

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : PRAYOGA BETRIO L**  
**NIM : 193030503060**  
**KELAS : A**  
**MODUL : II (PEWARISAN)**

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

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



Nama : PRAYOGA BETRIO L  
NIM : 193030503060  
Kelas : A  
Modul : II (PEWARISAN)

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	8
BAB II Pembahasan	60	49
BAB III Kesimpulan	20	13
Daftar Pustaka	5	5
Lampiran	5	4
Jumlah	100	79

Penilai  
Asisten Praktikum

Diana

# **BAB I**

## **TUJUAN DAN LANDASAN TEORI**

### **1.1. TUJUAN**

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

### **1.2. DASAR TEORI**

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

Selain melakukan kategorisasi terhadap objek yang memiliki sekumpulan atribut dan perilaku yang sama, manusia sering melakukan pengelompokkan terhadap objek yang memiliki kesamaan atas beberapa (**tidak semua**) atribut/perilaku. Contoh : pengelompokkan 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 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 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 nam atribut yang diakses.

Contoh :

- o Kelas C adalah kelas anak dari kelas B
- o Kelas B merupakan kelas anak dari kelas A.
- o Maka sifat sifat yang diwariskan kelas A ke kelas B juga akan diwariskan ke kelas C.

Keyword untuk pewarisan : **extends**

#### **Contoh Program:**

```
class Induk {
```

```

private String var1; //hak akses private berarti tidak dapat diakses oleh kls
anak
public int var2; //hak akses public berarti dapat diakses oleh kls anak
//konstruktor
Induk () {
}
//method
public void cetakData() {
}
}
class Anak extends Induk {
private String dataAnak; //variabel
//konstruktor
Anak () {
}
public void cetak() {
cetakData();//method milik Induk
// ada jg yang memanggil dgn super.cetakData()
}
}
class DemoPewarisan {
public static void main(String args[]) {
Anak anak = new Anak();
anak.cetak();
anak.cetakData();
System.out.println("Demo Pewarisan");
System.out.println("isi var 2="+anak.var2);
}
}

```

### Output Program:

```
C:\Documents and Settings\suherman\My Documents>javac
```

```
DemoPewarisan.java
```

```
C:\Documents and Settings\suherman\My Documents>java DemoPewarisan
```

```
Demo Pewarisan
```

```
isi var 2=0
```

### A. Keuntungan Pewarisan

#### 1. Bersifat reusable

Tidak harus menyalin semua data dan method dari suatu kelas jika akan menggunakannya lagi

#### 2. Kemudahan dalam *me-manage* kelas yang memiliki data dan method yang sama

Untuk memodifikasi suatu data atau method untuk semua subkelas / kelas anak, maka tidak perlu melakukan perubahan di masing-masing kelas anak melainkan hanya pada kelas induk saja.

Dari mekanisme pewarisan yang sudah diuraikan diatas, dapat disimpulkan bahwa pewarisan ini dikelompokkan menjadi tiga, yaitu :

- a. Pewarisan Tunggal (*single inheritance*)
- b. Pewarisan Jamak (*multiple inheritance*), dan
- c. Pewarisan Jamak Maya (*virtual multiple inheritance*)

#### a. Pewarisan Tunggal (Single Inheritance)

Pewarisan Tunggal 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.

Contoh program pewarisan tunggal :

```

#include<iostream.h>
class makhluk
{
public:
    void berkembang();
};
class hewan : public makhluk
{
public:
    void bergerak();
};
class kuda : public hewan
{
public:
    void berlari();
};
main()
{
    makhluk mk; hewan hw; kuda kd;
    cout<<endl<<" Sifat-sifat dari Makhluk adalah : "<<endl;
    mk.berkembang();
}

```

```

cout<<endl<<" Sifat-sifat dari Hewan adalah : "<<endl;
hw.berkembang(); hw.bergerak();
cout<<endl<<" Sifat-sifat dari Kuda adalah : "<<endl;
mk.berkembang(); hw.bergerak(); kd.berlari();
}
void makhluk::berkembang()
{
cout<<" Berkembang biak"<<endl;
}
void hewan::bergerak()
{
cout<<" Bergerak berpindah tempat"<<endl;
}
void kuda::berlari()
{
    cout<<" Berlari sangat kencang seperti angin"<<endl;
}

```

#### **b. Pewarisan Jamak (Multiple Inheritance)**

Pewarisan Jamak 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.

Perhatikan contoh program berikut ini:

```

#include<iostream.h>
class kuda
{
public :
void berlari()
{
cout<<" > Berlarinya sangat cepat"<<endl;
}
};
class burung
{
public:
void terbang()
{
cout<<" > Terbang menembus awan"<<endl;
}
};
class pegasus: public kuda, public burung
{

```

```

public:
void lariterbang()
{
cout<<" > Bersayap, lari dan dapat terbang ke
angkasa"<<endl;
}
};
main()
{
pegasus pg;
cout<<"Sifat dari PEGASUS yaitu : "<<endl;
pg.berlari();
pg.terbang();
pg.lariterbang();
}

```

### c. Pewarisan Jamak Maya (Virtual Multiple Inheritance)

Pewarisan Jamak Maya 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.

Contoh program dalam pewarisan ini :

```

#include<iostream.h>

class hewan
{
public:
    void bergerak()
    {
        cout<<" # Bergerak berpindah tempat"<<endl;
    }
};

class kuda: virtual public hewan
{
public :
    void berlari()
    {
        cout<<" # Berlarinya sangat cepat"<<endl;
    }
}

```

```

};
class burung: virtual public hewan
{
public:
void terbang()
{
cout<<" # Terbang menembus awan"<<endl;
}
};
class pegasus: public kuda, public burung
{
public:
void lariterbang()
{
cout<<"# Bersayap, lari dan dapat terbang ke angkasa"<<endl;
}
};
main()
{
pegasus pg;
cout<<">> Sifat dari PEGASUS << "<<endl;
cout<<"===== "<<endl;
pg.bergerak();
pg.berlari();
pg.terbang();
pg.lariterbang();
}

```

### 3. Konstruktor dan Destruktor Pada Pewarisan

Di dalam contoh-contoh diatas belum melibatkan konstruktor dan destruktur secara eksplisit. Di dalam pewarisan suatu 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.

Contoh programnya :

```
#include "iostream.h"
#include "conio.h"
#include "string.h"
class Kendaraan
{
private:
char nama[15];
public:
Kendaraan(char *nama_kendaraan = "T1AS")
{
strcpy(nama, nama_kendaraan);
cout<<" Hidupkan mesin kendaraan anda ..."<<endl;
}
~Kendaraan()
{
cout<<" Matikan mesin kendaraan anda ..."<<endl;
}
void info_kendaraan()
{
cout<<nama<<" Sedang berjalan ..."<<endl;
}
};
class Mercy : public Kendaraan
{
public:
Mercy(char *nama_mercy) : Kendaraan(nama_mercy)
```

```

{
cout<<" Hidupkan mesin mobil merah ..."<<endl;
}
~Mercy()
{
cout<<" Matikan mesin mobil merah itu ..."<<endl;
}
};
void main()
{
clrscr();
Mercy mewah(" Mobil Yang Mewah");
mewah.info_kendaraan();
cout<<" Akhir dari permulaaan()..."<<endl;
}

```

Keluarannya :

Hidupkan mesin kendaraan anda ...

Hidupkan mesin mobil merah ...

Mobil Yang Mewah Sedang berjalan ...

Akhir dari permulaan()...

Matikan mesin mobil merah itu ...

Matikan mesin kendaraan anda ...

#### 4. Pewarisan

Pemrograman C++ memungkinkan suatu class dapat mewarisi data ataupun keanggota class lain. Dalam hal ini class yang mewarisi sifat disebut class turunan, sedangkan kelas yang mewarisi disebut class dasar. Untuk menurunkan class dari yang lain, kita menggunakan operator : colon dalam deklarasi class turunan dengan cara sebagai berikut :

Class derived\_class\_name: public base\_class\_name;

Dimana `derived_class_name` adalah nama class turunan dan `base_class_name` adalah nama class yang menjadi dasar. Public dapat diganti dengan akses lain misalnya `protected` atau `private`, dan menjelaskan akses untuk member yang diturunkan.

Kelas dasar :

- Sifat a
- Sifat b

Kelas turunan :

- Sifat a
- Sifat b
- Sifat c

Sifat a dan sifat b diwariskan pada kelas turunan.

Keuntungan pewarisan sifat adalah suatu kode yang telah ditulis mudah sekali untuk dipanggil kembali. Sifat yang dimaksud dalam class adalah `private`, `protected` dan `public`. Sifat yang diwariskan adalah `protected` dan `public`. `Private` bersifat khusus sehingga tidak dapat diwariskan.

margin

## BAB II PEMBAHASAN

### 2.1 program buah yang pertama dijalankan

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

Pada program ini dengan menggunakan bahasa C++ yang bersifat yaitu case-sensitive, yang berarti bahwa huruf besar dan huruf kecil akan dianggap berbeda. Tidak seperti bahasa pemrograman Pascal, dimana penggunaan huruf besar dan kecil pada coding tidak akan mempengaruhi program. “Include” merupakan salah satu jenis pengarah preprocessor yang digunakan untuk membaca file.

Pada bagian program pertama ini digunakan dua header, yaitu `#include<iostream>` dan `#include<string>`.

Include merupakan salah satu jenis pengarah preprocessor yang digunakan untuk membaca file.

- a. `#include<iostream.h>` diperlukan pada program yang melibatkan objek `cout` dan `cin`.
- b. `#Include <iomanip.h>` Digunakan untuk menampilkan perintah `:Setiosflags()` Digunakan untuk mengatur jumlah digit decimal dibelakang koma.

Dan pada bagian `using namespace std;` adalah untuk memberitahukan kepada kompiler bahwa akan menggunakan semua fungsi , *class* atau *file* yang terdapat pada memori *namespace std*.



```

struct buah{
    char No [10];
    char NAMA [20];
    float Harga,Jumlah,Total;

};
typedef buah buahArr[50];

```

Pada bagian program diatas *Struct buah* adalah kumpulan data yang memiliki tipe data yang berbeda. Struct digunakan apabila data yang ingin dikelompokkan memiliki tipe data yang berbeda. Dengan deklarasi di atas merupakan suatu tipe data yang bernama *buah* di mana setiap data yang dideklarasikan menggunakan tipe data *char*.

```

int main(){
    int jumlahbuah,pilihan;
    buahArr X;
    while (pilihan!=7){
        cout<<"-----\n";
        cout<<"Menu Input Data Penjualan Buah-Buahan\n";
        cout<<"1. Masukan Data Buah\n";
        cout<<"2. Lihat Data Buah\n";
        cout<<"3. Tampilkan Jumlah Penjualan(Kg)\n";
        cout<<"4. Tampilkan Jumlah Total Penjualan(Rp)\n";
        cout<<"5. Tampilkan Rata-Rata Penjualan\n";
        cout<<"6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah\n";
        cout<<"7. Keluar\n";
        cout<<"-----\n";
    }
}

```

```

cin>>pilihan;

if (pilihan==1){
    cout<<"Masukan Jumlah Data Buah=";
    cin>>jumlahbuah;
    for(int i=0 ; i<jumlahbuah;i++){
        cout<<"Data Buah ke" << i+1;
        cout<<"\n No:";
        cin>>X[i].No ;
        cout<<"NAMA:";
        cin>>X[i].NAMA ;
        cout<<"Harga:";
        cin>>X[i].Harga ;
        cout<<"Jumlah Buah(Kg):";
        cin>>X[i].Jumlah ;
        X[i].Total=X[i].Harga * X[i].Jumlah ;
        cout<<"Total Bayar:"<<X[i].Total;
        cout<<endl;
        cout<<"\n";
    }
} else if (pilihan==2){
    cout<<"\n |" << setw(5)<<"No |";
    cout<<setw(10)<<"NAMA    |" <<setw(14)<<"Harga    |"
    <<setw(18)<<"Jumlah Jual |" <<setw(20)<<"Total Bayar |" <<endl;
    cout<<"-----"
    -----\n";
    for(int i=0 ; i<jumlahbuah;i++){
        cout <<" |" <<setw(3)<<X[i].No <<" |";
        cout <<setw(8)<<X[i].NAMA <<" |";
        cout <<setw(12)<<X[i].Harga <<" |" ;
        cout <<setw(16)<<X[i].Jumlah <<" |" ;
    }
}

```

```
cout <<setw(18)<<X[i].Total <<"|"<<endl ;
```

```
}
```

```
}else if (pilihan==3){
```

```
cout<<"Jumlah Penjualan(Kg)=";
```

```
int Jumlah=0;
```

```
float Jumlahp;
```

```
for(int i=0 ; i<jumlahbuah;i++){
```

```
Jumlah+=X[i].Jumlah ;
```

```
}
```

```
Jumlahp=Jumlah;
```

```
cout<<Jumlahp<<"\n";
```

```
}else if (pilihan==4){
```

```
cout<<"Jumlah Total Penjualan(Rp)=";
```

```
int Total=0;
```

```
float Totalp;
```

```
for (int i=0; i<jumlahbuah ; i++){
```

```
X[i].Total=X[i].Harga * X[i].Jumlah ;
```

```
Total+=X[i].Total ;
```

```
}
```

```
Totalp=Total ;
```

```
cout<<Totalp<<"\n";
```

```
}else if (pilihan==5){
```

```
cout<<"Jumlah Rata-Rata Penjualan(Rp)=";
```

```
int Total=0;
```

```
float rata;
```

```
for (int i=0; i<jumlahbuah ; i++){
```

```

        Total+=X[i].Total ;
    }
    rata=Total/jumlahbuah;
    cout<<rata<<"\n";

} else if (pilihan==6){
    int Max=X[0].Total ,Min=X[0].Total ;
    int indexMax=0,indexMin=0;
    for(int i=0 ; i<jumlahbuah;i++){
        if (Max<X[i].Total ){
            Max=X[i].Total ;
            indexMax=i;
        }
        if (Min>X[i].Total ){
            Min=X[i].Total ;
            indexMin=i;
        }
    }
    cout<<"Data Total Penjualan Buah Tertinggi\n";
    cout <<" |"<<setw(3)<<X[indexMax].No <<" |";
    cout <<setw(8)<<X[indexMax].NAMA <<" |";
    cout <<setw(12)<<X[indexMax].Harga <<" |" ;
    cout <<setw(16)<<X[indexMax].Jumlah <<" |" ;
    cout <<setw(18)<<X[indexMax].Total <<" |" <<endl;
    cout<<"-----\n";

}

cout<<"Data Total Penjualan Buah Terendah\n";
cout <<" |"<<setw(3)<<X[indexMin].No <<" |";
cout <<setw(8)<<X[indexMin].NAMA <<" |";
cout <<setw(12)<<X[indexMin].Harga <<" |" ;

```

```
cout <<setw(16)<<X[indexMin].Jumlah <<" |" ;  
cout <<setw(18)<<X[indexMin].Total <<" |" <<endl;  
  
}  
}  
}
```

Pada bagian program ini merupakan bagian utama program. *int main()* merupakan menandakan awalan dari bagian utama program dan menandakan berakhirnya program.

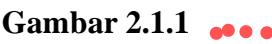
Tanda ‘**}**’ menandakan berakhirnya program.

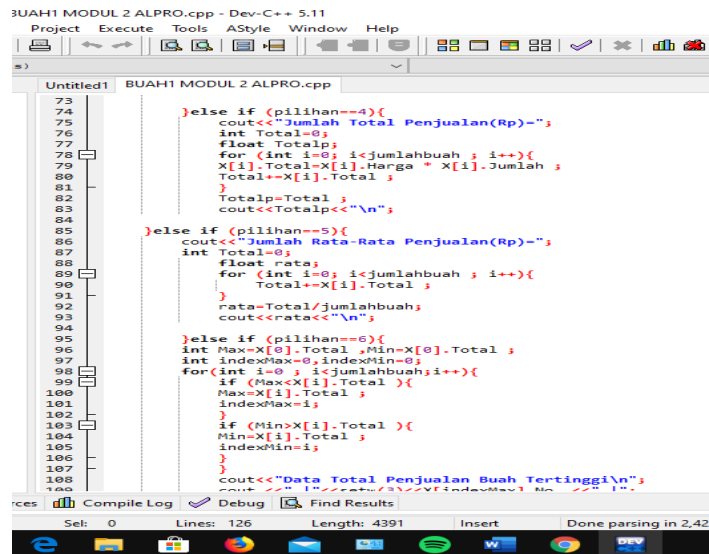
Pada program ini juga digunakan **cout** yang berfungsi untuk menampilkan data ke dalam output (cetak pada layar). Tanda ‘<<’ merupakan operator yang fungsinya sebagai penghubung antara stream dengan kalimat inputan. **endl** merupakan manipulator yang digunakan untuk menyisipkan karakter mengatur pindah baris pada program. Tanda ‘;’ artinya menyatakan bahwa akhir dari sebuah pernyataan. Tanda ‘{’ dan ‘}’ berfungsi untuk menandakan awal dan akhir sebuah definisi fungsi dan program.

**Setw** merupakan satu fungsi manipulator yang digunakan untuk mengatur lebar tampilan dilayar dari suatu nilai variable. File header yang harus disertakan adalah file header *iomanip.h*.

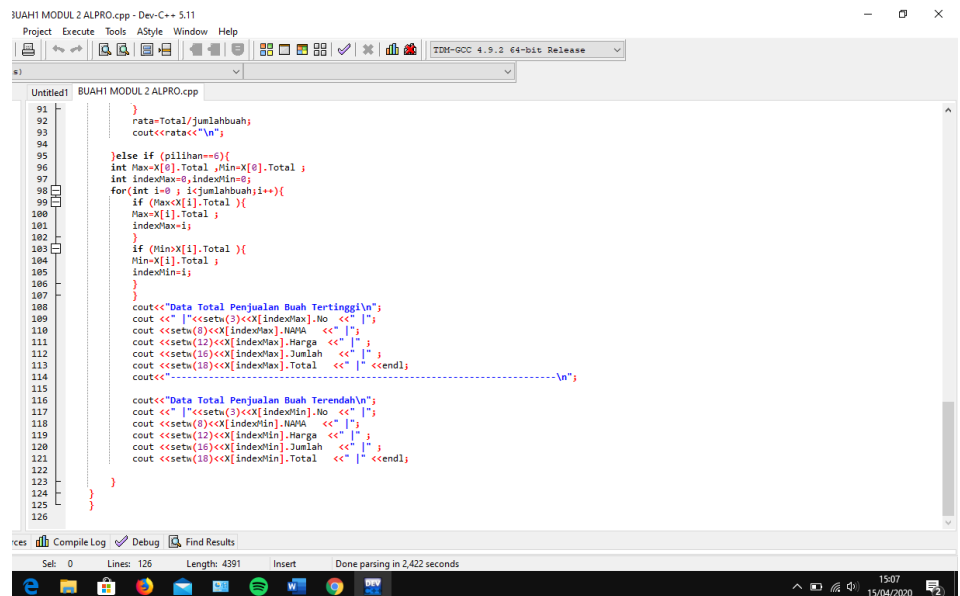
Program dijalankan kan

**Input :**



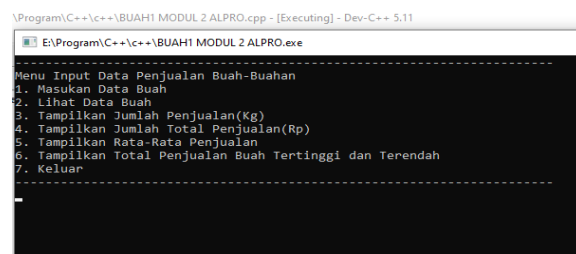


Gambar 2.1.3



Gambar 2.1.4

Output :



Gambar 2.1.5

```
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
1
Masukan Jumlah Data Buah=2
Data Buah ke1
No:1
NAMA:pisang
Harga:5000
Jumlah Buah(Kg):1
Total Bayar:5000
Data Buah ke2
No:1
NAMA:jambu
Harga:4000
Jumlah Buah(Kg):1
Total Bayar:4000
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
```

Gambar 2.1.6

```
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
Total Bayar:4000
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
2
| No | NAMA | Harga | Jumlah Jual | Total Bayar |
| 1 | pisang | 5000 | 1 | 5000 |
| 1 | jambu | 4000 | 1 | 4000 |
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
2
```

Gambar 2.1.7

```
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
2
-----
| No | NAMA | Harga | Jumlah Jual | Total Bayar |
| 1 | pisang | 5000 | 1 | 5000 |
| 1 | jambu | 4000 | 1 | 4000 |
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
3
Jumlah Penjualan(Kg)=2
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
```

Gambar 2.1.8



```

A:\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
Jumlah Penjualan(Kg)=2
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
Jumlah Total Penjualan(Rp)=9000
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----

```

Gambar 2.1.9

```

A:\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
Jumlah Total Penjualan(Rp)=9000
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
Jumlah Rata-Rata Penjualan(Rp)=4500
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----

```

Gambar 2.1.10

```

\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
5
Jumlah Rata-Rata Penjualan(Rp)=4500
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
6
Data Total Penjualan Buah Tertinggi
| 1 | pisang | 5000 | 1 | 5000 |
-----
Data Total Penjualan Buah Terendah
| 1 | jambu | 4000 | 1 | 4000 |
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----

```

Gambar 2.1.11

```
(Program)C++>c++\BUAH1 MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++>c++\BUAH1 MODUL 2 ALPRO.exe
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
6
Data Total Penjualan Buah Tertinggi
| 1 | pisang | 5000 |
-----
Data Total Penjualan Buah Terendah
| 1 | jambu | 4000 |
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
7
-----
Process exited after 162.7 seconds with return value 0
Press any key to continue . . .
```

**Gambar 2.1.12**

## 2.2 program buah kedua yang dijalankan

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

Pada program ini dengan menggunakan bahasa C++ yang bersifat yaitu case-sensitive, yang berarti bahwa huruf besar dan huruf kecil akan dianggap berbeda. Tidak seperti bahasa pemrograman Pascal, dimana penggunaan huruf besar dan kecil pada coding tidak akan mempengaruhi program. “Include” merupakan salah satu jenis pengarah preprocessor yang digunakan untuk membaca file.

Pada bagian program kedua ini digunakan satu header, yaitu `#include<iostream>`.

Include merupakan salah satu jenis pengarah preprocessor yang digunakan untuk membaca file.

- a. `#include<iostream.h>` diperlukan pada program yang melibatkan objek `cout` dan `cin`.

Dan pada bagian `using namespace std;` adalah untuk memberitahukan kepada kompiler bahwa akan menggunakan semua fungsi, *class* atau *file* yang terdapat pada memori *namespace std*.

```
int main (){
```

```

int kode,harga,jumlah,total,bayar,kembali;
char mad;
do
{
cout<<"===== "<<endl;
cout<<"TOKO BUAH PRAYOGA BETRIO L"<<endl;
cout<<"===== "<<endl;
cout<<" "<<endl;
cout<<"DAFTAR HARGA BUAH"<<endl;
cout<<"1. MANGGA Rp. 16.000"<<endl;
cout<<"2. SEMANGKA Rp. 12.000"<<endl;
cout<<"3. DURIAN Rp. 50.000"<<endl;
cout<<"4. APEL Rp. 8.000"<<endl;
cout<<"===== "<<endl;
cout<<"\n"<<"Masukan No Menu Pilihan : ";
cin>>kode;
switch (kode){
case 1:
cout<<"\n"<<"YANG DI BELI MANGGA"<<endl;
harga=16000;
cout<<"Masukan Jumlah Pesanan : ";
cin>>jumlah;
total=harga*jumlah;
cout<<"Total Harga : Rp. "<<total<<endl;
cout<<"Uang yang Dibayar : Rp. ";
cin>>bayar;
kembali=bayar-total;
cout<<"Kembali : Rp. "<<kembali<<endl;
cout<<"Masih ada Y/T : ";
cin>>mad;
break;

```

case 2:

```
cout<<"\n'<<"YANG DI BELI SEMANGKA"<<endl;
harga=12000;
cout<<"Masukan Jumlah Pesanan : ";
cin>>jumlah;
total=harga*jumlah;
cout<<"Total Harga : Rp. "<<total<<endl;
cout<<"Uang yang Dibayar : Rp. ";
cin>>bayar;
kembali=bayar-total;
cout<<"Kembali : Rp. "<<kembali<<endl;
cout<<"Masih ada Y/T : ";
cin>>mad;
break;
```

case 3:

```
cout<<"\n'<<"YANG DI BELI DURIAN"<<endl;
harga=50000;
cout<<"Masukan Jumlah Pesanan : ";
cin>>jumlah;
total=harga*jumlah;
cout<<"Total Harga : Rp. "<<total<<endl;
cout<<"Uang yang Dibayar : Rp. ";
cin>>bayar;
kembali=bayar-total;
cout<<"Kembali : Rp. "<<kembali<<endl;
cout<<"Masih ada Y/T : ";
cin>>mad;
break;
```

case 4:

```
cout<<"\n'<<"YANG DI BELI APEL"<<endl;
harga=8000;
```

```

cout<<"Masukan Jumlah Pesanan : ";
cin>>jumlah;
total=harga*jumlah;
cout<<"Total Harga : Rp. "<<total<<endl;
cout<<"Uang yang Dibayar : Rp. ";
cin>>bayar;
kembali=bayar-total;
cout<<"Kembali : Rp. "<<kembali<<endl;
cout<<"Masih ada Y/T : ";
cin>>mad;
break;
default:
cout<<"Kode yang anda masukkan tidak ada";
}
}
while (mad!='Y');
cout<<"Terimah Kasih SUDAH KE TOKO PRAYOGA
BETRIO L"<<endl;
return 0;
}

```

Pada bagian program ini merupakan bagian utama program. *int main()* merupakan menandakan awalan dari bagian utama program dan menandakan berakhirnya program.

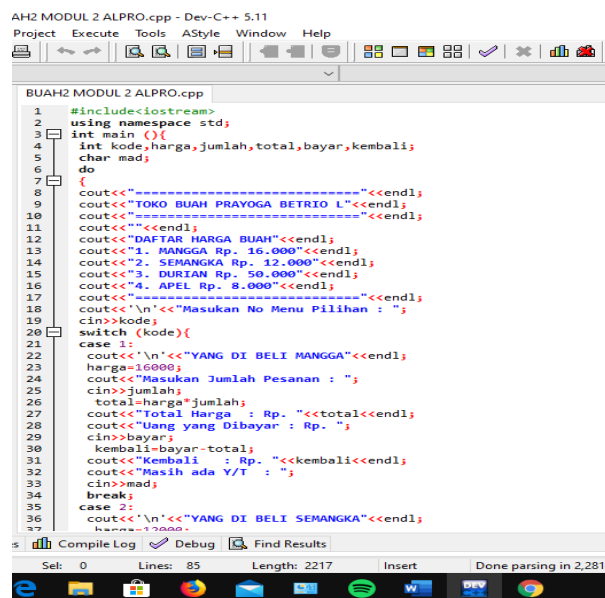
**cout** yang berfungsi untuk menampilkan data ke dalam output (cetak pada layar). Tanda '<<' merupakan operator yang fungsinya sebagai penghubung antara stream dengan kalimat inputan. Jadi, jika kita ingin menampilkan kalimat ke layar, kita perlu menghubungkan kalimat yang ingin ditampilkan dengan cout. Untuk itulah kita menggunakan operator << ini. Saat operator ini digunakan, maka kalimat akan dialirkan ke cout dan cout akan

mencetaknya ke layar. Sedangkan **endl** merupakan manipulator yang digunakan untuk menyisipkan karakter mengatur pindah baris pada program. Tanda ';' artinya menyatakan bahwa akhir dari sebuah pernyataan.

**Return** digunakan karena pada main program menggunakan tipe data integer. Tanda '}' menandakan berakhirnya program.

Program yang dijalankan :

**Input :**



```
1 #include<iostream>
2 using namespace std;
3 int main () {
4     int kode, harga, jumlah, total, bayar, kembali;
5     char mad;
6     do
7     {
8         cout<<"===== "<<endl;
9         cout<<"TOKO BUAH PRAYOGA BETRIO L."<<endl;
10        cout<<" "<<endl;
11        cout<<"DAFTAR HARGA BUAH"<<endl;
12        cout<<"1. MANGGA Rp. 16.000"<<endl;
13        cout<<"2. SEMANGKA Rp. 12.000"<<endl;
14        cout<<"3. DURIAN Rp. 50.000"<<endl;
15        cout<<"4. APEL Rp. 8.000"<<endl;
16        cout<<" "<<endl;
17        cout<<"\n"<<"Masukan No Menu Pilihan : ";
18        cin>>kode;
19        switch (kode){
20        case 1:
21            cout<<"\n"<<"YANG DI BELI MANGGA"<<endl;
22            harga=16000;
23            cout<<"Masukan Jumlah Pesanan : ";
24            cin>>jumlah;
25            total=harga*jumlah;
26            cout<<"Total Harga : Rp. "<<total<<endl;
27            cout<<"Uang yang Dibayar : Rp. ";
28            cin>>bayar;
29            kembali=total-bayar;
30            cout<<"Kembali : Rp. "<<kembali<<endl;
31            cout<<"Masih ada Y/T : ";
32            cin>>mad;
33            break;
34        case 2:
35            cout<<"\n"<<"YANG DI BELI SEMANGKA"<<endl;
36            harga=12000;
37            break;
38        }
```

**Gambar 2.2.1**

```

36      cout<<"\n"<<"YANG DI BELI SEMANGKA"<<endl;
37      harga=12000;
38      cout<<"Masukan Jumlah Pesanan : ";
39      cin>>jumlah;
40      total=harga*jumlah;
41      cout<<"Total Harga : Rp. "<<total<<endl;
42      cout<<"Uang yang Dibayar : Rp. ";
43      cin>>bayar;
44      kembali=bayar-total;
45      cout<<"Kembali : Rp. "<<kembali<<endl;
46      cout<<"Masih ada Y/T : ";
47      cin>>mad;
48      break;
49      case 3:
50      cout<<"\n"<<"YANG DI BELI DURIAN"<<endl;
51      harga=50000;
52      cout<<"Masukan Jumlah Pesanan : ";
53      cin>>jumlah;
54      total=harga*jumlah;
55      cout<<"Total Harga : Rp. "<<total<<endl;
56      cout<<"Uang yang Dibayar : Rp. ";
57      cin>>bayar;
58      kembali=bayar-total;
59      cout<<"Kembali : Rp. "<<kembali<<endl;
60      cout<<"Masih ada Y/T : ";
61      cin>>mad;
62      break;
63      case 4:
64      cout<<"\n"<<"YANG DI BELI APEL"<<endl;
65      harga=8000;
66      cout<<"Masukan Jumlah Pesanan : ";
67      cin>>jumlah;
68      total=harga*jumlah;
69      cout<<"Total Harga : Rp. "<<total<<endl;
70      cout<<"Uang yang Dibayar : Rp. ";
71      cin>>bayar;
72      kembali=bayar-total;
73      cout<<"Kembali : Rp. "<<kembali<<endl;
74      cout<<"Masih ada Y/T : ";
75      cin>>mad;
76      break;
77      default:
78      cout<<"Kode yang anda masukkan tidak ada";
79      }
80      }
81      while (mad!="Y");
82      cout<<"Terimah Kasih SUDAH KE TOKO PRAYOGA BETRIO L"<<endl;
83      return 0;
84      }
85

```

Gambar 2.2.2

```

50      cout<<"\n"<<"YANG DI BELI DURIAN"<<endl;
51      harga=50000;
52      cout<<"Masukan Jumlah Pesanan : ";
53      cin>>jumlah;
54      total=harga*jumlah;
55      cout<<"Total Harga : Rp. "<<total<<endl;
56      cout<<"Uang yang Dibayar : Rp. ";
57      cin>>bayar;
58      kembali=bayar-total;
59      cout<<"Kembali : Rp. "<<kembali<<endl;
60      cout<<"Masih ada Y/T : ";
61      cin>>mad;
62      break;
63      case 4:
64      cout<<"\n"<<"YANG DI BELI APEL"<<endl;
65      harga=8000;
66      cout<<"Masukan Jumlah Pesanan : ";
67      cin>>jumlah;
68      total=harga*jumlah;
69      cout<<"Total Harga : Rp. "<<total<<endl;
70      cout<<"Uang yang Dibayar : Rp. ";
71      cin>>bayar;
72      kembali=bayar-total;
73      cout<<"Kembali : Rp. "<<kembali<<endl;
74      cout<<"Masih ada Y/T : ";
75      cin>>mad;
76      break;
77      default:
78      cout<<"Kode yang anda masukkan tidak ada";
79      }
80      }
81      while (mad!="Y");
82      cout<<"Terimah Kasih SUDAH KE TOKO PRAYOGA BETRIO L"<<endl;
83      return 0;
84      }
85

```

Gambar 2.2.3

Output :

```
E:\Program\C++\c++\BUAH2 MODUL 2 ALPRO.exe
=====
TOKO BUAH PRAYOGA BETRIO L
=====

DAFTAR HARGA BUAH
1. MANGGA Rp. 16.000
2. SEMANGKA Rp. 12.000
3. DURIAN Rp. 50.000
4. APEL Rp. 8.000
=====

Masukan No Menu Pilihan :
```

Gambar 2.2.4

```
E:\Program\C++\c++\BUAH2 MODUL 2 ALP
=====
TOKO BUAH PRAYOGA BETRIO L
=====

DAFTAR HARGA BUAH
1. MANGGA Rp. 16.000
2. SEMANGKA Rp. 12.000
3. DURIAN Rp. 50.000
4. APEL Rp. 8.000
=====

Masukan No Menu Pilihan : 1

YANG DI BELI MANGGA
Masukan Jumlah Pesanan : 2
Total Harga : Rp. 32000
Uang yang Dibayar : Rp. 50000
Kembali : Rp. 18000
Masih ada Y/T : Y

=====
TOKO BUAH PRAYOGA BETRIO L
=====

DAFTAR HARGA BUAH
1. MANGGA Rp. 16.000
2. SEMANGKA Rp. 12.000
3. DURIAN Rp. 50.000
4. APEL Rp. 8.000
=====
```

Gambar 2.2.5

```
E:\Program\C++\c++\BUAH2 MODUL 2 ALPRO.exe
YANG DI BELI MANGGA
Masukan Jumlah Pesanan : 2
Total Harga : Rp. 32000
Uang yang Dibayar : Rp. 50000
Kembali : Rp. 18000
Masih ada Y/T : Y
=====
TOKO BUAH PRAYOGA BETRIO L
=====

DAFTAR HARGA BUAH
1. MANGGA Rp. 16.000
2. SEMANGKA Rp. 12.000
3. DURIAN Rp. 50.000
4. APEL Rp. 8.000
=====

Masukan No Menu Pilihan : 2

YANG DI BELI SEMANGKA
Masukan Jumlah Pesanan : 1
Total Harga : Rp. 12000
Uang yang Dibayar : Rp. 15000
Kembali : Rp. 3000
Masih ada Y/T : T
Terimah Kasih SUDAH KE TOKO PRAYOGA BETRIO L

=====
Process exited after 77.01 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.2.6



### **BAB III**

### **KESIMPULAN**

~~Dari modul ini dapat ditarik kesimpulan sebagai berikut:~~

Pewarisan (*inheritance*) adalah penurunan sifat yang ada pada suatu kelas kepada kelas baru yang menjadi turunannya. 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. Dari mekanisme pewarisan yang sudah diuraikan diatas, dapat disimpulkan bahwa pewarisan ini dikelompokkan menjadi tiga, yaitu :

1. Pewarisan Tunggal (*single inheritance*)
2. Pewarisan Jamak (*multiple inheritance*), dan
3. Pewarisan Jamak Maya (*virtual multiple inheritance*)

## DAFTAR PUSTAKA

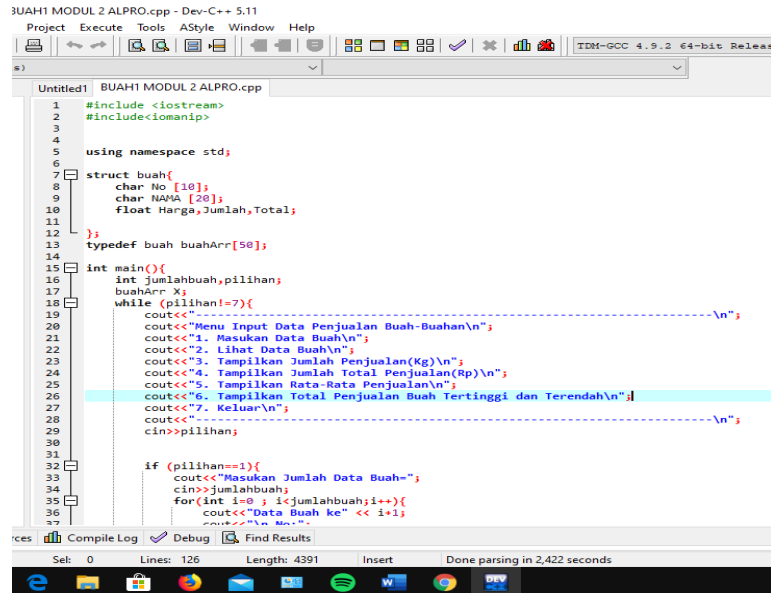
Tim Dosen Algoritma Pemrograman II. 2017. *Modul Praktikum Algoritma Pemrograman II*. Universitas Palangka Raya : UPR. Fakultas Teknik.

Movita09. 2013. *Pewarisan*. <http://blogspot.com/2013/12/pewarisan.html>  
Diakses 15 April 2020 jam 18.20 WIB.

Windasf Alinha. 2014. *Diagram Kelas Class Diagram*.  
<http://blogspot.com/2014/12/diagram-kelas-class-diagram.html>. Diakses 15 April 2020 jam 20.30 WIB.

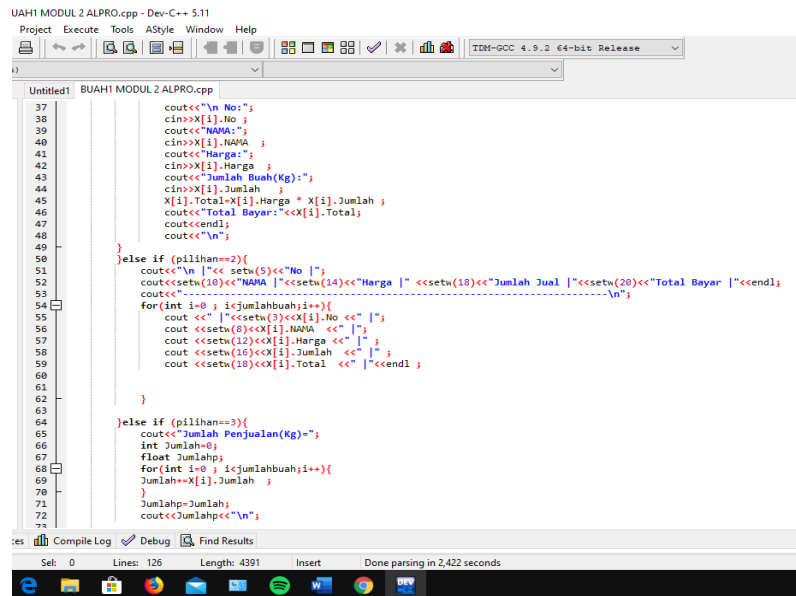
## LAMPIRAN

~~Gambar Program buah pertama: input dan output :~~  
~~Input :~~



```
1 #include <iostream>
2 #include <iomanip>
3
4
5 using namespace std;
6
7 struct buah{
8     char No [10];
9     char NAMA [20];
10    float Harga, Jumlah, Total;
11 }
12
13 typedef buah buahArr[50];
14
15 int main(){
16     int jumlahbuah, pilihan;
17     buahArr X;
18     while (pilihan!=7){
19         cout<<"-----\n";
20         cout<<"Menu Input Data Penjualan Buah-Buahan\n";
21         cout<<"1. Masukan Data Buah\n";
22         cout<<"2. Lihat Data Buah\n";
23         cout<<"3. Tampilkan Jumlah Penjualan(Kg)\n";
24         cout<<"4. Tampilkan Jumlah Total Penjualan(Rp)\n";
25         cout<<"5. Tampilkan Rata-Rata Penjualan\n";
26         cout<<"6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah\n";
27         cout<<"7. Keluar\n";
28         cout<<"-----\n";
29         cin>>pilihan;
30
31         if (pilihan==1){
32             cout<<"Masukan Jumlah Data Buah=";
33             cin>>jumlahbuah;
34             for(int i=0 ; i<jumlahbuah;i++){
35                 cout<<"Data Buah ke" << i+1;
36                 cout<<" ";
37             }
38         }
39     }
40 }
```

Gambar 2.1.1



```
37     cout<<"\n No:";
38     cin>>X[i].No ;
39     cout<<"NAMA:";
40     cin>>X[i].NAMA ;
41     cout<<"Harga:";
42     cin>>X[i].Harga ;
43     cout<<"Jumlah Buah(Kg):";
44     cin>>X[i].Jumlah ;
45     X[i].Total=X[i].Harga * X[i].Jumlah ;
46     cout<<"Total Bayar: <<X[i].Total;
47     cout<<"\n";
48 }
49
50 }else if (pilihan==2){
51     cout<<"\n |"<< setw(5)<<"No |";
52     cout<<setw(10)<<"NAMA |"<<setw(14)<<"Harga |" <<setw(10)<<"Jumlah Jual |"<<setw(20)<<"Total Bayar |"<<endl;
53     cout<<"-----\n";
54     for(int i=0 ; i<jumlahbuah;i++){
55         cout <<" |"<<setw(3)<<X[i].No <<" |";
56         cout <<setw(8)<<X[i].NAMA <<" |";
57         cout <<setw(12)<<X[i].Harga <<" |";
58         cout <<setw(16)<<X[i].Jumlah <<" |";
59         cout <<setw(18)<<X[i].Total <<" |"<<endl ;
60     }
61 }
62
63 }else if (pilihan==3){
64     cout<<"Jumlah Penjualan(Kg)=";
65     int Jumlah=0;
66     float Jumlahp;
67     for(int i=0 ; i<jumlahbuah;i++){
68         Jumlah+=X[i].Jumlah ;
69     }
70     Jumlahp=Jumlah;
71     cout<<Jumlahp<<"\n";
72 }
73 }
```

Gambar 2.1.2

```

73
74 }else if (pilihan==4){
75     cout<<"Jumlah Total Penjualan(Rp)-";
76     int Total=0;
77     float Totalp;
78     for (int i=0; i<jumlahbuah ; i++){
79         X[i].Total=X[i].Harga * X[i].Jumlah ;
80         Total+=X[i].Total ;
81     }
82     Totalp=Total ;
83     cout<<Totalp<<"\n";
84
85 }else if (pilihan==5){
86     cout<<"Jumlah Rata-Rata Penjualan(Rp)-";
87     int Total=0;
88     float rata;
89     for (int i=0; i<jumlahbuah ; i++){
90         Total+=X[i].Total ;
91     }
92     rata=Total/jumlahbuah;
93     cout<<rata<<"\n";
94
95 }else if (pilihan==6){
96     int Max=X[0].Total ,Min=X[0].Total ;
97     int indexMax=0,indexMin=0;
98     for(int i=0 ; i<jumlahbuah;i++){
99         if (Max<X[i].Total ){
100             Max=X[i].Total ;
101             indexMax=i;
102         }
103         if (Min>X[i].Total ){
104             Min=X[i].Total ;
105             indexMin=i;
106         }
107     }
108     cout<<"Data Total Penjualan Buah Tertinggi\n";
109     cout<<Max<<"\n";
110     cout<<"Data Total Penjualan Buah Terendah\n";
111     cout<<Min<<"\n";
112 }
113
114
115
116
117
118
119
120
121
122
123
124
125
126

```

Gambar 2.1.3

```

91
92     rata=Total/jumlahbuah;
93     cout<<rata<<"\n";
94
95 }else if (pilihan==6){
96     int Max=X[0].Total ,Min=X[0].Total ;
97     int indexMax=0,indexMin=0;
98     for(int i=0 ; i<jumlahbuah;i++){
99         if (Max<X[i].Total ){
100             Max=X[i].Total ;
101             indexMax=i;
102         }
103         if (Min>X[i].Total ){
104             Min=X[i].Total ;
105             indexMin=i;
106         }
107     }
108     cout<<"Data Total Penjualan Buah Tertinggi\n";
109     cout << " | " <<setw(3)<<X[indexMax].No << " | " ;
110     cout <<setw(8)<<X[indexMax].NAMA << " | " ;
111     cout <<setw(12)<<X[indexMax].Harga << " | " ;
112     cout <<setw(16)<<X[indexMax].Jumlah << " | " ;
113     cout <<setw(18)<<X[indexMax].Total << " | " <<endl;
114     cout<<"-----\n";
115
116     cout<<"Data Total Penjualan Buah Terendah\n";
117     cout << " | " <<setw(3)<<X[indexMin].No << " | " ;
118     cout <<setw(8)<<X[indexMin].NAMA << " | " ;
119     cout <<setw(12)<<X[indexMin].Harga << " | " ;
120     cout <<setw(16)<<X[indexMin].Jumlah << " | " ;
121     cout <<setw(18)<<X[indexMin].Total << " | " <<endl;
122
123 }
124
125
126

```

Gambar 2.1.4



```

\\Program\\C++\\c++\\BUAH1 MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\\Program\\C++\\c++\\BUAH1 MODUL 2 ALPRO.exe
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----

```

Gambar 2.1.5

```
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
1
Masukan Jumlah Data Buah=2
Data Buah ke1
No:1
NAMA:pisang
Harga:5000
Jumlah Buah(Kg):1
Total Bayar:5000
Data Buah ke2
No:1
NAMA:jambu
Harga:4000
Jumlah Buah(Kg):1
Total Bayar:4000
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
```

Gambar 2.1.6

```
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
Total Bayar:4000
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
2
| No | NAMA | Harga | Jumlah Jual | Total Bayar |
| 1 | pisang | 5000 | 1 | 5000 |
| 1 | jambu | 4000 | 1 | 4000 |
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
2
```

Gambar 2.1.7

```
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
2
-----
| No | NAMA | Harga | Jumlah Jual | Total Bayar |
| 1 | pisang | 5000 | 1 | 5000 |
| 1 | jambu | 4000 | 1 | 4000 |
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
3
Jumlah Penjualan(Kg)=2
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
```

Gambar 2.1.8

```

A:\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
Jumlah Penjualan(Kg)=2
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
Jumlah Total Penjualan(Rp)=9000
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----

```

Gambar 2.1.9

```

A:\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
Jumlah Total Penjualan(Rp)=9000
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
Jumlah Rata-Rata Penjualan(Rp)=4500
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----

```

Gambar 2.1.10

```

\Program\C++\c++\BUAHI MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAHI MODUL 2 ALPRO.exe
5
-----
Jumlah Rata-Rata Penjualan(Rp)=4500
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
6
Data Total Penjualan Buah Tertinggi
| 1 | pisang | 5000 | 1 | 5000 |
-----
Data Total Penjualan Buah Terendah
| 1 | jambu | 4000 | 1 | 4000 |
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukkan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----

```

Gambar 2.1.11

```

i:\Program\C++\c++\BUAH1 MODUL 2 ALPRO.cpp - [Executing] - Dev-C++ 5.11
E:\Program\C++\c++\BUAH1 MODUL 2 ALPRO.exe
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
6
Data Total Penjualan Buah Tertinggi
| 1 | pisang | 5000 |
-----
Data Total Penjualan Buah Terendah
| 1 | jambu | 4000 |
-----
Menu Input Data Penjualan Buah-Buahan
1. Masukan Data Buah
2. Lihat Data Buah
3. Tampilkan Jumlah Penjualan(Kg)
4. Tampilkan Jumlah Total Penjualan(Rp)
5. Tampilkan Rata-Rata Penjualan
6. Tampilkan Total Penjualan Buah Tertinggi dan Terendah
7. Keluar
-----
7
-----
Process exited after 162.7 seconds with return value 0
Press any key to continue . . .

```

Gambar 2.1.12

Gambar Program buah kedua input dan output :

Input :

```

AH2 MODUL 2 ALPRO.cpp - Dev-C++ 5.11
Project Execute Tools AStyle Window Help
BUAH2 MODUL 2 ALPRO.cpp
1 #include<iostream>
2 using namespace std;
3 int main () {
4     int kode,harga,jumlah,total,bayar,kembali;
5     char mad;
6     do
7     {
8         cout<<"=====<<endl;
9         cout<<"TOKO BUAH PRAYOGA BETRIO L"<<endl;
10        cout<<"=====<<endl;
11        cout<<"<<endl;
12        cout<<"DAFTAR HARGA BUAH"<<endl;
13        cout<<"1. MANGGA Rp. 16.000"<<endl;
14        cout<<"2. SEMANGKA Rp. 12.000"<<endl;
15        cout<<"3. DURIAN Rp. 50.000"<<endl;
16        cout<<"4. APEL Rp. 8.000"<<endl;
17        cout<<"=====<<endl;
18        cout<<"\n"<<"Masukan No Menu Pilihan : ";
19        cin>>kode;
20        switch (kode){
21            case 1:
22                cout<<"\n"<<"YANG DI BELI MANGGA"<<endl;
23                harga=16000;
24                cout<<"Masukan Jumlah Pesanan : ";
25                cin>>jumlah;
26                total=harga*jumlah;
27                cout<<"Total Harga : Rp. "<<total<<endl;
28                cout<<"Uang yang Dibayar : Rp. ";
29                cin>>bayar;
30                kembali=bayar-total;
31                cout<<"Kembali : Rp. "<<kembali<<endl;
32                cout<<"Masih ada Y/T : ";
33                cin>>mad;
34                break;
35            case 2:
36                cout<<"\n"<<"YANG DI BELI SEMANGKA"<<endl;
37
38        }
39    } while (mad != 'Y' && mad != 'T');
40    return 0;
41 }

```

Gambar 2.2.1

```

36      cout<<"\n"<<"YANG DI BELI SEMANGKA"<<endl;
37      harga=12000;
38      cout<<"Masukan Jumlah Pesanan : ";
39      cin>>jumlah;
40      total=harga*jumlah;
41      cout<<"Total Harga : Rp. "<<total<<endl;
42      cout<<"Uang yang Dibayar : Rp. ";
43      cin>>bayar;
44      kembali=bayar-total;
45      cout<<"Kembali : Rp. "<<kembali<<endl;
46      cout<<"Masih ada Y/T : ";
47      cin>>mad;
48      break;
49      case 3:
50      cout<<"\n"<<"YANG DI BELI DURIAN"<<endl;
51      harga=50000;
52      cout<<"Masukan Jumlah Pesanan : ";
53      cin>>jumlah;
54      total=harga*jumlah;
55      cout<<"Total Harga : Rp. "<<total<<endl;
56      cout<<"Uang yang Dibayar : Rp. ";
57      cin>>bayar;
58      kembali=bayar-total;
59      cout<<"Kembali : Rp. "<<kembali<<endl;
60      cout<<"Masih ada Y/T : ";
61      cin>>mad;
62      break;
63      case 4:
64      cout<<"\n"<<"YANG DI BELI APEL"<<endl;
65      harga=8000;
66      cout<<"Masukan Jumlah Pesanan : ";
67      cin>>jumlah;
68      total=harga*jumlah;
69      cout<<"Total Harga : Rp. "<<total<<endl;
70      cout<<"Uang yang Dibayar : Rp. ";
71      cin>>bayar;
72      kembali=bayar-total;
73      cout<<"Kembali : Rp. "<<kembali<<endl;
74      cout<<"Masih ada Y/T : ";
75      cin>>mad;
76      break;
77      default:
78      cout<<"Kode yang anda masukkan tidak ada";
79      }
80      }
81      while (mad!="Y");
82      cout<<"Terimah Kasih SUDAH KE TOKO PRAYOGA BETRIO L"<<endl;
83      return 0;
84      }
85

```

Gambar 2.2.2

```

50      cout<<"\n"<<"YANG DI BELI DURIAN"<<endl;
51      harga=50000;
52      cout<<"Masukan Jumlah Pesanan : ";
53      cin>>jumlah;
54      total=harga*jumlah;
55      cout<<"Total Harga : Rp. "<<total<<endl;
56      cout<<"Uang yang Dibayar : Rp. ";
57      cin>>bayar;
58      kembali=bayar-total;
59      cout<<"Kembali : Rp. "<<kembali<<endl;
60      cout<<"Masih ada Y/T : ";
61      cin>>mad;
62      break;
63      case 4:
64      cout<<"\n"<<"YANG DI BELI APEL"<<endl;
65      harga=8000;
66      cout<<"Masukan Jumlah Pesanan : ";
67      cin>>jumlah;
68      total=harga*jumlah;
69      cout<<"Total Harga : Rp. "<<total<<endl;
70      cout<<"Uang yang Dibayar : Rp. ";
71      cin>>bayar;
72      kembali=bayar-total;
73      cout<<"Kembali : Rp. "<<kembali<<endl;
74      cout<<"Masih ada Y/T : ";
75      cin>>mad;
76      break;
77      default:
78      cout<<"Kode yang anda masukkan tidak ada";
79      }
80      }
81      while (mad!="Y");
82      cout<<"Terimah Kasih SUDAH KE TOKO PRAYOGA BETRIO L"<<endl;
83      return 0;
84      }
85

```

Gambar 2.2.3

Output :



```
E:\Program\C++\c++\BUAH2 MODUL 2 ALPRO.exe
=====
TOKO BUAH PRAYOGA BETRIO L
=====
DAFTAR HARGA BUAH
1. MANGGA Rp. 16.000
2. SEMANGKA Rp. 12.000
3. DURIAN Rp. 50.000
4. APEL Rp. 8.000
=====
Masukan No Menu Pilihan :
```

Gambar 2.2.4

```
E:\Program\C++\c++\BUAH2 MODUL 2 ALPF
=====
TOKO BUAH PRAYOGA BETRIO L
=====
DAFTAR HARGA BUAH
1. MANGGA Rp. 16.000
2. SEMANGKA Rp. 12.000
3. DURIAN Rp. 50.000
4. APEL Rp. 8.000
=====
Masukan No Menu Pilihan : 1
YANG DI BELI MANGGA
Masukan Jumlah Pesanan : 2
Total Harga : Rp. 32000
Uang yang Dibayar : Rp. 50000
Kembali : Rp. 18000
Masih ada Y/T : Y
=====
TOKO BUAH PRAYOGA BETRIO L
=====
DAFTAR HARGA BUAH
1. MANGGA Rp. 16.000
2. SEMANGKA Rp. 12.000
3. DURIAN Rp. 50.000
4. APEL Rp. 8.000
=====
```

Gambar 2.2.5

```
E:\Program\C++\c++\BUAH2 MODUL 2 ALPRO.exe
YANG DI BELI MANGGA
Masukan Jumlah Pesanan : 2
Total Harga : Rp. 32000
Uang yang Dibayar : Rp. 50000
Kembali : Rp. 18000
Masih ada Y/T : Y
=====
TOKO BUAH PRAYOGA BETRIO L
=====
DAFTAR HARGA BUAH
1. MANGGA Rp. 16.000
2. SEMANGKA Rp. 12.000
3. DURIAN Rp. 50.000
4. APEL Rp. 8.000
=====
Masukan No Menu Pilihan : 2
YANG DI BELI SEMANGKA
Masukan Jumlah Pesanan : 1
Total Harga : Rp. 12000
Uang yang Dibayar : Rp. 15000
Kembali : Rp. 3000
Masih ada Y/T : T
Terimah Kasih SUDAH KE TOKO PRAYOGA BETRIO L
=====
Process exited after 77.01 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.2.6