

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



NAMA : BRYANT APRILIAN BAHAN
NIM : DBC 118 100
KELAS : A
MODUL : II (PEWARISAN)

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

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*).

Dengan pewarisan dapat diciptakan suatu kelas baru yang mana kelas tersebut mewarisi seluruh sifat kelas dasar yang mempunyai akses public atau protected ditambah sifat khusus dari kelas yang bersangkutan. Ketentuan ini tidak berlaku sebaliknya, artinya sifat yang ada pada kelas turunan tidak diwariskan pada kelas dasar.

Ketentuan utama dengan adanya pewarisan yaitu memungkinkan suatu kode yang telah ditulis mudah sekali untuk digunakan kembali. Anda telah membuat suatu kelas yang telah diuji. Jika suatu nanti, anda ingin menerapkan kode tersebut pada sesuatu yang mempunyai sifat-sifat kelas tersebut, anda tinggal mewariskan kelas yang telah ada di kelas baru (kelas turunan). Anda tidak perlu mengotak-atik kelas yang teruji, sehingga efek samping yang tidak diharapkan tidak terjadi. Dengan cara ini pengembangan program menjadi lebih efisien dan menghemat waktu. Selain itu, anda dapat menambahkan sifat-sifat baru yang tidak ada pada kelas dasar atau bahkan dapat mengganti sifat-sifat pada kelas turunan, yang berbeda dengan sifat kelas dasar.

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*)

PEWARISAN TUNGGAL (SINGLE INHERITANCE)

Adalah pewarisan yang mana jumlah kelas dasarnya tunggal. Pada pewarisan ini, kelas turunan dapat berjumlah lebih dari satu.

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.

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.

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.

BAB II

LANGKAH KERJA

1. Buatlah program seperti di bawah ini!

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

class Point {
public:
    Point(float = 0.0, float = 0.0);
    void cetakPoint();
protected:
    float x,y;
};

Point::Point(float a, float b)
{
    cout<< "Konstruktor Point dijalankan "<<endl;
    x = a;
    y = b;
}

void Point::cetakPoint()
{
    cout<< "Point ["<<x<< ", "<<y<< "]"<<endl;
}

class Circle : public Point {
public:
    Circle(float r = 0.0, float a = 0.0, float b =
0.0); //konstruktor
    float area();
    void cetakPoint();
protected:
```

```

    float radius,l;
};
Circle::Circle(float r, float a, float b)
Point(a, b){
    cout<< "Konstruktor Circle dijalankan"<<endl;
    radius = r;
}
float Circle::area()
{
    l=3.14*radius*radius;
    return l;
}
void Circle::cetakPoint()
{
    Point::cetakPoint();
    cout<< "Circle dgn r : "<<radius<< " Center : ["<<x<< ",
"<<y<<"]"<<endl;

cout<< "Luas circle : "<<l<<endl;
}
class Cylinder:public Circle
public:

Cylinder(float h = 0.0, float r = 0.0, float a = 0.0, float b
= 0.0);//konstruktor
    float area();
    float vol();
    void cetakPoint();
protected:
    float height,hv;
};
Cylinder::Cylinder(float h, float r, float a, float b)
: Circle(r, a, b)

```

```
{
    cout<< "Konstruktor Cylinder dijalankan"<<endl;
    height = h;
}

float Cylinder::area()
{
    l= (2 * Circle::area() + 2*3.14*radius*height);
    return l;
}

float Cylinder::vol()
{
    v= (Circle::area() * height);
    return v;
}

void Cylinder::cetakPoint()
{
    Circle::cetakPoint();
    cout<< "Tinggi tabung (h) : "<< height<<endl;
    cout<< "Luas tabung : "<<l<<endl;
    cout<< "Volume tabung : "<<v<<endl;
}

int main()
{
    Point p(1.1, 2.2);
    cout<<endl;
    Circle lingkaran(10, 5, 5);
    lingkaran.area();
    lingkaran.cetakPoint();
    cout<<endl;
    Cylinder tabung(20, 30, 3, 4);
    tabung.area();
    tabung.vol();
    tabung.cetakPoint();
}
```

```
    getch();  
    return 0;  
}
```

2. Buatlah program untuk menghitung luas segitiga, luas persegi panjang, dan luas jajargenjang. Program tersebut harus menerapkan prinsip pewarisan dan harus menerapkan pilihan menu!

3. Buatlah 2 program dengan tema buah bagi yang nomor ganjil dan jika nomor genap maka tema hewan. Program dibuat menggunakan prinsip pewarisan.

BAB II PEMBAHASAN

1. Program Pertama

```
#include<iostream.h>
#include<conio.h>
class Point {
public:
    Point(float = 0.0, float = 0.0);
    void cetakPoint();
protected:
    float x,y;
};
Point::Point(float a, float b)
{
    cout<<"Konstruktor Point dijalankan "<<endl;
    x = a;
    y = b;
}
void Point::cetakPoint()
{
    cout<<"Point          : "<<'['<<x<<" "<<y<<']'<<endl;
}
class Circle:public Point {
public:
    Circle(float r = 0.0, float a = 0.0, float b =
0.0); //konstruktor
    float area();
    void cetakPoint();
protected:
    float radius,l;
};
Circle::Circle(float r, float a, float b)
```



```

: Point (a,b) {
    cout<<"Konstruktor Circle dijalankan"<<endl;
    radius = r;
}
float Circle::area()
{
    l=3.14*radius*radius;
    return l;
}
void Circle::cetakPoint()
{
    Point::cetakPoint();
    cout<<"Circle dgn r      : "<<radius<<" Center :
["<<x<<","<<y<<"]"<<endl;
    cout<<"Luas circle      : "<<l<<endl;
}
class Cylinder:public Circle {
public:
    Cylinder(float h = 0.0, float r = 0.0, float a = 0.0,
float b = 0.0);//konstruktor
    float area();
    float vol();
    void cetakPoint();
protected:
    float height,l,v;
};
Cylinder::Cylinder(float h, float r, float a, float b)
: Circle (r, a, b)
{
    cout<<"Konstruktor Cylinder dijalankan"<<endl;
    height = h;
}
float Cylinder::area()
{
    l= (2*Circle::area() + 2*3.14*radius*height);
    return l;
}
float Cylinder::vol()
{
    v= (Circle::area()*height);
    return v;
}

```

```

}
void Cylinder::cetakPoint()
{
    Circle::cetakPoint();
    cout<< "Tinggi tabung (h) : "<< height<<endl;
    cout<< "Luas tabung      : "<< l<<endl;
    cout<< "Volume tabung    : "<< v<<endl;
}
int main()
{
    Point p(1.1 , 2.2);
    cout<<endl;
    Circle lingkaran(10, 5, 5);
    lingkaran.area();
    lingkaran.cetakPoint();
    cout<<endl;
    Cylinder tabung(20, 30, 3, 4);
    tabung.area();
    tabung.vol();
    tabung.cetakPoint();
    getch();
    return 0;
}

```

Penjelasan :

Awal program dimulai dengan pemberian hak akses agar dapat menggunakan perintah dari isi library yang akan digunakan yaitu `<iostream.h>` dan `<conio.h>`. Kemudian ada pendeklarasian kelas pertama yaitu kelas Point. Dalam kelas point ini berisikan akses public yang mempunyai sebuah constructor, fungsi cetakPoint() dengan tipe data void. Sedangkan pada akses protected ada 2 variabel x dan y yang bertipe data float. Konstruktor dari kelas Point mempunyai 2 parameter variabel yang bertipe data float dengan presisi 1 angka dibelakang koma. Fungsi cetakPoint berguna untuk mencetak nilai dari variabel yang ada pada kelas Point. Hak akses Protected digunakan karena dengan menggunakan tipe akses ini variabel dari kelas Point hanya dapat diakses oleh kelas – kelas

turunan yang akan dibuat nanti. Ada 2 variabel pada kelas point yaitu x dan y yang bertipe float. Float merupakan salah satu tipe data yang dapat menampung nilai decimal. Selanjutnya ada bagian untuk menampilkan konstruktor untuk kelas Point, konstruktor ini mempunyai 2 parameter yaitu parameter a dan b bertipe float. Saat konstruktor ini di panggil maka akan menampilkan data string dan kemudian memasukkan nilai dari parameter konstruktor tadi kedalam variabel x dan y. Kemudian ada fungsi Point::cetakPoint() yang merupakan fungsi dari kelas Point, fungsi ini bertipe data void yang berarti fungsi ini tidak memberikan nilai balikan. Saat dipanggil fungsi ini akan mencetak nilai dari variabel x dan y.

Selanjutnya ada kelas turunan, yaitu kelas Circle yang merupakan turunan dari kelas Point pada kelas Circle telah dideklarasikan 2 buah fungsi dan 2 buah variabel. Untuk konstruktor dari kelas Circle mempunyai 3 parameter yaitu a, b, dan r yang bertipe float. Sedangkan untuk fungsi area() dari kelas Circle mempunyai nilai kembalian yaitu float. Fungsi ini digunakan untuk menghitung nilai luas dari objek Circle dan fungsi cetakPoint() tidak mempunyai nilai balikan karenan bertipe data void. Tugas dari fungsi ini adalah mencetak nilai dari keluaran fungsi area(). Untuk konstruktor dan fungsi tersebut menggunakan tipe akses public, agar dapat diakses oleh siapa saja, sedangkan fungsi radius memiliki tipe akses protected. Dalam class circle terdapat pendeklarasian fungsi dari konstruktor, konstruktor ini mempunyai 3 parameter dan saat dipanggil konstruktor dari kelas Circle akan mencetak kalimat yang telah ditulis pada program dan memasukan nilai dari variabel r kedalam nilai radius. Selanjutnya ada fungsi area() yang ada pada kelas Circle berberisi rumus untuk menghitung nilai dari luas lingkaran. Hasil dari rumus yang ada itu nanti akan di masukkan kedalam nilai dari variabel l dan saat selesai fungsi area() ini akan memberikan nilai kembalian dari nilai variabel l. fungsi ini sendiri bertipe float yang artinya dapat menampung nilai decimal hingga beberapa angka dibelakang koma. Setelah itu ada fungsi cetakPoint yang merupakan fungsi dari kelas turunan Circle fungsi ini bertipe data void yang berarti tidak akan memberikan nilai kembalian apapun. Saat fungsi ini dipanggil fungsi ini akan memanggil fungsi dari kelas utama yaitu fungsi cetakPoint dari kelas Point. Kemudian akan mencetak nilai dari variabel radius , x, y, dan variabel l yang merupakan nilai dari luas lingkaran.

Pada tahap selanjutnya ada kelas cylinder yang merupakan turunan lagi yang diturunkan dari kelas circle yang merupakan kelas turunan dari kelas utama point. Dengan tipe akses public ada 3 fungsi dan sebuah konstruktor pada kelas yang mempunyai 4 parameter yaitu h, r, a, dan b yang bertipe data float dengan presisi 1 angka dibelakang koma. Dan dengan tipe akses protected mempunyai 3 variabel dengan tipe data float yaitu height, l, dan v yang merupakan sifat yang hanya dimiliki oleh kelas cylinder. Fungsi area() digunakan untuk menghitung nilai dari luas lingkaran pada kelas Cylinder. Fungsi vol() digunakan untuk menghitung nilai dari volume dari objek Cylinder. Sedangkan fungsi cetakpoint() berfungsi untuk mencetak hasil dari fungsi – fungsi tadi yang telah dimasukkan kedalam variabel yang ada. Dengan tipe akses protected ada 3 variabel yang bertipe float untuk menampung nilai luas lingkaran, tinggi tabung dan volume tabung. Pada pendeklarasian konstruktor Cylinder ada 4 parameter pada konstruktor ini yaitu h, r, a, dan b yang semuanya bertipe float. Saat konstruktor ini dipanggil maka akan di tampilkan sebuah kalimat yang telah ditulis sebelumnya pada kode program. Selain itu akan nilai variabel height adalah nilai dari parameter h. Pada fungsi area() digunakan untuk menghitung nilai dari luas Cylinder, saat dijalankan fungsi ini akan mengembalikan nilai dari variabel l yang bertipe float pada program. Pada fungsi vol pada kelas Cylinder digunakan untuk menghitung nilai dari volume tabung. Saat dijalankan fungsi ini akan mengembalikan nilai dari variabel v yang bertipe float. Dan terakhir adalah fungsi dari kelas Cylinder yang berfungsi untuk mencetak nilai dari variabel height, l, dan v. Saat dipanggil fungsi ini terlebih dahulu akan memanggil fungsi dari kelas Circle yaitu cetakPoint().

Pada bagian terakhir program yaitu merupakan inti dari program 1, pada inti program ini berisi pendeklarasian dari nilai variabel dan pembuatan objek oleh konstruktor – konstruktor yang ada. Dalam program inti ini berfungsi memanggil fungsi – fungsi yang telah dideklarasikan dari tiap kelas. Program ini bertujuan untuk simulasi penggunaan kelas, kelas turunan dan kelas turunan dari kelas turunan.

2. Program Kedua

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

class bangun
{
protected:
    double x,y,z;
public:
    virtual void get_data()=0;
    virtual void display_area()=0;
};

class persegi_panjang : public bangun
{
public:
    void get_data(void)
    {
        cout<<"\n\nMenghitung Luas Persegi Panjang\n\n";
        cout<<"Panjang = "; cin>>x;
        cout<<"Lebar   = "; cin>>y;
    }
    void display_area(void)
    {
        double luas;
        luas = x * y;
        cout<<"Luas Persegi Panjang = "<<luas;
    }
};

class segitiga : public bangun
{
public:
    void get_data(void)
    {
        cout<<"\n\nMenghitung Luas Segitiga\n\n";
        cout<<"Alas    = "; cin>>x;
        cout<<"Tinggi = "; cin>>y;
    }
    void display_area(void)
    {
        double luas;
        luas = 0.5 * x * y;
        cout<<"Luas Segitiga = "<<luas;
    }
};

class jajaran_genjang : public bangun
{
public:
    void get_data(void)
    {
        cout<<"\n\nMenghitung Luas Jajaran Genjang\n\n";
        cout<<"Alas    = "; cin>>x;
        cout<<"Tinggi = "; cin>>y;
    }
};
```

```

void display_area(void)
{
    double luas;
    luas = 0.5 * x * y;
    cout<<"Luas Jajaran Genjang = "<<luas;
}

};

int main()
{
    system("cls");
    persegi_panjang panjang;
    segitiga tiga;
    jajaran_genjang genjang;

    bangun *list[4];

    list[0]=&panjang;
    list[1]=&tiga;
    list[2]=&genjang;

    int pilih;
    while(1)
    {
        system("cls");
        cout<<"\nMenghitung Luas Bangun Datar\n";
        cout<<"1) Luas Persegi Panjang\n";
        cout<<"2) Luas Segituga\n";
        cout<<"3) Luas Jajaran Genjang\n";
        cout<<"4) Keluar\n";
        cout<<"Masukkan Pilihan:-";
        cin>>pilih;

        if      (pilih==1){
            list[0]->get_data();
            list[0]->display_area();
            getch();
        }
        else if (pilih==2){
            list[1]->get_data();
            list[1]->display_area();
            getch();
        }
        else if (pilih==3){
            list[2]->get_data();
            list[2]->display_area();
            getch();
        }
        else
            exit(4);
    }
}

```

Penjelasan :

Program kedua menggunakan header `<iostream.h>` dan `<conio.h>`. Program ini memiliki kelas utama yaitu class bangun. Kelas ini memiliki tipe akses public yang berisi fungsi untuk memberikan nilai input. Lalu pada konstruktor untuk memberikan nilai awal p menjadi 0 setelah itu ada fungsi void untuk meminta input panjang dan tidak memberikan nilai balik karena bertipe void.

Program utama *int main()* memiliki nilai balikan dengan tipe data integer. Program utama ini akan menampilkan menu dan memanggil fungsi-fungsi pada kelas-kelas yang ada. Program ini bertujuan untuk menghitung luas persegi panjang, segitiga dan jajargenjang.

3. Program Ketiga

- Program Pertama

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

class hewan
{
protected:

    int jumlahKaki;
    char nama[20];
    char warna[20];

public:
    //setter
    void setJumlahKaki(int jumlahKaki)
    {
        this->jumlahKaki = jumlahKaki;
    }
    void setNama(char n[20])
    {
        strcpy(nama,n);
    }
    void setWarna(char w[20])
    {
        strcpy(warna,w);
    }

    //getter
    int getJumlahKaki()
    {
        return jumlahKaki;
    }

    char *getNama()
    {
        return nama;
    }
    char *getWarna()
    {
        return warna;
    }
};
```



```

//class turunan

class ikan: public hewan
{
private:
    int sirip;
public :
    void setSirip(int s)
    {
        sirip = s;
    }
    int getSirip()
    {
        return sirip;
    }
};

int main( )
{
    ikan ikanSaya;
    ikanSaya.setNama("Nigga");
    ikanSaya.setSirip(2);
    ikanSaya.setJumlahKaki(0);
    ikanSaya.setWarna("Hitam");

    cout<<"===== Biodata Ikan Saya ===== " <<endl;
    cout<<" Nama          : " <<ikanSaya.getNama() <<endl;
    cout<<" Warna          : " <<ikanSaya.getWarna()<<endl;
    cout<<" Jumlah Sirip    : " <<ikanSaya.getSirip()<<endl;
    cin.get();
};

```

- **Penjelasan**

A. `#include<string>` memuat fungsi sebagai berikut:

String merupakan library yang menyimpan fungsi-fungsi yang digunakan untuk menangani string ataupun substring. Fungsi – fungsi dalam library `string.h` antara lain : `strcpy`, `strncpy`, `strcat`, `strncat`, `strcmp`, `strncmp`, `strlen`, `strchr`

B. `Void` adalah fungsi yang hanya dimaksudkan untuk menampilkan suatu keterangan saja.

C. `Int` adalah tipe data yang memiliki nilai berupa angka. Bagian ini yang berfungsi sebagai identitas pada program ini.

D. `Protected` adalah label yang berfungsi untuk menentukan sifat akses semua member yang mengikutinya, sehingga memiliki sifat yang tidak dapat diakses dari luar class tapi masih dapat diakses dari dalam class maupun anak class (derived class).

E. `Char` adalah salah satu tipe data sembarang huruf, angka, dan simbol. Yang memungkinkan kita untuk memesan memori yang dapat menyimpan nilai dalam bentuk karakter tunggal seperti 'a', '\$' dan lain-lain. `char x= 'A' ;` //Variabel Char hanya dapat memuat satu karakter.

F. Program ini juga menggunakan sebuah “class hewan” sebagai class utama, dan “class ikan” sebagai turunannya

- Program Kedua

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

class hewan
{
protected:

    int jumlahKaki;
    char nama[20];
    char warna[20];

public:
    //setter
    void setJumlahKaki(int jumlahKaki)
    {
        this->jumlahKaki = jumlahKaki;
    }
    void setNama(char n[20])
    {
        strcpy(nama,n);
    }
    void setWarna(char w[20])
    {
        strcpy(warna,w);
    }

    //getter
    int getJumlahKaki()
    {
        return jumlahKaki;
    }

    char *getNama()
    {
        return nama;
    }
    char *getWarna()
    {
        return warna;
    }
};
```

```

//class turunan
class ayam :public hewan
{
    private:
        int paruh;
    public:
        void setParuh(int p)
        {
            paruh = p;
        }

        int getParuh()
        {
            return paruh;
        }
};

int main( )
{
    ayam ayamSaya;
    ayamSaya.setNama("Rambo");
    ayamSaya.setParuh( 1 );
    ayamSaya.setJumlahKaki( 2 );
    ayamSaya.setWarna("Merah Burik");

    cout<<"===== Biodata Ayam Saya ===== " <<endl;
    cout<<" Nama          : " <<ayamSaya.getNama() <<endl;
    cout<<" Warna          : " <<ayamSaya.getWarna()<<endl;
    cout<<" Jumlah Paruh    : " <<ayamSaya.getParuh()<<endl;
    cout<<" Jumlah Kaki     : " <<ayamSaya.getJumlahKaki()<<endl;
    cin.get();
};

```

- **Penjelasan**

G. *#include<string>* memuat fungsi sebagai berikut:

String merupakan library yang menyimpan fungsi-fungsi yang digunakan untuk menangani string ataupun substring. Fungsi – fungsi dalam library string.h antara lain : strcpy, strncpy, strcat, strncat, strcmp, strncmp, strlen, strchr

H. Void adalah fungsi yang hanya dimaksudkan untuk menampilkan suatu keterangan saja.

I. Int adalah tipe data yang memiliki nilai berupa angka. Bagian ini yang berfungsi sebagai identitas pada program ini.

J. Protected adalah label yang berfungsi untuk menentukan sifat akses semua member yang mengikutinya, sehingga memiliki sifat yang tidak dapat diakses dari luar class tapi masih dapat diakses dari dalam class maupun anak class (derived class).

K. Program ini juga menggunakan sebuah “class hewan” sebagai class utama, dan “class ayam” sebagai turunannya

BAB IV

KESIMPULAN

1. Pewarisan berfungsi untuk mewarisi sifat/atribut pada kelas turunannya.
2. Tiap kelas memiliki tipe akses berbeda beda.
3. Tipe akses public bersifat bebas digunakan oleh semua kelas.
4. Tipe akses protected bersifat hanya dapat diakses oleh kelas turunannya.
5. Kelas turunan dapat memiliki kelas turunan lagi.
6. Kelas turunan dapat menggunakan fungsi yang ada pada kelas induknya.
7. Fungsi dari setiap kelas dapat dipanggil setiap saat.
- 8.

Dikaguyuk



DAFTAR PUSTAKA

Dosen TI. 2020. Modul Praktikum Algoritma & Pemrograman II.

Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya.

Nurmaulida, Alfina. 2014. *Inheritance turunan atau pewarisan*.

<https://zegyjib.wordpress.com/matkul/oop2/oop2-inheritance-turunan-atau-pewarisan-c-java-c/> (diakses pada Senin, 13 April 2020 pukul 14:16 WIB)

Karmatoli, Puby. 2010. *Pewarisan dalam C++*.

<https://12puby.wordpress.com/2010/06/09/pewarisan-dalam-c/> (diakses pada Senin, 13 April 2020 pukul 15:10 WIB)

Lampiran



gambar 1.2 ..