

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : YUDHA ARTHA NUGRAHA**  
**NIM : 193030503045**  
**KELAS : A**  
**MODUL : PEWARISAN**

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

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



Nama : **YUDHA ARTHA NUGRAHA**  
NIM : **193030503045**  
Kelas : **A**  
Modul : **PEWARISAN**

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	7
BAB II Pembahasan	60	50
BAB III Kesimpulan	20	10
Daftar Pustaka	5	5
Lampiran	5	5
Jumlah	100	77

Penilai  
Asisten Praktikum

# BAB I

## LANDASAN TEORI

### 1.1. TUJUAN

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

### 1.1. 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.

## BAB II

### PEMBAHASAN

2.1. Buatlah konsep pbo menggunakan konstruktor dan protected !

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

class kucing{
    public:
        kucing(char *k,char *e,char *t);
        void info();
    protected:
        char kaki[2], ekor[2], telinga[2];
};

kucing::kucing(char *k, char *e, char *t){
    strcpy(kaki,k);
    strcpy(ekor,e);
    strcpy(telinga,t);
}

void kucing::info() {
    cout<<"Menampilkan informasi hewan :\n"
         <<"-----\n\n";
}

class K:public kucing {
    protected:
        char status[20];
    public:
        K(char *k, char *e, char *t, char *s);
        void info();
};

K::K(char *k,char *e, char *t, char *s):kucing(k,e,t) {
    strcpy(status,s);
}
```

```

void K::info() {
    kucing::info();
    cout<<"Jumlah kaki kucing  : "<<kaki<<endl;
    cout<<"Jumlah ekor kucing  : "<<ekor<<endl;
    cout<<"Jumlah telinga kucing : "<<telinga<<endl;
    cout<<"Status kucing      : "<<status<<endl<<endl;
}

int main(){
    K kcg("4", "1", "2", "karnivora");

    kcg.info();

    system("pause");
    return 0;
}

```

Pembahasan :

Disini diketahui singkat

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).
3. Private Private menyatakan bahwa semua anggota yang berada dalam lingkungan private hanya dapat diakses oleh kelas yang mendefinisikannya. "Private dapat diibaratkan dengan suatu klub yang diperuntukkan bagi anggotanya saja. Semua fasilitas klub dapat diakses oleh anggota klub. Tetapi tidak demikian halnya dengan orang luar (bukan anggota klub)."

Using Namespace Std adalah sebuah intruksi terhadap compiler untuk menggunakan semua fungsi yang terkait untuk kerangka yang sama, bisa berupa berkas, class, dan sejenisnya yang berkaitan dengan std. sehingga fungsinya ialah berguna untuk meminimalisir kesalahan akibat tidak dikenalnya fungsi cout dan cin apabila dalam penulisan secara langsung.

Pada umumnya, penulisan perintah cout dan cin sebenarnya adalah std::cout atau std::cin namun dengan adanya penggunaan Using Namespace Std sehingga kamu tidak perlu menuliskan std:: lagi, contoh tersebut sangat singkat dan mungkin anda berfikir tidak terlalu repot.

```
class kucing{
    public:
        kucing(char *k,char *e,char *t);
        void info();
    protected:
        char kaki[2], ekor[2], telinga[2];
};
```

Program diatas adalah class kucing dalam class kucing terdapat public yang dapat di akses secara umum dan protected. Dalam clas terdapat char \*k,char \*e,char \*t sebagai variabel sekaligus pointer, juga protected char kaki,ekor,dan telinga

```
kucing::kucing(char *k, char *e, char *t){
    strcpy(kaki,k);
    strcpy(ekor,e);
    strcpy(telinga,t);
}
```

Pada program ini digunakan untuk mengubah char kedalam bentuk string menggunakan *strcpy* . disini juga digunakan untuk menentukan kaki sebagai k, ekor sebagai e dan telinga sebagai t.

```
void kucing::info() {
    cout<<"Menampilkan informasi hewan :\n"
    <<"-----\n\n";
}
```

Pada program ini untuk menampilkan isi dari cout

```

class K:public kucing {
    protected:
        char status[20];
    public:
        K(char *k, char *e, char *t, char *s);
        void info();
};

K::K(char *k,char *e, char *t, char *s):kucing(k,e,t) {
    strcpy(status,s);
}

```

Program ini merupakan class turunan dari class induk pada class ini digunakan untuk menambahkan char status . pada bagian bawah class turunan ada fungsi untuk mengenali status sebagai char \*s dan tidak lupa pula menyematkan fungsi sebelumnya yaitu variabel K,E, dan T .

```

void K::info() {
    kucing::info();
    cout<<"Jumlah kaki kucing   : "<<kaki<<endl;
    cout<<"Jumlah ekor kucing   : "<<ekor<<endl;
    cout<<"Jumlah telinga kucing : "<<telinga<<endl;
    cout<<"Status kucing       : "<<status<<endl<<endl;
}

```

Pada program ini akan menampilkan data dari program yang akan di tampilkan di output. Data yang di tampilkan berupa :

Jumlah kaki  
 Jumlah ekor  
 Jumlah telunga dan status

```

int main(){
    K kcg("4", "1", "2", "karnivora");

    kcg.info();

    system("pause");
    return 0;
}

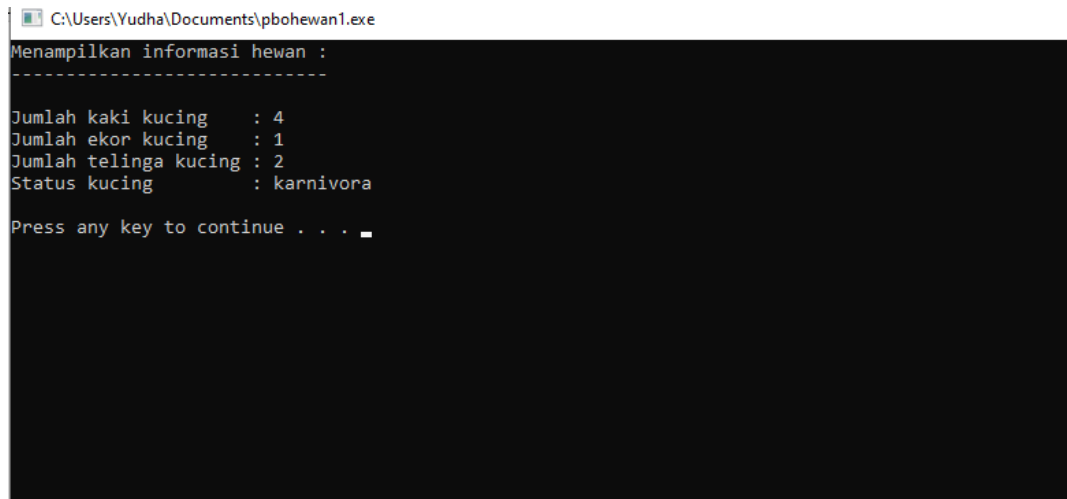
```

Pada program di atas digunakan sebagai program untuk menjalankan semua yang di rencanakan. Dalam int main() telah disematkan jawaban dari void



K::info yaitu `K kg("4", "1", "2", "karnivora")`; disini yang menentukan program akan memiliki jawaban yang diinginkan.

Contoh output :



```
C:\Users\Yudha\Documents\pbohewan1.exe
Menampilkan informasi hewan :
-----
Jumlah kaki kucing      : 4
Jumlah ekor kucing      : 1
Jumlah telinga kucing   : 2
Status kucing           : karnivora
Press any key to continue . . .
```

**Gambar 2.1** hasil output pbo hewan contoh 1

2.2. Buatlah konsep pbo menggunakan konstruktor dan protected contoh kedua !

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

class komodo{
public:
    komodo(char *k,char *e,char *t);
    void info();
protected:
    char kaki[2], ekor[2], c_k[20];
};

komodo::komodo(char *k, char *e, char *c){
    strcpy(kaki,k);
    strcpy(ekor,e);
    strcpy(c_k,c);
}

void komodo::info() {
    cout<<"===== hewan
===== \n"
    <<"-----\n\n";
}

class K:public komodo {
protected:
    char status[20];
public:
    K(char *k, char *e, char *c, char *s);
    void info();
};

K::K(char *k,char *e, char *c, char *s):komodo(k,e,c) {
    strcpy(status,s);
}
```

```

void K::info() {
    komodo::info();
    cout<<"Jumlah kaki komodo   : "<<kaki<<endl;
    cout<<"Jumlah ekor komodo   : "<<ekor<<endl;
    cout<<"Ciri khusus komodo   : "<<c_k<<endl;
    cout<<"Status komodo       : "<<status<<endl;
    cout<<"=====
===== \n"<<endl;
}

int main(){
    K kcg("4", "1", "Berbisa", "Hampir punah");

    kcg.info();

    system("pause");
    return 0;
}

```

Pembahasan :

```

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

```

Using Namespace Std adalah sebuah intruksi terhadap compiler untuk menggunakan semua fungsi yang terkait untuk kerangka yang sama, bisa berupa berkas, class, dan sejenisnya yang berkaitan dengan std. sehingga fungsinya ialah berguna untuk meminimalisir kesalahan akibat tidak dikenalnya fungsi cout dan cin apabila dalam penulisan secara langsung.

Pada umumnya, penulisan perintah cout dan cin sebenarnya adalah std::cout atau std::cin namun dengan adanya penggunaan Using Namespace Std sehingga kamu tidak perlu menuliskan std:: lagi, contoh tersebut sangat singkat dan mungkin anda berfikir tidak terlalu repot.

```

class komodo{
    public:
        komodo(char *k,char *e,char *t);
        void info();
    protected:
        char kaki[2], ekor[2], c_k[20];
};

```

Program diatas adalah class komodo dalam class komodo terdapat public yang dapat di akses secara umum dan protected. Dalam clas terdapat char \*k,char \*e,char \*t sebagai variabel sekaligus pointer, juga protected char kaki,ekor,dan telinga

```
void komodo::info() {
    cout<<"===== hewan
===== \n"
    <<"-----\n\n";
}
```

pada program ini untuk menampilkan isi dari cout

```
class K:public komodo {
    protected:
        char status[20];
    public:
        K(char *k, char *e, char *c, char *s);
        void info();
};

K::K(char *k,char *e, char *c, char *s):komodo(k,e,c) {
    strcpy(status,s);
}
```

Program ini merupakan class turunan dari class induk pada class ini digunakan untuk menambahkan char status . pada bagian bawah class turunan ada fungsi untuk mengenali status sebagai char \*s dan tidak lupa pula menyematkan fungsi sebelumnya yaitu variabel K,E, dan c .

```
void K::info() {
    komodo::info();
    cout<<"Jumlah kaki komodo   : "<<kaki<<endl;
    cout<<"Jumlah ekor komodo   : "<<ekor<<endl;
    cout<<"Ciri khusus komodo   : "<<c_k<<endl;
    cout<<"Status komodo       : "<<status<<endl;
    cout<<"=====
===== \n"<<endl;
}
```

Pada program ini akan menampilkan data dari program yang akan di tampilkan di output. Data yang di tampilkan berupa :

Jumlah kaki :  
Jumlah ekor :  
Ciri khusus :  
Status :

```
int main(){
    K kcg("4", "1", "Berbisa", "Hampir punah");

    kcg.info();

    system("pause");
    return 0;
}
```

Pada program di atas disematkan jawaban untuk *void K::info()* yang isinya *K kcg("4", "1", "Berbisa", "Hampir punah");* disini yang menentukan program akan memiliki jawaban yang diinginkan.

Seperti bahasa pemrograman yang menggunakan konsep OOP lainnya, C++ juga dapat melakukan inheritance atau pewarisan. Inheritance atau pewarisan memungkinkan variabel atau fungsi anggota dari suatu kelas diwariskan kelas lain. Kelas yang mewariskan disebut kelas dasar, sedangkan kelas yang diwarisi disebut kelas turunan.

Salah satu keuntungan penggunaan konsep inheritance adalah membuat suatu blok kode menjadi reuseable. Hal ini memungkinkan blok kode yang pernah dibuat dapat digunakan kembali.

Contoh output :



```
C:\Users\Yudha\Documents\pbohewan2.exe
===== hewan =====
-----
Jumlah kaki komodo      : 4
Jumlah ekor komodo      : 1
Ciri khusus komodo      : Berbisa
Status komodo           : Hampir punah
=====
Press any key to continue . . .
```

**Gambar 2.2** hasil output pbo hewan contoh 2

## **BAB III**

### **KESIMPULAN**

Seperti bahasa pemrograman yang menggunakan konsep OOP lainnya, C++ juga dapat melakukan inheritance atau pewarisan. Inheritance atau pewarisan memungkinkan variabel atau fungsi anggota dari suatu kelas diwariskan kelas lain. Kelas yang mewariskan disebut kelas dasar, sedangkan kelas yang diwarisi disebut kelas turunan.

Salah satu keuntungan penggunaan konsep inheritance adalah membuat suatu blok kode menjadi reusable. Hal ini memungkinkan blok kode yang pernah dibuat dapat digunakan kembali.

Public adalah fungsi yang dapat diakses oleh umum atau dapat diakses oleh siapa saja. Public adalah fungsi yang dapat diakses oleh umum atau dapat diakses oleh siapa saja.

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).
3. Private Private menyatakan bahwa semua anggota yang berada dalam lingkungan private hanya dapat diakses oleh kelas yang mendefinisikannya. "Private dapat diibaratkan dengan suatu klub yang diperuntukkan bagi anggotanya saja. Semua fasilitas klub dapat diakses oleh anggota klub. Tetapi tidak demikian halnya dengan orang luar (bukan anggota klub)."

## DAFTAR PUSTAKA

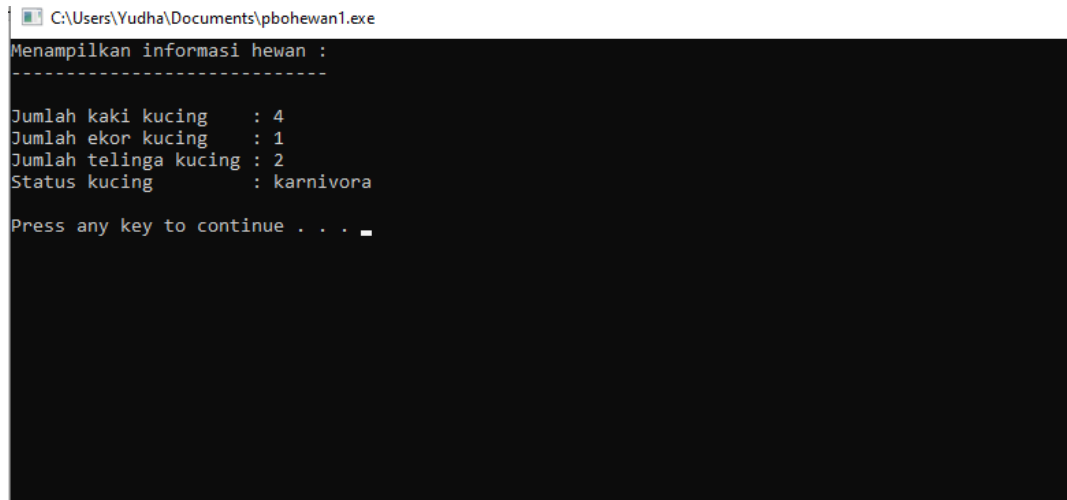
dosen teknik informatika.2020. MODUL PRAKTIKUM ALGORITMA PEMROGRAMAN II

pahlevi, rizka reza .2014. Penggunaan Private dan Public dalam C++.[http://jagocoding.com/tutorial/920/Penggunaan\\_Private\\_dan\\_Public\\_dalam\\_C](http://jagocoding.com/tutorial/920/Penggunaan_Private_dan_Public_dalam_C). (16 april 2020)

unknown. 2014. Public, Protected, Private pada OOP. <http://www.nblognlife.com/2014/04/public-protected-private-pada-oop.html>. (16 april 2020)

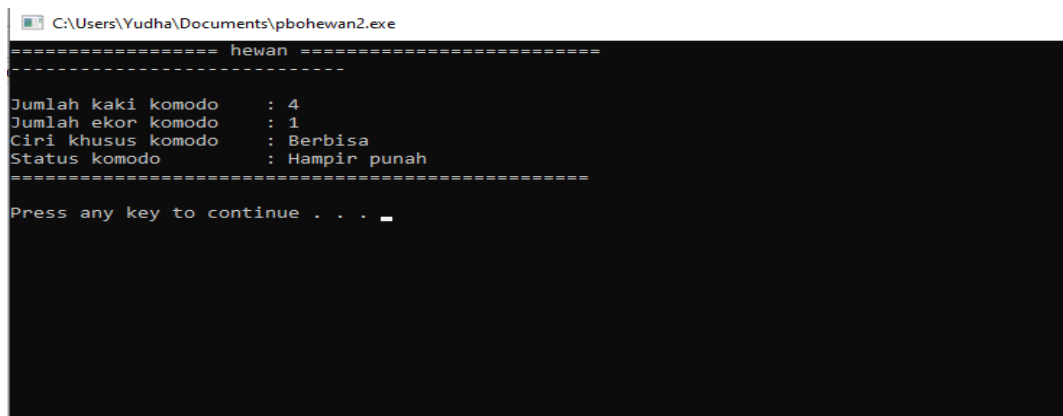
santoso, rachmat. 2017. C++ - Inheritance (Dasar Pewarisan) <http://www.nblognlife.com/2017/04/c-inheritance-dasar-pewarisan.html>. (16 april 2020)

## LAMPIRAN



```
C:\Users\Yudha\Documents\pbohewan1.exe
Menampilkan informasi hewan :
-----
Jumlah kaki kucing      : 4
Jumlah ekor kucing      : 1
Jumlah telinga kucing   : 2
Status kucing           : karnivora
Press any key to continue . . .
```

**Gambar 2.1** hasil output pbo hewan contoh 1



```
C:\Users\Yudha\Documents\pbohewan2.exe
===== hewan =====
-----
Jumlah kaki komodo      : 4
Jumlah ekor komodo      : 1
Ciri khusus komodo     : Berbisa
Status komodo           : Hampir punah
=====
Press any key to continue . . .
```

**Gambar 2.2** hasil output pbo hewan contoh 2