

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : IRWANDI**  
**NIM : 193030503054**  
**KELAS : A**  
**MODUL : II (PEWARISAN)**

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

**LAPORAN HASIL PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN II**



Nama : Irwandi  
NIM : 193030503054  
Kelas : A  
Modul : II (Pewarisan)

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	5
BAB II Pembahasan	60	40
BAB III Kesimpulan	20	10
Daftar Pustaka	5	3
Lampiran	5	3
Jumlah	100	61

Penilai

Asisten Pratikum

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

1.5

#### II. Landasan Teori

Dalam PBO, kita mengambil realita kehidupan sehari-hari. Kita melakukan pengamatan bahwa manusia secara alami sering melakukan pengelompokan atas objek atau benda. sejauh ini kita mengetahui cara untuk melakukan pengelompokan-pengelompokan atas objek-objek yang serupa (menjadi kelas objek).

Selain melakukan 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 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 ~~x~~ Mobil adalah ("is - a") kendaraan bermotor.
- b ~~x~~ Truk adalah ("is - a") kendaraan bermotor.
- c ~~x~~ Sepeda motor adalah ("is - a") kendaraan bermotor.

Objek yang dikelompokkan dalam suatu 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 atau 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 reuabilitas 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 member kelas turunan. Kelas turunan dapat merujuk/mengakses langsung **public** dan **protected** data member kelas induk dengan menggunakan nama atribut yang diakses.

### **Memahami Konsep Inheritance (Pewarisan) Pada OOP**

#### **I. KONSEP PEWARISAN (INHERITANCE) PADA OOP**

Sebagai Informasi bahwa Ciri khas Pemrograman Berorientasi Objek ada tiga hal yaitu:

1. Enkaptulasi (Pembungkusan)
2. Inheritance (Pewarisan)
3. Polymorphism (Perbedaan Bentuk)

31.5

#### **A. Pengertian Pewarisan (INHERITANCE)**

Konsep inheritance ini mengadopsi dunia riil dimana suatu entitas/obyek dapat mempunyai entitas/obyek turunan. Dengan konsep inheritance, sebuah Class dapat mempunyai class turunan. 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 binatang, 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 classnya masing-masing.

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

## **B. Deklarasi Pewarisan (INHERITANCE)**

Deklarasi yang digunakan adalah dengan menambahkan kata kunci extends setelah deklarasi nama class, kemudian diikuti dengan nama parent class-nya. Kata kunci extends tersebut memberitahu kompiler Java bahwa kita ingin melakukan perluasan class.

```
Public class B extends A {...}
```

Pada saat dikompilasi, Kompiler Java akan membaca sebagai subclass dari class Object.

```
Public class A extends Object {...
```

## BAB II

### PEMBAHASAN

#### 1.1. Program pertama

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

//Kelas Dasar "Hewan"
class Hewan
{
protected:
    char nama[15];
    char habitat[15];
    char namaHewan[15];
    char habittatHewan[15];

public:
    void infoHewan()
    {
        cout<<endl;
        cout<<"Info salah satu jenis hewan"<<endl;
    }
};

class Kucing : public Hewan
{
private:
    char ciri_khusus[20];

public:
    //konstruktor
    Kucing(char *nm, char *hbt, char *cr_k)
    {
        strcpy(nama, nm);
        strcpy(habitat, hbt);
        strcpy(ciri_khusus, cr_k);
    }

    void infoKucing()
    {
        cout<<"Nama      : "<<nama<<endl;
        cout<<"Habitat   : "<<habitat<<endl;
```

```

        cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
    }
};

int main()
{
    //menciptakan objek "kcg" sekaligus melakukan inisialisasi
    melalui Konstruktor dari Kelas Kucing
    Kucing kcg("Kucing", "Rumah","Imut dan bulunya indah");

    cout<<"INHERITANCE PADA C++"<<endl;
    cout<<"-----"<<endl;

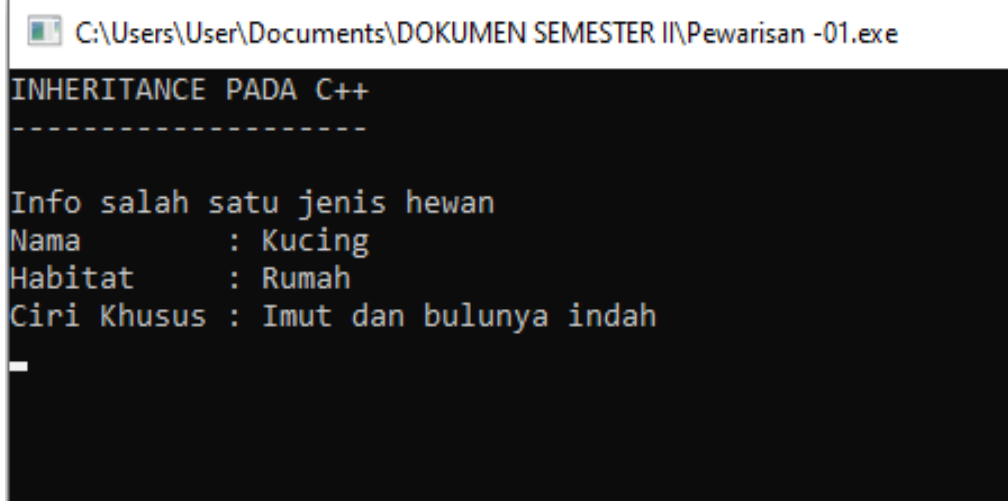
    kcg.infoHewan();
    kcg.infoKucing();

    _getche();
    return 0;
}

```

### 1.1 Program Pertama Pewarisan Class Hewan

Output :



```

C:\Users\User\Documents\DOKUMEN SEMESTER II\Pewarisan -01.exe
INHERITANCE PADA C++
-----
Info salah satu jenis hewan
Nama       : Kucing
Habitat    : Rumah
Ciri Khusus : Imut dan bulunya indah
_

```

### 1.1 Hasil Output Program pertama Pewarisan Class Hewan

Penjelasan :

Seperti terlihat pada program di atas, variabel *nama*, *habitat*, dan fungsi *infoHewan* tidak pernah dibuat pada kelas Kucing namun dapat dikenali oleh kelas Kucing, inilah konsep dari inheritance (Pewarisan).

Seperti yang pernah saya singgung sebelumnya, inilah keuntungan dari inheritance (reuseable). Misalnya kita ingin membuat kelas Burung, burung adalah hewan dan setiap hewan pasti memiliki nama/sebutan dan juga habitat, sehingga kita tidak perlu membuat blok kode lagi, tinggal mewariskan dari kelas Hewan saja. Hal ini secara otomatis juga memunculkan keuntungan lain yaitu dalam hal keefektifan dan keefesienan dalam membuat program.

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

Penjelasan :

Ini adalah header file, atau bisa juga disebut library. File ini sebenarnya kumpulan fungsi-fungsi yang sering digunakan untuk pembuatan program. Fungsi #include adalah untuk memasukkan file library. Sedangkan conio.h dan iostream.h berisi library untuk memanggil fungsi yang kita butuhkan yaitu getch();.

```
public:
    //konstruktor
    Kucing(char *nm, char *hbt, char *cr_k)
    {
        strcpy(nama, nm);
        strcpy(habitat, hbt);
        strcpy(ciri_khusus, cr_k);
    }

    void infoKucing()
    {
        cout<<"Nama      : "<<nama<<endl;
        cout<<"Habitat   : "<<habitat<<endl;
        cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
    }
};
```



Penjelasan :

Pada bagian coding yang selanjutnya **Kucing(char\*nm, char\*hbt, char\*cr\_k)**

Baris selanjutnya pada bagian dari coding tersebut terdapat tanda kurung kurawal ( { ) Kurung kurawal ( { ) menandakan awal program.

Pada baris selanjutnya lagi Bagian program ini merupakan konstruktor untuk class Kucing. Konstruktor ini memiliki dua buah parameter yang tipe datanya float. Pada saat dilakukan pemanggilan, maka akan ditampilkan tulisan "**Konstruktor Kucing dijalankan** " dan nilai dari parameter akan dimasukkan ke dalam variabel x dan y (nilai parameter a dimasukkan ke variabel x dan nilai parameter b yang dimasukkan ke variabel y).

Kemudian bagian coding tersebut diakhiri dengan tanda kurung kurawal ( } ) yang menandakan akhir program.

```
//menciptakan objek "kcg" sekaligus melakukan inisialisasi
melalui Konstruktor dari Kelas Gajah
Kucing kcg("Kucing", "Rumah", "Imut dan bulunya indah");

cout<<"INHERITANCE PADA C++"<<endl;
cout<<"-----"<<endl;

kcg.infoHewan();
kcg.infoKucing();

_getche();
return 0;
}
```

Penjelasan :

Perhatikan potongan program di bawah ini dan output program,

Gajah gjh("Kucing", "Rumah", "Imut dan Bulunya Indah");

Ketika objek gjh diciptakan secara otomatis konstruktor akan dijalankan.

Lalu bagaimana urutannya? Seperti yang terlihat pada ouput program konstruktor dari kelas dasar (Hewan) dijalankan terlebih dahulu.

kemudian baru konstruktor dari kelas turunan (Kucing)

31.5

dijalankan. Destruktor kelas turunan (Kucing) dijalankan terlebih dahulu, kemudian baru destruktur kelas dasar (Hewan) dijalankan.

Perhatikan konstruktor Kucing(char \*nm, char \*hbt, char \*cr\_k)  
: Hewan(nm, hbt). Pernyataan : Hewan(nm, hbt) digunakan untuk melakukan inisialisasi terhadap variabel *nama* dan *habitat* pada kelas Hewan. Penginisialisasian terjadi ketika objek gih diciptakan tepatnya ketika konstruktor pada kelas Hewan dijalankan. Penginisialisasian dilakukan dengan memanfaatkan nilai yang ditangkap oleh variabel pada parameter fungsi, yaitu nm dalam hal ini “Kucing” dan variabel hbt dalam hal ini “Rumah

## 1.2. Program Kedua

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

//Kelas Dasar "Hewan"
class Hewan
{
protected:
    char nama[15];
    char habitat[15];
    char namaHewan[15];
    char habittatHewan[15];

public:
    void infoHewan()
    {
        cout<<endl;
        cout<<"Info salah satu jenis hewan"<<endl;
    }
};

class Tikus : public Hewan
{
private:
    char ciri_khusus[20];

public:
    //konstruktor
    Tikus(char *nm, char *hbt, char *cr_k)
    {
```

```

        strcpy(nama, nm);
        strcpy(habitat, hbt);
        strcpy(ciri_khusus, cr_k);
    }

    void infoTikus()
    {
        cout<<"Nama      : "<<nama<<endl;
        cout<<"Habitat   : "<<habitat<<endl;
        cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
    }
};

int main()
{
    //menciptakan objek "tks" sekaligus melakukan inisialisasi melalui
    Konstruktor dari Kelas Tikus
    Tikus tks("Tikus", "Sawah", "Gigi besar dan kuat");

    cout<<"INHERITANCE PADA C++"<<endl;
    cout<<"-----"<<endl;

    tks.infoHewan();
    tks.infoTikus();

    _getche();
    return 0;
}

```

## 1.2 Program Kedua Pewarisan Class Hewan

Output :

```

C:\Users\User\Documents\DOKUMEN SEMESTER II\tikus irwan.exe
INHERITANCE PADA C++
-----
Info salah satu jenis hewan
Nama      : Tikus
Habitat   : Sawah
Ciri Khusus : Gigi besar dan kuat
_

```

## 1.2 Contoh Hasil Program pewarisan Class hewan

Penjelasan :

Seperti terlihat pada program di atas, variabel *nama*, *habitat*, dan fungsi *infoHewan* tidak pernah dibuat pada kelas Tikus namun dapat dikenali oleh kelas Kucing, inilah konsep dari inheritance (Pewarisan).

Seperti yang pernah saya singgung sebelumnya, inilah keuntungan dari inheritance (reuseable). Misalnya kita ingin membuat kelas Burung, burung adalah hewan dan setiap hewan pasti memiliki nama/sebutan dan juga habitat, sehingga kita tidak perlu membuat blok kode lagi, tinggal mewariskan dari kelas Hewan saja. Hal ini secara otomatis juga memunculkan keuntungan lain yaitu dalam hal keefektifan dan keefesienan dalam membuat program.

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

Penjelasan :

Ini adalah header file, atau bisa juga disebut library. File ini sebenarnya kumpulan fungsi-fungsi yang sering digunakan untuk pembuatan program. Fungsi `#include` adalah untuk memasukkan file library. Sedangkan `conio.h` dan `iostream.h` berisi library untuk memanggil fungsi yang kita butuhkan yaitu `getch()`;

```
public:
    //konstruktor
    Tikus tks(char *nm, char *hbt, char *cr_k)
    {
        strcpy(nama, nm);
        strcpy(habitat, hbt);
        strcpy(ciri_khusus, cr_k);
    }

    void infoTikus()
    {
        cout<<"Nama      : "<<nama<<endl;
        cout<<"Habitat   : "<<habitat<<endl;
        cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
    }
```

```
};
```

Penjelasan :

Pada bagian coding yang selanjutnya **Tikus(char\*nm, char\*hbt, char\*cr\_k)**

Baris selanjutnya pada bagian dari coding tersebut terdapat tanda kurung kurawal ( { ) Kurung kurawal ( { ) menandakan awal program.

Pada baris selanjutnya lagi Bagian program ini merupakan konstruktor untuk class Kucing. Konstruktor ini memiliki dua buah parameter yang tipe datanya float. Pada saat dilakukan pemanggilan, maka akan ditampilkan tulisan "**Konstruktor Tikus dijalankan** " dan nilai dari parameter akan dimasukkan ke dalam variabel x dan y (nilai parameter a dimasukkan ke variabel x dan nilai parameter b yang dimasukkan ke variabel y).

Kemudian bagian coding tersebut diakhiri dengan tanda kurung kurawal ( } ) yang menandakan akhir program.

```
//menciptakan objek "kcg" sekaligus melakukan inisialisasi
melalui Konstruktor dari Kelas Tikus
Kucing kcg("Kucing", "Rumah", "Imut dan bulunya indah");

cout<<"INHERITANCE PADA C++"<<endl;
cout<<"-----"<<endl;

kcg.infoHewan();
kcg.infoKucing();

_getche();
return 0;
}
```

Penjelasan :

Perhatikan potongan program di bawah ini dan output program,

Tikus("Tikus", "Sawah", "Gigi kuat ");

Ketika objek gjh diciptakan secara otomatis konstruktor akan dijalankan.

Lalu bagaimana urutannya? Seperti yang terlihat pada ouput

program kontruktor dari kelas dasar (Hewan) dijalankan terlebih dahulu, kemudian baru konstruktor dari kelas turunan (Kucing) dijalankan. Destruktor kelas turunan (Kucing) dijalankan terlebih dahulu, kemudian baru destruktur kelas dasar (Hewan) dijalankan.

Perhatikan konstruktor Kucing(char \*nm, char \*hbt, char \*cr\_k)  
: Hewan(nm, hbt). Pernyataan : Hewan(nm, hbt) digunakan untuk melakukan inisialisasi terhadap variabel *nama* dan *habitat* pada kelas Hewan. Penginisialisasian terjadi ketika objek gih diciptakan tepatnya ketika konstruktor pada kelas Hewan dijalankan. Penginisialisasian dilakukan dengan memanfaatkan nilai yang ditangkap oleh variabel pada parameter fungsi, yaitu nm dalam hal ini “Kucing” dan variabel hbt dalam hal ini “Rumah

---

### BAB III

### KESIMPULAN

panwghrhl

### 1.3. Kesimpulan

❌ Pewarisan merupakan sebuah Kelas yang menurunkan sifat-sifat class disebut kelas dasar (*base class*), sedangkan yang kelas baru yang mewarisi sifat kelas dasar disebut kelas turunan (*derived class*).

❌ **Float** merupakan tipe data yang menyatakan bilangan pecahan.

❌ fungsi **strcpy** adalah untuk penyalinan(copy) string.

❌ **KONSTRUKTOR** adalah fungsi anggota yang mempunyai nama sama dengan nama kelasnya, sebagai fungsi yang tidak mempunyai nilai balik, harus diletakkan pada bagian public, dijalankan secara otomatis saat objek diciptakan, dapat memiliki argument dan argument dengan nilai default, dapat diletakkan overloading. Kegunaannya : mengalokasikan ruang bagi sebuah objek, memberikan nilai awal terhadap anggota data suatu objek, membentuk tugas-tugas umum lainnya, melakukan tugasnya pada saat objek diciptakan.

❌ **(//)** merupakan sebuah komentar.

#### 1.4 Daftar Fustaka

Modul Praktikum Algoritma dan Pemrograman II, Jurusan Teknik Informatika Universitas Palangkaraya 2015.

<http://mungjava.blogspot.com/2009/05/memahami-konsep-inheritance-pewarisan.html> ( diakses pada hari rabu tanggal 30 april 2015 pukul 23:53 WIB)

<http://www.findamo.com/search.html?q=pewarisan+pada+pemrograman+c++&region=id&lang=en> ( diakses pada hari rabu tanggal 2 mei 2015 pukul 01:00 WIB)



## 1.5 ~~Lampiran~~

### 1. Program Pertama Pewarisan Class Hewan

```
1  #include <iostream>
2  #include <conio.h>
3  #include <string.h>
4  using namespace std;
5
6  //Kelas Dasar "Hewan"
7  class Hewan
8  {
9  protected:
10     char nama[15];
11     char habitat[15];
12     char namaHewan[15];
13     char habittatHewan[15];
14
15 public:
16     void initHewan()
17     {
18         cout<<endl;
19         cout<<"Ini adalah satu jenis hewan"<<endl;
20     }
21 };
22
23 class Kucing : public Hewan
24 {
25 private:
26     char ciri_khusus[20];
27
28 public:
29     //konstruktor
30     Kucing(char *nm, char *hbt, char *cr_k)
31     {
32         strcpy(nama, nm);
33         strcpy(habitat, hbt);
34         strcpy(ciri_khusus, cr_k);
35     }
36 }
```

(coding  
tidak  
di masukkan  
ke lampiran)

```

37 void infoKucing()
38 {
39     cout<<"Nama      : "<<nama<<endl;
40     cout<<"Habitat   : "<<habitat<<endl;
41     cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
42 }
43 };
44
45 int main()
46 {
47     //menciptakan objek "kcg" sekaligus melakukan inisialisasi melalui Konstruktor dari Kelas Gajah
48     Kucing kcg("Kucing", "Rumah", "Imut dan bulunya indah");
49
50     cout<<"INHERITANCE PADA C++"<<endl;
51     cout<<"-----"<<endl;
52
53     kcg.infoHewan();
54     kcg.infoKucing();
55
56     _getche();
57     return 0;
58 }

```

Output :

```

C:\Users\User\Documents\DOKUMEN SEMESTER II\Pewarisan -01.exe
INHERITANCE PADA C++
-----
Info salah satu jenis hewan
Nama      : Kucing
Habitat   : Rumah
Ciri Khusus : Imut dan bulunya indah

```

*Terima kasih.*

## 2. Program Kedua Pewarisan Class Hewan

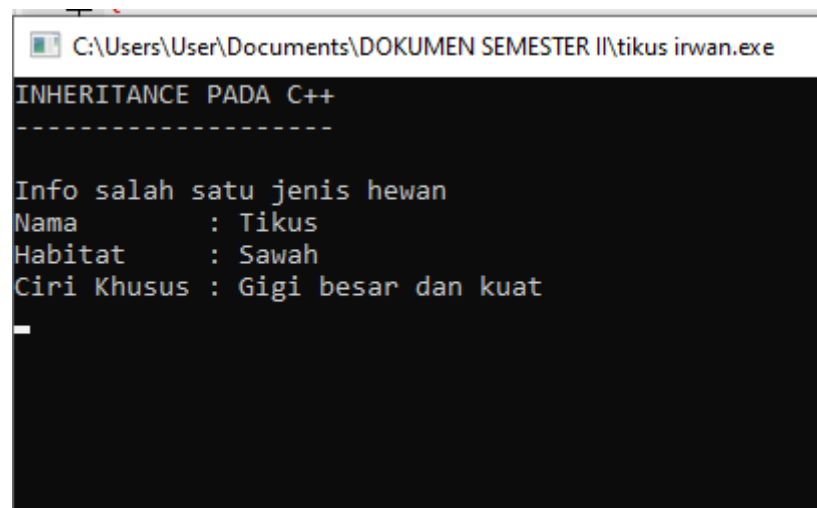
```
1  #include <iostream>
2  #include <conio.h>
3  #include <string.h>
4  using namespace std;
5
6  //Kelas Dasar "Hewan"
7  class Hewan
8  {
9  protected:
10     char nama[15];
11     char habitat[15];
12     char namaHewan[15];
13     char habittatHewan[15];
14
15 public:
16     void infoHewan()
17     {
18         cout<<endl;
19         cout<<"Info salah satu jenis hewan"<<endl;
20     }
21 };
22
23 class Tikus : public Hewan
24 {
25 private:
26     char ciri_khusus[20];
27
28 public:
29     //konstruktor
30     Tikus(char *nm, char *hbt, char *cr_k)
31     {
32         strcpy(nama, nm);
33         strcpy(habitat, hbt);
34         strcpy(ciri_khusus, cr_k);
35     }
36 }
```

```

37     void infoTikus()
38     {
39         cout<<"Nama      : "<<nama<<endl;
40         cout<<"Habitat   : "<<habitat<<endl;
41         cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
42     }
43 };
44
45 int main()
46 {
47     //menciptakan objek "tks" sekaligus melakukan inisialisasi melalui Konstruktor dari Kelas Tikus
48     Tikus tks("Tikus", "Sawah", "Gigi besar dan kuat");
49
50     cout<<"INHERITANCE PADA C++"<<endl;
51     cout<<"-----"<<endl;
52
53     tks.infoHewan();
54     tks.infoTikus();
55
56     _getche();
57     return 0;
58 }

```

Output :



```

C:\Users\User\Documents\DOKUMEN SEMESTER II\tikus irwan.exe
INHERITANCE PADA C++
-----

Info salah satu jenis hewan
Nama      : Tikus
Habitat   : Sawah
Ciri Khusus : Gigi besar dan kuat

```

C:\Users\User\Documents\DOKUMEN SEMESTER II\hewan gajah.exe

\*=====HASIL DATA DARI HEWAN=====\*

nama hewan : gajahbergading  
jenis hewan : bergading  
ciri hewan : belalai besar



