan void

Terdapat pula konstruktor yakni Ayam(char *nm, char *jns, char *hbt, char *cr_k). Konstruktor adalah fungsi anggota dari suatu kelas yang secara otomatis dijalankan ketika suatu objek diciptakan.

Terdapat perintah strcpy yang bisa dipakai untuk menyalin sebuah string ke dalam string lain, atau sebuah string ke dalam variabel. Dalam bahasa C maupun C++ dan sebagainya, strcpy adalah sebuah fungsi atau function.

Terdapat pula void. Void adalah sebuah fungsi (function) yang ada dalam sebuah bahasa pemrograman C, entah itu C++ atau C#. Fungsi ini juga disebut sebagai prosedur (procedure). Fungsi ini tidak mengembalikan nilai keluaran (return output) yang didapat dari hasil proses tersebut, ini kenapa fungsi ini disebut void, secara harfiah berarti kosong.

```

int main()
{
    //menciptakan objek "aym" sekaligus melakukan inisialisasi melalui
    Konstruktor dari Kelas ayam

    Ayam aym("Betty", "Ayam Silkie", "Jinak", "Memiliki bulu yang
    sangat lebat ");

    cout<<"DATA AYAM SILKIE MILIK HUSIN"<<endl;
    cout<<"-----"<<endl;

    aym.infoHewan();
    aym.infoAyam();

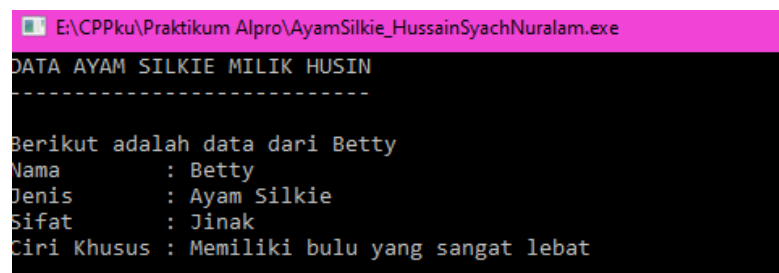
    _getche();
    return 0;
}

```

Gambar 2.20 Main program

Kode di atas merupakan int main(). Int main() artinya main program mengembalikan nilai int secara default, int main() akan mengembalikan nilai 0, dan fungsi main() tidak memiliki bagan deklarasi lokal, dan hanya memiliki sebuah pernyataan yang dapat dieksekusi, berupa fungsi output printf().

Sehingga didapat output sebagai berikut



```

E:\CPPku\Praktikum Alpro\AyamSilkie_HussainSyachNuralam.exe
DATA AYAM SILKIE MILIK HUSIN
-----
Berikut adalah data dari Betty
Nama      : Betty
Jenis     : Ayam Silkie
Sifat     : Jinak
Ciri Khusus : Memiliki bulu yang sangat lebat

```

Gambar 2.21 Output program

KESIMPULAN

Dapat disimpulkan bahwa pewarisan sangat berkaitan erat dengan dasar PBO atau Program Berorientasi Objek. 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. Kelas turunan (subclass) akan mewarisi data field (properti atau atribut) serta metoda terakses (accessible methods) yang dimiliki superclass. Superclass akan memiliki sifat lebih umum (general), sedangkan kelas turunan akan lebih spesifik. Umumnya, kelas turunan mempunyai data field dan metoda tambahan selain data field dan metoda yang diwarisi dari superclass.

DAFTAR PUSTAKA

- Admin. *Memahami Konsep Turunan (Inheritance) Di Java*. 2020.
<https://www.termasmedia.com/pemrograman/java/213-memahami-konsep-turunan-inheritance-di-java.html> (Diakses pada : Selasa, 14 April 2020 Pukul 18.00 WIB)
- Andre. *Tutorial Belajar OOP PHP Part 8: Pengertian Constructor dan Destructor*. 2014.
<https://www.duniailkom.com/tutorial-belajar-oop-php-pengertian-constructor-dan-destructor/> (Diakses pada : Selasa, 14 April 2020 Pukul 18.00 WIB)
- Andre. *Tutorial Belajar OOP PHP Part 9: Pengertian Inheritance (Pewarisan)*. 2014.
<https://www.duniailkom.com/tutorial-belajar-oop-php-pengertian-inheritance-pewarisan/> (Diakses pada : Rabu, 15 April 2020 Pukul 18.00 WIB)
- Andre. *Tutorial Belajar C Part 22: Cara Mengubah Isi String dengan Fungsi strcpy*. 2018.
<https://www.duniailkom.com/tutorial-belajar-c-cara-mengubah-isi-string-dengan-fungsi-strcpy/> (Diakses pada : Rabu, 15 April 2020 Pukul 18.00 WIB)
- Andrians, Rendy. *Perbedaan Int main dan void main*. 2014.
<http://rendyandrians.blogspot.com/2014/08/perbedaan-int-main-dan-void-main.html> (Diakses pada : Rabu, 15 April 2020 Pukul 21.30 WIB)
- Anji. *Public, Protected, Private pada OOP*. 2014.
<http://www.nblognlife.com/2014/04/public-protected-private-pada-oop.html> (Diakses pada : Selasa, 14 April 2020 Pukul 21.00 WIB)
- Dosen Teknik Informatika. *Algoritma dan Pemrograman II*. 2019. Modul Praktikum Algoritma dan Pemrograman II. Universitas Palangka Raya. Fakultas Teknik. Jurusan Teknik Informatika.
- Fajar. *Macam-macam dan Penjelasan Tipe Data dalam C++*. 2018.
<https://www.belajarcpp.com/tutorial/cpp/tipe-data/> (Diakses pada : Rabu, 15 April 2020 Pukul 21.30 WIB)

- Fajar. *Penjelasan Pernyataan Switch*. 2019. <https://www.belajarcpp.com/tutorial/cpp/switch/> (Diakses pada : Kamis, 16 April 2020 Pukul 01.30 WIB)
- Fajar. *Pernyataan Goto*. 2018. <https://www.belajarcpp.com/tutorial/cpp/goto/> (Diakses pada : Kamis, 16 April 2020 Pukul 01.30 WIB)
- MateriDosen. *Pernyataan Switch-Case C++, Lengkap Contoh Program*. 2017. <http://www.materidosen.com/2017/05/pernyataan-switch-case-c-lengkap-contoh.html> (Diakses pada : Kamis, 16 April 2020 Pukul 05.30 WIB)
- Pascal, Dhika. *Macam-macam file header dan fungsinya di bahasa c++*. 2014. http://pascaldhika.blogspot.com/2014/03/macam-macam-file-header-dan-fungsinya_22.html (Diakses pada : Selasa, 14 April 2020 Pukul 18.00 WIB)
- Pratama, Ari. *Apa Itu Void Dan Ciri-Cirinya*. 2015. <https://aripratama.com/apa-itu-void-dan-ciri-cirinya/> (Diakses pada : Selasa, 14 April 2020 Pukul 18.00 WIB)

LAMPIRAN

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

Gambar 1.1 Bentuk penulisan base class

```
class pekerja : public orang{  
    //...  
};
```

Gambar 1.2 Contoh base class pada program

```

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

using namespace std;

class Hewan { //kelas Hewan
    public:
        Hewan(char *a,char *b);
        void tampildata();

    protected:
        char nama[15],kelas[15];
};

Hewan::Hewan(char *a,char *b){ //konstruktor
    strcpy(nama, a);
    strcpy(kelas, b);
}

void Hewan::tampildata(){
    cout<<"Baik. Hewan yang anda pilih adalah:\n";
}

```

Gambar 2.1 Tugas Pewarisan pertama

```

class Kucing:public Hewan { //kelas turunan
    protected:
        char jenis[15];
    public:
        Kucing(char *a,char *b,char *c);
        void tampilkandata();
};

Kucing::Kucing(char *a,char *b, char *c):Hewan(a,b) { //konstruktor kelas
turunan
    strcpy(jenis,c);
}

void Kucing::tampilkandata() {
    Hewan::tampilkandata();
    cout<<"Nama           : "<<nama<<endl;
    cout<<"Kelas           : "<<kelas<<endl;
    cout<<"Jenis             : "<<jenis<<endl<<endl;
}

int main() { //program utama
    int pilih;
    do{
        ulang:
        system("cls");
        cout<<"--SELAMAT DATANG DI RUMAH HUSIN--\n"
        <<"-----HEWAN PELIHARAAN HUSIN-----\n\n"
        <<"HEWAN APA YANG INGIN ANDA LIHAT?\n"

```

Gambar 2.2 Tugas Pewarisan pertama

```

        <<"[1] Kucing\n"
        <<"[2] Ayam\n"
        <<"[3] Kadal\n"
        <<"[4] Keluar\n\n"
        <<"Masukan pilihan : "; cin>>pilih;
cout<<endl;

switch(pilih) {
    case 1:{
        Kucing bb("Kucing","Mamalia","Anggora");
        bb.tampildata();
        system("pause");
        break;
    }
    case 2:{
        Kucing bb("Ayam","Aves","Silkie");
        bb.tampildata();
        system("pause");
        break;
    }
    case 3:{
        Kucing bb("Kadal","Reptil","Iguana");
        bb.tampildata();
        system("pause");
        break;
    }
}

```

Gambar 2.3 Tugas Pewarisan pertama

```

        case 4:{
            system("cls");
            break;
        }
        default:{
            cout<<"Pilihan tidak ada!\n"
            <<"Silahkan memilih sesuai nomor yang ada.";
            goto ulang;
            break;
        }
    }
}
while(pilih!=4);
return 0;
}

```

Gambar 2.4 Tugas Pewarisan pertama

```

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

using namespace std;

```

Gambar 2.5 File Header


```

class Hewan { //kelas Hewan
public:
    Hewan(char *a,char *b);
    void tampildata();

protected:
    char nama[15],kelas[15];
};
Hewan::Hewan(char *a,char *b){ //konstruktor
    strcpy(nama, a);
    strcpy(kelas, b);
}

```

Gambar 2.6 Class

```

void Hewan::tampildata(){
    cout<<"Baik. Hewan yang anda pilih adalah:\n";
}

```

Gambar 2.7 Void

```

class Kucing:public Hewan { //kelas turunan
protected:
    char jenis[15];
public:
    Kucing(char *a,char *b,char *c);
    void tampildata();
};

```

Gambar 2.8 Kelas turunan

```
int main() { //program utama
    int pilih;
    do{
        ulang:
        system("cls");
        cout<<"--SELAMAT DATANG DI RUMAH HUSIN--\n"
        <<"-----HEWAN PELIHARAAN HUSIN-----\n\n"
        <<"HEWAN APA YANG INGIN ANDA LIHAT?\n"
            <<"[1] Kucing\n"
            <<"[2] Ayam\n"
            <<"[3] Kadal\n"
            <<"[4] Keluar\n\n"
            <<"Masukan pilihan : "; cin>>pilih;
        cout<<endl;
```

Gambar 2.9 Main program

```
switch(pilih) {  
    case 1:{  
        Kucing bb("Kucing","Mamalia","Anggora");  
        bb.tampildata();  
        system("pause");  
        break;  
    }  
    case 2:{  
        Kucing bb("Ayam","Aves","Silkie");  
        bb.tampildata();  
        system("pause");  
        break;  
    }  
    case 3:{  
        Kucing bb("Kadal","Reptil","Iguana");  
        bb.tampildata();  
        system("pause");  
        break;  
    }  
    case 4:{  
        system("cls");  
        break;  
    }  
}
```

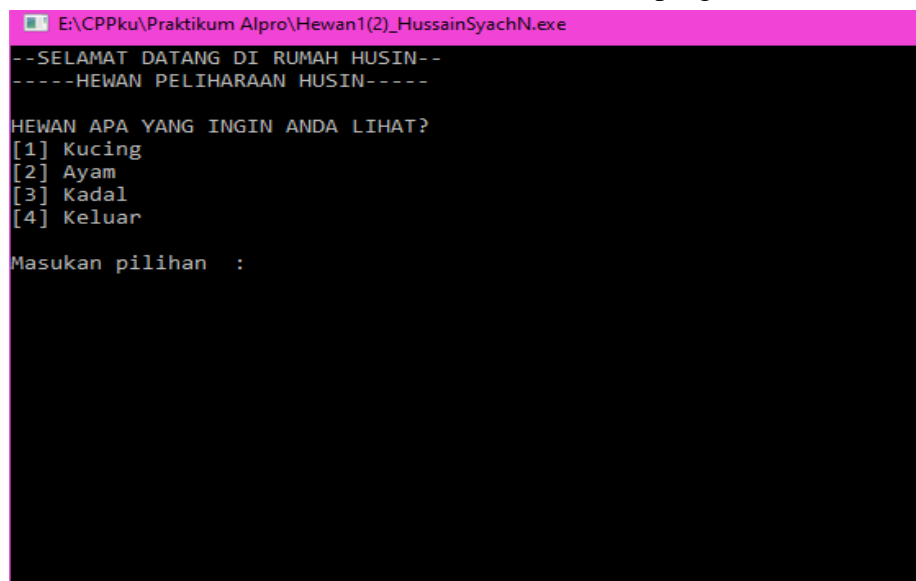
Gambar 2.10 Switch pada main program

```

        default:{
            cout<<"Pilihan tidak ada!\n"
            <<"Silahkan memilih sesuai nomor yang ada.";
            goto ulang;
            break;
        }
    }
}
while(pilih!=4);
return 0;
}

```

Gambar 2.11 Switch dan return main program



```

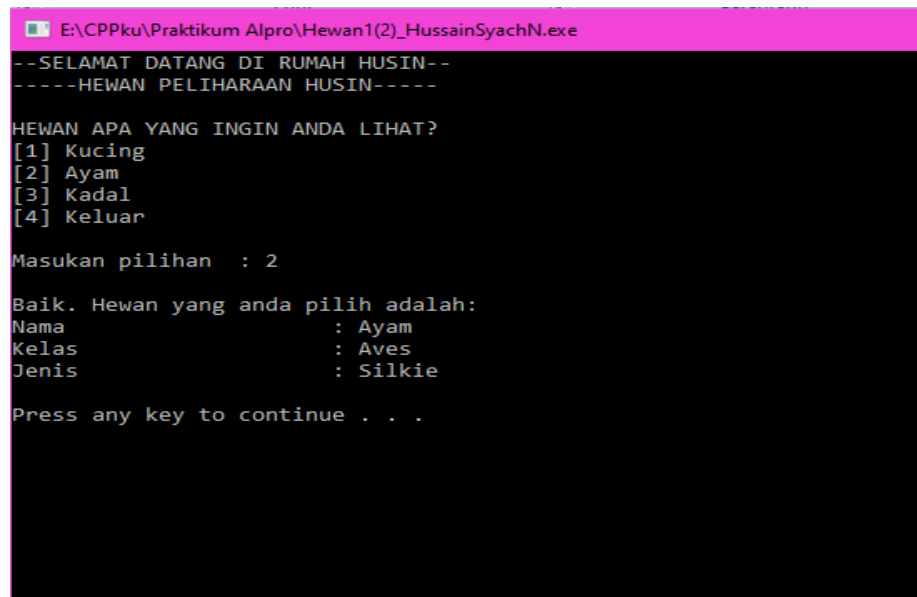
E:\CPPku\Praktikum Alpro\Hewan1(2)_HussainSyachN.exe
--SELAMAT DATANG DI RUMAH HUSIN--
-----HEWAN PELIHARAAN HUSIN-----

HEWAN APA YANG INGIN ANDA LIHAT?
[1] Kucing
[2] Ayam
[3] Kadal
[4] Keluar

Masukan pilihan :

```

Gambar 2.12 Output untuk masukkan data

A screenshot of a Windows command prompt window with a pink title bar. The title bar text is "E:\CPPku\Praktikum Alpro\Hewan1(2)_HussainSyachN.exe". The terminal output is as follows:

```
--SELAMAT DATANG DI RUMAH HUSIN--  
-----HEWAN PELIHARAAN HUSIN-----  
  
HEWAN APA YANG INGIN ANDA LIHAT?  
[1] Kucing  
[2] Ayam  
[3] Kadal  
[4] Keluar  
  
Masukan pilihan : 2  
  
Baik. Hewan yang anda pilih adalah:  
Nama           : Ayam  
Kelas          : Aves  
Jenis           : Silkie  
  
Press any key to continue . . .
```

Gambar 2.13 Output untuk keluaran data

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

using namespace std;

class Hewan{ //Kelas Dasar "Hewan"
    protected:
        char nama[15];
        char jenis[15];
        char habit[15];

    public:
        void infoHewan(){
            cout<<endl;
            cout<<"Berikut adalah data dari Betty"<<endl;
        }
};

class Ayam : public Hewan
{
    protected:
        char ciri_khusus[20];
```

Gambar 2.14 Program pewarisan kedua

```

public:
    //konstruktor
    Ayam(char *nm, char *jns, char *hbt, char *cr_k)
    {
        strcpy(nama, nm);
        strcpy(jenis, jns);
        strcpy(habit, hbt);
        strcpy(ciri_khusus, cr_k);
    }

    void infoAyam()
    {
        cout<<"Nama      : "<<nama<<endl;
        cout<<"Jenis      : "<<jenis<<endl;
        cout<<"Sifat       : "<<habit<<endl;
        cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
    }
};

```

Gambar 2.15 Program pewarisan kedua

```

int main()
{
    //menciptakan objek "aym" sekaligus melakukan inisialisasi melalui
    Konstruktor dari Kelas ayam

    Ayam aym("Betty", "Ayam Silkie", "Jinak", "Memiliki bulu yang
    sangat lebat ");

    cout<<"DATA AYAM SILKIE MILIK HUSIN"<<endl;
    cout<<"-----"<<endl;

    aym.infoHewan();
    aym.infoAyam();

    _getche();
    return 0;
}

```

Gambar 2.16 Program pewarisan kedua

```

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

using namespace std;

```

Gambar 2.17 File Header


```
class Hewan{ //Kelas Dasar "Hewan"
protected:
    char nama[15];
    char jenis[15];
    char habit[15];

public:
    void infoHewan(){
        cout<<endl;
        cout<<"Berikut adalah data dari Betty"<<endl;
    }
};

class Ayam : public Hewan
{
protected:
    char ciri_khusus[20];
```

Gambar 2.18 Class

```

public:

    //konstruktor
    Ayam(char *nm, char *jns, char *hbt, char *cr_k)
    {
        strcpy(nama, nm);
        strcpy(jenis, jns);
        strcpy(habit, hbt);
        strcpy(ciri_khusus, cr_k);
    }

    void infoAyam()
    {
        cout<<"Nama      : "<<nama<<endl;
        cout<<"Jenis      : "<<jenis<<endl;
        cout<<"Sifat       : "<<habit<<endl;
        cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
    }

```

Gambar 2.19 Class, konstruktor, dan void

```

int main()
{
    //menciptakan objek "aym" sekaligus melakukan inisialisasi melalui
    Konstruktor dari Kelas ayam

    Ayam aym("Betty", "Ayam Silkie", "Jinak", "Memiliki bulu yang
    sangat lebat ");

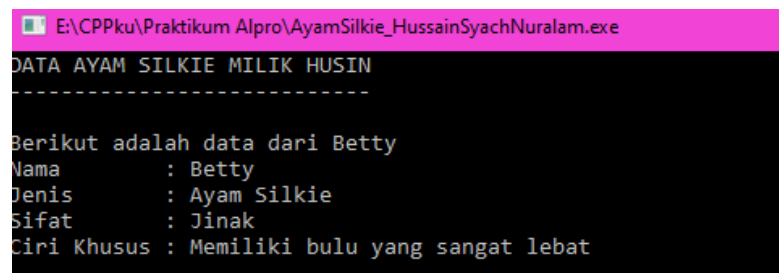
    cout<<"DATA AYAM SILKIE MILIK HUSIN"<<endl;
    cout<<"-----"<<endl;

    aym.infoHewan();
    aym.infoAyam();

    _getche();
    return 0;
}

```

Gambar 2.20 Main program



```

E:\CPPku\Praktikum Alpro\AyamSilkie_HussainSyachNuralam.exe
DATA AYAM SILKIE MILIK HUSIN
-----
Berikut adalah data dari Betty
Nama       : Betty
Denis      : Ayam Silkie
Sifat      : Jinak
Ciri Khusus : Memiliki bulu yang sangat lebat

```

Gambar 2.21 Output program