

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : JORGI JACKO EXCEL**  
**NIM : 193030503064**  
**KELAS : A**  
**MODUL : PEWARISAN**

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

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



Nama : Jorgi Jacko Excel  
NIM : 193030503064  
Kelas : A  
Modul : II ( PEWARISAN )

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	8
BAB II Pembahasan	60	48
BAB III Kesimpulan	20	10
Daftar Pustaka	5	5
Lampiran	5	4
Jumlah	100	75

Penilai  
Asisten Praktikum

Diana

## **BAB I**

### **TUJUAN DAN LANDASAN TEORI**

#### **1. TUJUAN**

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

#### **2. LANDASAN TEORI**

Pewarisan (inheritance) adalah penurunan sifat yang ada pada suatu kelas kepada kelas baru yang menjadi turunannya. Sifat-sifat suatu kelas ditentukan oleh data anggota dan metode, sehingga yang dimaksud penurunan sifat adalah penurunan data anggota atau metode. Kelas yang menurunkan sifat disebut kelas dasar (base class), sedangkan yang kelas baru yang mewarisi sifat kelas dasar disebut kelas turunan (derived class).

Suatu kelas dapat diciptakan berdasarkan kelas lain. Kelas baru ini mempunyai sifat-sifat yang sama dengan kelas pembentuknya, ditambah sifat-sifat khusus lainnya. Dengan pewarisan kita dapat menciptakan kelas baru yang mempunyai sifat sama dengan kelas lain tanpa harus menulis ulang bagian-bagian yang sama. Pewarisan merupakan unsur penting dalam pemrograman berorientasi objek dan merupakan blok bangunan dasar pertama penggunaan kode ulang (code reuse). Jika tidak ada fasilitas pewarisan ini, maka pemrograman dalam C++ akan tidak banyak berbeda dengan pemrograman C, hanya perbedaan dalam pengkapsulan saja yang menggunakan kelas sebagai pengganti struktur. Yang perlu menjadi catatan di sini adalah bahwa data dan fungsi yang dapat diwariskan hanya yang bersifat public dan protected. Untuk data dan fungsi private tetap tidak dapat diwariskan. Hal ini disebabkan sifat protected yang hanya dapat diakses dari dalam kelas saja.

Pewarisan dikelompokkan menjadi 3, yaitu:

##### **2.1 Pewarisan Tunggal (single inheritance)**

Pewarisan Tunggal (single inheritance) adalah pewarisan yang mana jumlah kelas dasarnya tunggal. Pada pewarisan ini, kelas turunan

dapat berjumlah lebih dari satu. Pewarisan tunggal dapat digambarkan dengan sintak program sebagai berikut:

```
class A
{
...
};
class B : public A
{
...
}
```

Sintak di atas adalah mekanisme pewarisan secara public. Dengan implementasi di atas, kelas B merupakan kelas turunan dari kelas A. Selain pewarisan public, pewarisan juga dilakukan secara protected maupun private.

## 2.2 Pewarisan Jamak (multiple inheritance)

Pewarisan Jamak (multiple inheritance) adalah pewarisan dimana satu kelas diturunkan lebih dari satu kelas yang berbeda. Dalam pewarisan ini jumlah kelas dasarnya lebih dari satu, dan perlu dicatat bahwa kelas dasarnya bukan merupakan turunan dari satu kelas. Kelas turunan mewarisi seluruh sifat dari kelas dasarnya, sehingga sifat dari beberapa kelas dasar dan sifat khas dirinya. Perhatikan sintak dari pewarisan tunggal berikut ini:

```
class A
{
...
};
class B
{
...
}
class C: public A, public B
```

```
{  
...  
}
```

Pada bentuk tersebut terdapat dua kelas dasar yaitu kelas A dan kelas B yang menurunkan kelas C. Kelas C akan mewarisi sifat dari kelas A maupun sifat dari kelas B tetapi tidak berlaku sebaliknya.

### 2.3 Pewarisan Jamak Maya (virtual multiple inheritance)

Pewarisan Jamak Maya (virtual multiple inheritance) adalah pewarisan yang mana kelas dasarnya lebih dari satu dan beberapa di antara kelas dasar tersebut merupakan kelas turunan dari kelas dasar yang sama. Mekanisme pewarisan sifat suatu kelas dasar kepada kelas turunan sama dengan pewarisan yang lain. Perhatikan sintak berikut ini:

```
class A  
{  
...  
};  
class B: virtual public A  
{  
...  
};  
class C: virtual public A  
{  
...  
};  
class D: public B, public C  
{  
...  
};
```

Kelas D merupakan turunan dari kelas B dan C, sedangkan kelas B dan C merupakan kelas turunan dari kelas dasar yang sama yaitu kelas A. Supaya berjalan pewarisan dari kelas A kepada kelas B maupun C harus

secara virtual. Kelas virtual A pertama menurunkan kelas B, sedangkan kelas virtual A kedua menurunkan kelas C.

### **Konstruktor dan Destruktor pada Pewarisan**

Didalam pewarisan konstruktor perlu diperhatikan terutama konstruktor penyalinan. Jika konstruktor kelas dasar hanya berisi pernyataan memberi nilai data anggota private, maka akses private dapat diganti dengan akses protected agar data anggota pada kelas dasar dapat diakses dari kelas turunan. Pada saat objek berkelas turunan diciptakan konstruktor kelas dasar dengan sendirinya dijalankan, kemudian konstruktor kelas turunan baru dijalankan. Pada saat objek berakhir destruktur kelas turunan dijalankan terlebih dahulu, kemudian destruktur kelas dasar dijalankan.

## BAB II

### PEMBAHASAN

#### 1. Membuat program pewarisan buah

Pembahasan Program:

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

**#Include "iostream.h"**

Menjalankan perintah-perintah berikut :

- a. Cout merupakan fungsi keluaran yang digunakan untuk menampilkan data ataupun informasi.
- b. Cin merupakan fungsi masukan yang digunakan untuk menyimpan data dalam suatu variable.
- c. Endl merupakan fungsi yang digunakan untuk perpindahan baris.

#### **Using namespace std**

digunakan untuk memanggil namespace yang memiliki nama 'std'. Namespace 'std' merupakan standar namespace dari C++ yang dapat kita gunakan untuk memanggil class/object/fungsi yang terdapat didalam namespace tersebut.

```
class buah
{
protected:
    void jenis();
    void warna();
    void rasa();

public:
    char jeruk1;
    char jeruk2;
    char jeruk3;
};
```

#### **Class dasar buah**

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.

#### **Protected**

Dimana member-member dibawah label ini seperti void jenis, warna, dan rasa dapat diakses oleh kelas turunan dari kelas tersebut. Untuk turunan kelas akan dibahas kemudian.

### **Public**

Merupakan hak akses maksimum, dimana fungsi dan kelas apapun selama masih dalam satu program dapat mengakses member yang bernaung di bawah label public. Di dalam public ini terdapat jeruk1, jeruk2, jeruk3 yang bertipe char yang nanti nya berfungsi sebagai turunan ataupun konstruktor

```
class jeruk1
{
    public:
    void jenis();
};
class jeruk2 : public jeruk1
{
    public:
    void rasa();
};
class jeruk3 : public jeruk2
{
    public:
    void warna();
};
```

### **Class jeruk 1**

Menurunkan sifat dasar yaitu class buah dimana disini yang diturunkan adalah sifat jenis

### **Class jeruk 2**

Menurunkan sifat class pertama dimana disini yang diturunkan adalah sifat jenis dan ditambah dengan sifat class dasar yaitu rasa

### **Class jeruk 3**

Menurunkan sifat class kedua dimana disini yang diturunkan adalah sifat jenis, dan rasa dan ditambah dengan sifat class dasar yaitu warna

```
int main()
{
    jeruk1 jrk; jeruk2 jerk; jeruk3 jruk;
    cout<<endl<<"Buah Jeruk 1 : "<<endl;
    jrk.jenis();
    cout<<endl<<"Rasa Jeruk 2 : "<<endl;
```



```

jrk.jenis(); jerk.rasa();
cout<<endl<<"Buah Jeruk 3 : "<<endl;
jrk.jenis(); jerk.rasa(); jruk.warna();
cout<<endl<<"Buah Jeruk dasar : "<<endl;
jrk.jenis(); jerk.rasa(); jruk.warna();
}

```

Program ini akan menampilkan output dimana pada buah 1 akan menghasilkan jenis buah pada class pertama, buah 2 akan menampilkan jenis dan rasa seperti pada class kedua, buah 3 akan menampilkan jenis, rasa, dan warna pada seperti class ketiga pada program

Disini juga memuat class induk yaitu class awal, yang akan diwariskan ke anak-anak kelas selanjutnya baik anak pertama, kedua, dan ketiga

```

void jeruk1::jenis()
{
    cout<<" Jeruk Mandarin "<<endl;
}
void jeruk2::rasa()
{
    cout<<" Manis "<<endl;
}
void jeruk3::warna()
{
    cout<<" Orange "<<endl;
}

```

Berikut ada adalah konstruktor yang berfungsi untuk menginputkan data pada konstruktor pertama terdapat void jeruk1::jenis() yang diberi inputan yaitu “jeruk mandarin” sebagai jenis, Void jeruk2::rasa() yang diberi inputan “manis” sebagai rasa, Void jeruk3::warna() yang diberi inputan “orange” sebagai warna. Dimana data yang di input nanti akan di tampilkan pada output program

2. Membuat program pewarisan buah sama seperti program pertama tetapi dengan jenis buah yang berbeda.

Pembahasan Program:

```

#include<iostream>
using namespace std;

```

**#Include "iostream.h"**

Menjalankan perintah-perintah berikut :

- d. Cout merupakan fungsi keluaran yang digunakan untuk menampilkan data ataupun informasi.

- e. Cin merupakan fungsi masukan yang digunakan untuk menyimpan data dalam suatu variable.
- f. Endl merupakan fungsi yang digunakan untuk perpindahan baris.

### Using namespace std

digunakan untuk memanggil namespace yang memiliki nama 'std'. Namespace 'std' merupakan standar namespace dari C++ yang dapat kita gunakan untuk memanggil class/object/fungsi yang terdapat didalam namespace tersebut.

```
class buah
{
protected:
    void jenis();
    void warna();
    void rasa();

public:
    char jambu1;
    char jambu2;
    char jambu3;
};
```

### Class dasar buah

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.

### Protected

Dimana member-member dibawah label ini seperti void jenis, warna, dan rasa dapat diakses oleh kelas turunan dari kelas tersebut. Untuk turunan kelas akan dibahas kemudian.

### Public

Merupakan hak akses maksimum, dimana fungsi dan kelas apapun selama masih dalam satu program dapat mengakses member yang bernaung di bawah label public. Di dalam public ini terdapat jambu1, jambu2, jambu3 yang bertipe char yang nanti nya berfungsi sebagai turunan ataupun konstruktor

```
class jambu1
{
```

```

public:
void jenis();
};
class jambu2 : public jambu1
{
public:
void rasa();
};
class jambu3 : public jambu2
{
public:
void warna();
};

```

### **Class jambu 1**

Menurunkan sifat dasar yaitu class buah dimana disini yang diturunkan adalah sifat jenis

### **Class jambu 2**

Menurunkan sifat class pertama dimana disini yang diturunkan adalah sifat jenis dan ditambah dengan sifat class dasar yaitu rasa

### **Class jambu 3**

Menurunkan sifat class kedua dimana disini yang diturunkan adalah sifat jenis, dan rasa dan ditambah dengan sifat class dasar yaitu warna

```

int main()
{
jambu1 jmu; jambu2 jbu; jambu3 jam;
cout<<endl<<"Jenis Jambu : "<<endl;
jmu.jenis();
cout<<endl<<"Rasa Jambu : "<<endl;
jmu.jenis(); jbu.rasa();
cout<<endl<<"Warna Buah Jambu : "<<endl;
jmu.jenis(); jbu.rasa(); jam.warna();
cout<<endl<<"Buah Jambu : "<<endl;
jmu.jenis(); jbu.rasa(); jam.warna();
}

```

Program ini akan menampilkan output dimana pada buah 1 akan menghasilkan jenis buah pada class pertama, buah 2 akan menampilkan jenis dan rasa seperti pada class kedua, buah 3 akan menampilkan jenis, rasa, dan warna pada seperti class ketiga pada program

Disini juga memuat class induk yaitu class awal, yang akan diwariskan ke anak-anak kelas selanjutnya baik anak pertama, kedua, dan ketiga

```
void jambu1::jenis()
{
    cout<<" Jambu biji "<<endl;
}
void jambu2::rasa()
{
    cout<<" Manis "<<endl;
}
void jambu3::warna()
{
    cout<<" kuning "<<endl;
}
```

Berikut ada adalah konstruktor yang berfungsi untuk menginputkan data pada konstruktor pertama terdapat void jambu1::jenis() yang diberi inputan yaitu “jambu buji” sebagai jenis, Void jambu2::rasa() yang diberi inputan “manis” sebagai rasa, Void jambu3::warna() yang diberi inputan “kuning” sebagai warna. Dimana data yang di input nanti akan di tampilkan pada output program.

### **BAB III**

#### **KESIMPULAN**

Pada praktikum ini dapat disimpulkan bahwa dalam kelas juga mempunyai cara untuk menghemat pembuatan objek/fungsi dengan cara pewarisan (inheritance). Jadi saat membuat kelas baru tidak perlu membuat kelas dasar yang baru. Cukup dengan membuat kelas turunan dari kelas yang ada. Karena sifatnya yang khusus yaitu dapat mewariskan fungsi dan variabel yang ada pada kelas dasar dan ditambah dengan fungsi dan variabel kelas turunan itu sendiri.

Kelas Pewarisan jamak adalah apabila kelas turunan / subclass tersebut "mewarisi" dari beberapa kelas dasar / superclass sekaligus. Sedangkan pewarisan tunggal adalah apabila kelas turunan tersebut "mewarisi" hanya dari sebuah kelas dasar / kelas induk.

Kelas turunan *dapat* mengakses setiap public member kelas dasar, kelas lain juga *dapat* mengakses member kelas dasar secara langsung. Kelas turunan *tidak dapat* mengakses private member kelas dasar, kelas lain juga *tidak dapat* mengakses member kelas dasar secara langsung. Kelas turunan *dapat* mengakses setiap protected member kelas dasar, *tetapi* kelas lain *tidak dapat* mengakses member kelas dasar secara langsung.

## DAFTAR PUSTAKA

Dosen Teknik Informatika. *Modul Praktikum Algoritma Pemrograman II*.

Palangkaraya: Jurusan Teknik Informatika, 2020. Print.

Sianipar, Rismon H. *Algoritma, Struktur Data dan Pemrograman*.

<http://rhsianipar.blogspot.com/2016/12/bab-4-c-untuk-programer.html>. Diakses pada 14 April 2020

Noname. *Belajar C++ Part 11*.

<http://pemrogramanajip.blogspot.com/2011/12/belajar-c-part-11.html>. Diakses pada 14 April 2020

## LAMPIRAN

### 1. Program pertama pada C++

```
1  #include<iostream>
2  using namespace std;
3  class buah
4  {
5  protected:
6      void jenis();
7      void warna();
8      void rasa();
9
10 public:
11     char jeruk1;
12     char jeruk2;
13     char jeruk3;
14 };
15
16 class jeruk1
17 {
18 public:
19     void jenis();
20 };
21 class jeruk2 : public jeruk1
22 {
23 public:
24     void rasa();
25 };
26 class jeruk3 : public jeruk2
27 {
28 public:
29     void warna();
30 };
31 main()
32 {
33     jeruk1 jrk; jeruk2 jerk; jeruk3 jruk;
34     cout<<endl<<"Jenis Jeruk : "<<endl;
35     jrk.jenis();
36     cout<<endl<<"Rasa Jeruk 2 : "<<endl;
37     jrk.jenis(); jerk.rasa();
38     cout<<endl<<"Buah Jeruk 3 : "<<endl;
39     jrk.jenis(); jerk.rasa(); jruk.warna();
40     cout<<endl<<"Buah Jeruk dasar : "<<endl;
41     jrk.jenis(); jerk.rasa(); jruk.warna();
42 }
43 void jeruk1::jenis()
44 {
45     cout<<" Jeruk Mandarin "<<endl;
46 }
47 void jeruk2::rasa()
48 {
49     cout<<" Manis "<<endl;
50 }
51 void jeruk3::warna()
52 {
53     cout<<" Orange "<<endl;
54 }
```

Gambar 1.1

```
C:\Users\ACER\Downloads\C++ Alpro 2\buah pewarisan.exe

Buah Jeruk 1 :
Jeruk Mandarin

Rasa Jeruk 2 :
Jeruk Mandarin
Manis

Buah Jeruk 3 :
Jeruk Mandarin
Manis
Orange

Buah Jeruk dasar :
Jeruk Mandarin
Manis
Orange

-----
Process exited after 0.1933 seconds with return value 0
Press any key to continue . . .
```

Gambar 1.2



## 2. Program kedua pada c++

```
buah pewarisan.cpp  buah pewarisan2.cpp
1  #include<iostream>
2  using namespace std;
3  class buah
4  {
5  protected:
6      void jenis();
7      void warna();
8      void rasa();
9
10 public:
11     char jambu1;
12     char jambu2;
13     char jambu3;
14 };
15
16 class jambu1
17 {
18 public:
19     void jenis();
20 };
21 class jambu2 : public jambu1
22 {
23 public:
24     void rasa();
25 };
26 class jambu3 : public jambu2
27 {
28 public:
29     void warna();
30 };
31
32 int main()
33 {
34     jambu1 jmu; jambu2 jbu; jambu3 jam;
35     cout<<endl<<"Jenis Jambu : "<<endl;
36     jmu.jenis();
37     cout<<endl<<"Rasa Jambu : "<<endl;
38     jmu.jenis(); jbu.rasa();
39     cout<<endl<<"Warna Buah Jambu : "<<endl;
40     jmu.jenis(); jbu.rasa(); jam.warna();
41     cout<<endl<<"Buah Jambu : "<<endl;
42     jmu.jenis(); jbu.rasa(); jam.warna();
43 }
44 void jambu1::jenis()
45 {
46     cout<<" Jambu biji "<<endl;
47 }
48 void jambu2::rasa()
49 {
50     cout<<" Manis "<<endl;
51 }
52 void jambu3::warna()
53 {
54     cout<<" Kuning "<<endl;
```

Gambar 1.3

```
C:\Users\ACER\Downloads\C++ Alpro 2\buah pewarisan2.exe

Jenis Jambu :
Jambu biji

Rasa Jambu :
Jambu biji
Manis

Warna Buah Jambu :
Jambu biji
Manis
Kuning

Buah Jambu :
Jambu biji
Manis
Kuning

-----
Process exited after 0.0707 seconds with return value 0
Press any key to continue . . .
```

Gambar 1.4 . . . .

↗  
jenis font