

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



NAMA : REDO
NIM : 193020503017
KELAS : A
MODUL : II(PEWARISAN)

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

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



Nama : Redo
NIM : 193020503017
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

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

1.1. TUJUAN

Setelah menyelesaikan modul ini mahasiswa di harapkan mampu untuk :

1. kelas baru dari kelas yang sudah ada dengan pewarisan.

1.2. LANDASAN TEORI

Pewarisan atau bisa di bilang juga inheritance adalah sebuah cara yang memiliki konsep untuk memberikan object yang di miliknya ke pada turunan yang sudah di tentukan sehingga turunan itu akan bisa memakai object yang sudah di berikan sebelumnya.

Dalam C++ pewarisan sangat lah berguna untuk PBO, dengan pewarisan kita tinggal mewariskan object yang berguna untuk turunannya tanpa harus membuat lagi object tersebut. Contoh pada bangunan 2 dimensi yang memiliki luas, jadi tanpa harus membuat kembali variabel yang akan menampung luas akan lebih mudah untuk di wariskan dan di pakai kembali. Format pewarisan pada C++ adalah dengan menambahkan nama class yang ingin diwariskan pada class anaknya dengan menambahkan terlebih dahulu titik dua (:).

Contoh code pada C++ :

```
class keturunan: public induk;
```

Pada C++ akan ada sebuah masalah ketika class induk dan class turunan memiliki nama object yang sama maka jika akan memanggil object tersebut akan di panggil object yang berada di class induk karena pointer yang di miliki tertuju ke variabel yang berada di class induk, maka dari itu kita harus menggunakan virtual methode yang akan membedakan pointer object walaupun object tersebut ber nama sama. Format untuk penggunaan virtual methode adalah dengan menggunakan keyword virtual pada object yang mempunya nama yang sama.

*ditambah
Lagi
Landasan
teorinya*

Contoh code pada C++ :

```
virtual void object();  
virtual int object;
```

Dan object yang akan bisa di wariskan hanyalah object yang memiliki akses protected atau public, yang artinya akses tersebut berfungsi supaya object tersebut dapat di akses turunannya sedangkan object yang memiliki akses private tidak dapat di wariskan karena hanya akan bisa digunakan pada class tersebut saja.

BAB II

PEMBAHASAN

2.1. Program Pertama

Code program :

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

using namespace std;

string a;
string b;
string c;

input(){

    system("cls");
    cout<<"Masukan warna induk pertama\t= "; cin.get();
    getline(cin, a);
    cout<<"Masukan warna induk kedua\t= "; getline(cin,b);

}

third(){
    c = a+" dan "+b;
}

class induk{
public:
    induk(string, string);
    print();
protected:
    string warnal,warna2;
```

```

};

    induk::print() {
        system("cls");
        cout<<"warna induk ke satu\t= "<<warna1<<endl;
        cout<<"warna induk ke dua\t= "<<warna2<<endl;
    }
    induk::induk(string indukA, string indukB){
        warna1 = indukA;
        warna2 = indukB;
    }
class keturunan1: public induk{
public :
    keturunan1(string, string, string);
    cetak();
private :
    string kombinasi;
};

keturunan1::cetak() {
    system("cls");
    cout<<"Kemungkinan Warna Keturunan =\n\n";
    cout<<"1. "<<warna1<<endl;
    cout<<"2. "<<warna2<<endl;
    cout<<"3. "<<kombinasi<<endl;
}

keturunan1::keturunan1(string indukA, string indukB, string
combine):induk(indukA, indukB){
    kombinasi=combine;
}

bool isEmpty() {
    if(a=="") {
        return 1;
    }else{
        return 0;
    }
}

```

```

}
}
int main(){
    int pilih;
    cout<<"===== >>Program Kemungkinan
Warna Keturunan Hewan<<=====\\n";
    cout<<"Nama = Redo"<<endl;
    cout<<"Nim  = 193020503017"<<"\\n\\n";
    system("pause");
    do{
        system("cls");
        cout<<"MENU PROGRAM\\n\\n";
        cout<<"1. Masukkan Warna Induk\\n";
        cout<<"2. Hasil Keturunan\\n";
        cout<<"3. Tutup\\n";
        cout<<"pilih = "; cin>>pilih;
        system("cls");
        switch(pilih){
            case 1:{
                input();
                induk gen1(a, b);
                gen1.print();
                break;
            }
            case 2:{
                if(IsEmpty()==1){
                    cout<<"Data Induk Kosong!!!!";
                    break;
                }else{
                    third();
                    keturunan1 gen2(a, b, c);
                    gen2.cetak();
                    break;
                }
            }
        }
    }
}

```

```

    }
    case 3:{
        cout<<"\t\t\tTerima Kasih Telah
Menggunakan Aplikasi Ini";
        break;
    }
}
cout<<endl;
system("pause");
}while(pilih>0 and pilih<3);
}

```

Pembahasan code :

```

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

```

Code di atas berfungsi memanggil library-library yang ingin di gunakan supaya kita dapat menggunakan perintah-perintah yang di sediakan masing-masing library. Contoh library iostream menyediakan perintah cout dan cin.

Sedangkan code `using namespace std;` digunakan untuk memberi penamaan std pada perintah yang membutuhkan contoh perintah cout dan cin.

```

string a;
string b;
string c;

```

Code diatas memiliki kegunaan untuk mendeklarasikan variabel yang bernama a, b, dan c yang semua memiliki tipe data string yang akan memiliki

akses public atau global karena di deklarasikan di luar fungsi atau program utama.

```
input() {  
  
    system("cls");  
    cout<<"Masukan warna induk pertama\t= "; cin.get();  
    getline(cin, a);  
    cout<<"Masukan warna induk kedua\t= "; getline(cin,b);  
  
}  
  
third() {  
    c = a+" dan "+b;  
}
```

Code diatas akan membuat fungsi yang bernama input dan third.

kegunaan dari fungsi input adalah untuk menginisiasikan data yang ingin dimasukan dengan menggunakan beberapa perintah cout dan cin yang akan menginisiasikan element variabel yang sudah di tentukan.

kegunaan dari fungsi third adalah untuk menginisiasikan variabel c dengan gabungan dari variabel string a, text "*dan*", dan b.

```
class induk{  
  
    public:  
        induk(string, string);  
        print();  
    protected:  
        string warna1, warna2;  
  
};
```

Code di atas akan di gunakan untuk membuat class yang bernama *induk* yang akan menampung object-object yang sudah dideklarasikan di dalam class tersebut.

```
induk::print(){
    system("cls");
    cout<<"warna induk ke satu\t= "<<warna1<<endl;
    cout<<"warna induk ke dua\t= "<<warna2<<endl;
}
induk::induk(string indukA, string indukB){
    warna1 = indukA;
    warna2 = indukB;
}
```

Code di atas berguna untuk membuat isi dari fungsi dan method yang di miliki class *induk* yang bernama *print* dan *induk* .

Kegunaan dari fungsi *print* adalah untuk mencetak semua variabel yang ingin kita cetak.

Kegunaan dari method *induk* adalah dengan menginisiasikan data dengan variabel parameter nya.

```
class keturunan1: public induk{
    public :
        keturunan1(string, string, string);
        cetak();
    private :
        string kombinasi;
};
```

Code diatas di gunakan untuk membuat class yang bernama *keturunan1* yang sudah mewarisi object yang di berikan oleh class *induk* yang juga akan menampung object-object yang sudah di deklarasikan pada class tersebut.

```

keturunan1::cetak() {
    system("cls");
    cout<<"Kemungkinan Warna Keturunan =\n\n";
    cout<<"1. "<<warna1<<endl;
    cout<<"2. "<<warna2<<endl;
    cout<<"3. "<<kombinasi<<endl;
}
keturunan1::keturunan1(string indukA, string indukB, string
combine):induk(indukA, indukB){
    kombinasi=combine;
}

```

Code diatas berguna untuk mengisi fungsi dan methode yang di miliki oleh class *keturunan1* yang memiliki nama *cetak* dan *keturunan1*.

Kegunaan dari fungsi *cetak* adalah mencetak variabel yang sudah kita tentukan untuk di cetak ke layar dengan menggunakan perintah utamanya yaitu cout.

Kegunaan dari methode *keturunan1* adalah menginisiasikan variabel dengan variabel parameternya yang beberapa juga di ambil dari pewarisan yang sudah dilakukan.

```

bool isEmpty(){
    if(a==""){
        return 1;
    }else{
        return 0;
    }
}

```

Code tersebut berguna untuk membuat fungsi bernama *IsEmpty* yang bertipe boolean yang akan mengmbalikan nilai true jika element dari variabel a kosong sedangkan jika tidak terpenuhi akan mengembalikan nilai false.

```

int main(){
    int pilih;
    cout<<"=====>>Program Kemungkinan
Warna Keturunan Hewan<=====\\n";
    cout<<"Nama = Redo"<<endl;
    cout<<"Nim  = 193020503017"<<"\\n\\n";
    system("pause");
    do{
        system("cls");
        cout<<"MENU PROGRAM\\n\\n";
        cout<<"1. Masukan Warna Induk\\n";
        cout<<"2. Hasil Keturunan\\n";
        cout<<"3. Tutup\\n";
        cout<<"pilih = "; cin>>pilih;
        system("cls");
        switch(pilih){
            case 1:{
                input();
                induk gen1(a, b);
                gen1.print();
                break;
            }
            case 2:{
                if(IsEmpty()==1){
                    cout<<"Data Induk Kosong!!!!";
                    break;
                }else{
                    third();
                    keturunan1 gen2(a, b, c);
                    gen2.cetak();
                    break;
                }
            }
            case 3:{

```

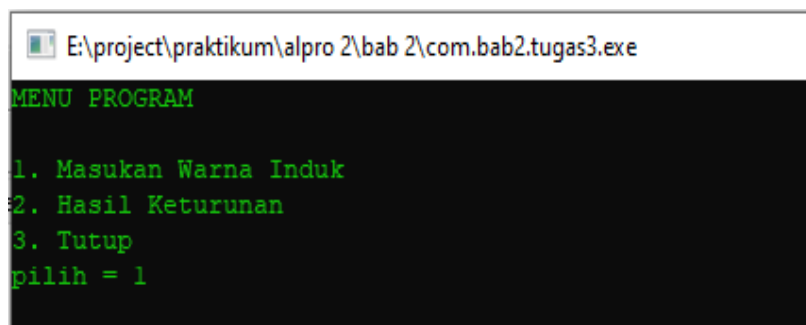
```

        cout<<"\t\t\tTerima Kasih Telah
Menggunakan Aplikasi Ini";
        break;
    }
}
cout<<endl;
system("pause");
}while(pilih>0 and pilih<3);
}

```

Code diatas adalah code dari program utama yang akan di jalankan dengan tujuan untuk mengkoordinasikan penggunaan fungsi-fungsi yang sudah di buat ke dalam menu dengan menggunakan cout untuk menampilkan text yang ingin di buat dan menggunakan pengulangan do untuk membungkus pemilihan kondisi switch yang akan bertindak menurut kondisi dari variabel *pilih* yang bertipe integer yang sudah di inisiasikan sebelumnya menggunakan perintah cin.

Output program utama :



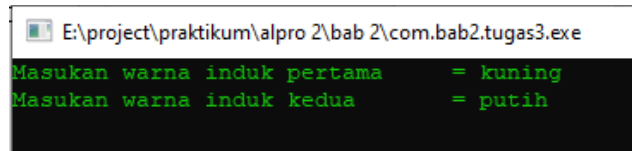
```

E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas3.exe
MENU PROGRAM
1. Masukan Warna Induk
2. Hasil Keturunan
3. Tutup
pilih = 1

```

Gambar 2.1 Program utama.

Pada kondisi pilih sama dengan 1 akan menjalankan perintah untuk menjalankan fungsi *input* dan dilanjutkan pada membuat penamaan pada class induk untuk menjalankan method yang akan di eksekusi ke fungsi *print*. Output keadaan pilih = 1 :

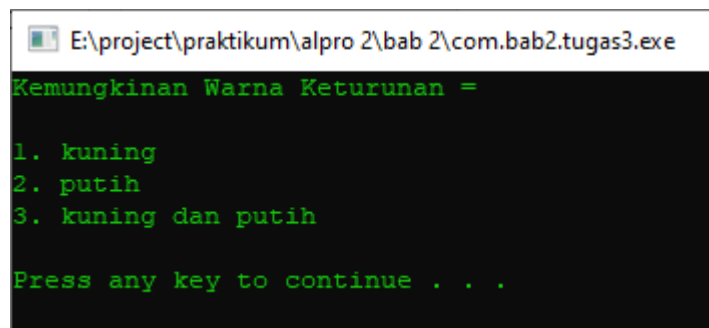


```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas3.exe
Masukan warna induk pertama = kuning
Masukan warna induk kedua = putih
```

Gambar 2.2 keadaan pilih = 1.

Pada kondisi pilih sama dengan 2 akan menjalankan perintah untuk menjalankan sebuah pemilihan if jika keadaan nilai balik fungsi *IsEmpty* bernilai true maka akan mencetak peringatan bahwa data yang dimiliki adalah data kosong, sedangkan pada kondisi selain di atas akan menjalankan fungsi *thrid* dilanjutkan dengan penamaan class *induk* untuk menjalankan methodnya yang akan di eksekusi dengan fungsi *cetak*.

Output keadaan pilih = 2 :

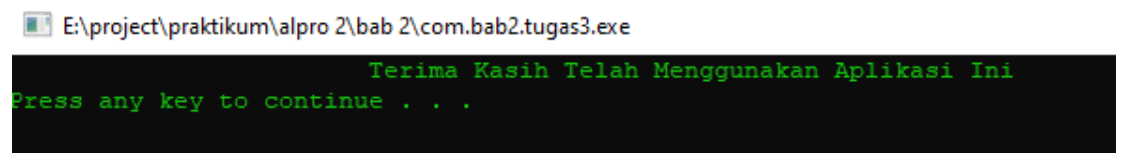


```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas3.exe
Kemungkinan Warna Keturunan =
1. kuning
2. putih
3. kuning dan putih
Press any key to continue . . .
```

Gambar 2.3 keadaan pilih =2.

Pada kondisi pilih sama dengan 3 akan menjalankan perintah untuk memberi tampilan teks bahwa program akan di tutup.

Output keadaan pilih=3 :



```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas3.exe
Terima Kasih Telah Menggunakan Aplikasi Ini
Press any key to continue . . .
```

Gambar 2.4 keadaan pilih=3.

Hal di atas akan di lakukan perulangan karena di bungkus dengan perulangan *do* sampai keadaan variabel *pilih*>0 and *pilih*<3 bernilai false.

2.2. Program Kedua

Code program

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

using namespace std;

string a;
string a1;
string b;
string b1;
string c;
string c1;

input(){

    system("cls");
    cout<<"Masukan Gen Induk Pertama\t= "; cin.get();
    getline(cin, a);
    cout<<"Sifat Gen\t\t\t= "; getline(cin, a1);
    cout<<"Masukan Gen Induk Kedua\t\t= "; getline(cin, b);
    cout<<"Sifat Gen\t\t\t= "; getline(cin, b1);
}

third(){

    c=a+b;
    if(a1=="merah" and b1=="kuning" or a1=="kuning" and
b1=="merah"){
        c1="oranye";
    }else if(a1=="biru" and b1=="merah" or a1=="merah" and
b1=="biru"){
```

```

        c1="ungu";
    }else if(a1=="biru" and b1=="kuning" or a1=="kuning" and
b1=="biru"){
        c1="hijau";
    }else if(a1=="merah" and b1=="hitam" or a1=="hitam" and
b1=="merah"){
        c1="merah tua";
    }else if(a1=="merah" and b1=="putih" or a1=="putih" and
b1=="merah"){
        c1="merah muda";
    }else if(a1=="biru" and b1=="hitam" or a1=="hitam" and
b1=="biru"){
        c1="biru tua";
    }else if(a1=="biru" and b1=="putih" or a1=="putih" and
b1=="biru"){
        c1="biru muda";
    }else if(a1=="kuning" and b1=="putih" or a1=="putih" and
b1=="kuning"){
        c1="kuning muda";
    }else if(a1=="putih" and b1=="hitam" or a1=="hitam" and
b1=="putih"){
        c1="abu";
    }
}

class induk{

    public:
        induk(string, string, string, string);

    private:

```



```
protected:

    string warnal, warna2, sifat1, sifat2;

};

    induk::induk(string indukA, string indukB, string sifatA,
string sifatB){

        warnal = indukA;
        warna2 = indukB;
        sifat1 = sifatA;
        sifat2 = sifatB;

    }

class keturunan1: public induk{

    public :

        keturunan1(string, string, string, string, string,
string);

        cetak();

    private :

        string kombinasi;
        string kombinasil;
```

```

};

keturunan1::cetak() {

    system("cls");
    cout<<"Gen induk ke satu\t= "<<warna1<<"
("<<sifat1<<")\n";
    cout<<"Gen induk ke dua\t= "<<warna2<<"
("<<sifat2<<")\n\n";
    cout<<"Kemungkinan Warna Keturunan 100% = "<<kombinasi<<"
("<<kombinasil<<") "<<endl;

}

keturunan1::keturunan1(string indukA, string indukB, string
sifatA, string sifatB, string combine, string
combine1):induk(indukA, indukB, sifatA, sifatB){

    kombinasi=combine;
    kombinasil=combine1;

}

bool IsEmpty(){

    if(a==""){
        return 1;
    }else{
        return 0;
    }

}

```

```

int main(){

    int pilih;

    cout<<"=====>>Program Keturunan Gen Hewan
Monohibrid<<=====\\n";
    cout<<"Nama = Redo"<<endl;
    cout<<"Nim  = 193020503017"<<"\\n\\n";
    system("pause");
    do{

        system("cls");

        cout<<"MENU PROGRAM\\n\\n";
        cout<<"1. Masukan Gen Warna Induk\\n";
        cout<<"2. Hasil Keturunan F1\\n";
        cout<<"3. Tutup\\n";
        cout<<"pilih = "; cin>>pilih;
        system("cls");
        switch(pilih){

            case 1:{

                input();
                induk gen1(a, b, a1, b1);
                break;

            }

            case 2:{

                if(IsEmpty()==1){

                    cout<<"Data Induk Kosong!!!!";

                    break;

```

```

        }else{

            third();
            keturunan1 gen2(a, b, a1, b1, c, c1);
            gen2.cetak();
            break;

        }
    }
    case 3:{
        cout<<"\t\t\tTerima Kasih Telah Menggunakan
Aplikasi Ini";
        break;
    }

}
cout<<endl;
system("pause");

}while(pilih>0 and pilih<3);
}

```

Pembahasan coding :

```

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

using namespace std;

```

Code di atas berfungsi memanggil library-library yang ingin di gunakan supaya kita dapat menggunakan perintah-perintah yang di sediakan masing-masing library. Contoh library iostream menyediakan perintah cout dan cin.

Sedangkan code `using namespace std;` digunakan untuk memberi penamaan std pada perintah yang membutuhkan contoh perintah cout dan cin.

```
string a;  
string a1;  
string b;  
string b1;  
string c;  
string c1;
```

Code diatas memiliki kegunaan untuk mendeklarasikan variabel yang bernama a, a1, b, b1, c dan c1 yang semua memiliki tipe data string yang akan memiliki akses public atau global karena di deklarasikan di luar fungsi atau program utama.

```
input() {  
  
    system("cls");  
    cout<<"Masukan Gen Induk Pertama\t= "; cin.get();  
    getline(cin, a);  
    cout<<"Sifat Gen\t\t\t= "; getline(cin, a1);  
    cout<<"Masukan Gen Induk Kedua\t\t= "; getline(cin, b);  
    cout<<"Sifat Gen\t\t\t= "; getline(cin, b1);  
}
```

Code diatas akan membuat fungsi yang bernama third yang kegunaannya untuk menginisiasikan data yang ingin dimasukan dengan menggunakan beberapa perintah cout dan cin yang akan menginisiasikan element variabel yang sudah di tentukan.

```
third() {  
  
    c=a+b;  
    if(a1=="merah" and b1=="kuning" or a1=="kuning" and  
b1=="merah") {  
        c1="oranye";  
    }
```

```

        }else if(a1=="biru" and b1=="merah" or a1=="merah" and
b1=="biru"){
            c1="ungu";
        }else if(a1=="biru" and b1=="kuning" or a1=="kuning" and
b1=="biru"){
            c1="hijau";
        }else if(a1=="merah" and b1=="hitam" or a1=="hitam" and
b1=="merah"){
            c1="merah tua";
        }else if(a1=="merah" and b1=="putih" or a1=="putih" and
b1=="merah"){
            c1="merah muda";
        }else if(a1=="biru" and b1=="hitam" or a1=="hitam" and
b1=="biru"){
            c1="biru tua";
        }else if(a1=="biru" and b1=="putih" or a1=="putih" and
b1=="biru"){
            c1="biru muda";
        }else if(a1=="kuning" and b1=="putih" or a1=="putih" and
b1=="kuning"){
            c1="kuning muda";
        }else if(a1=="putih" and b1=="hitam" or a1=="hitam" and
b1=="putih"){
            c1="abu";
        }
    }
}

```

Code diatas akan membuat fungsi yang bernama third yang memiliki kegunaannya untuk menginisiasikan variabel c dengan gabungan dari variabel string a, text "*dan*", dan b.

Selain itu juga akan menginisiasikan variabel c1 dengan element yang sudah di tentukan dengan menggunakan pemilihan untuk mendapat hasil yang akan dipengaruhi oleh kondisi variabel a1 dan b1.

```
class induk{

    public:
        induk(string, string, string, string);

    private:

    protected:

        string warna1, warna2, sifat1, sifat2;

};

    induk::induk(string indukA, string indukB, string
sifatA, string sifatB){

        warna1 = indukA;
        warna2 = indukB;
        sifat1 = sifatA;
        sifat2 = sifatB;

    }
```

Code diatas memiliki fungsi untuk membuat class yang bernama *induk* yang akan menampung object-object yang sudah di deklarasikan di dalamnya dan membuat perintah untuk method class itu sendiri dengan menginisiasikan variabel dengan variabel parameter fungsi itu sendiri.

```
class keturunan1: public induk{

    public :
        keturunan1(string, string, string, string,
string, string);
        cetak();

    private :
        string kombinasi;
        string kombinasil;

};

keturunan1::cetak() {

    system("cls");
    cout<<"Gen induk ke satu\t= "<<warna1<<"
("<<sifat1<<")\n";
    cout<<"Gen induk ke dua\t= "<<warna2<<"
("<<sifat2<<")\n\n";
    cout<<"Kemungkinan Warna Keturunan 100% =
"<<kombinasi<<" ("<<kombinasil<<")"<<endl;

}

keturunan1::keturunan1(string indukA, string indukB, string
sifatA, string sifatB, string combine, string
combine1):induk(indukA, indukB, sifatA, sifatB){
```



```

kombinasi=combine;
kombinasil=combine1;

}

```

Code diatas berguna untuk membuat class yang bernama *keturunan1* yang akan menampung object-object yang sudah di deklarasikan di dalamnya.

Class tersebut juga memiliki fungsi yang bernama cetak yang akan mencetak variabel yang sudah kita tentukan untuk ditampilkan ke layar.

konstruktor dari class tersebut berguna untuk menginisiasikan variabel dengan parameter konstruktor itu sendiri.

```

bool IsEmpty(){
    if(a==""){
        return 1;
    }else{
        return 0;
    }
}

```

Code tersebut berguna untuk membuat fungsi bernama *IsEmpty* yang bertipe boolean yang akan mengembalikan nilai true jika element dari variabel a kosong sedangkan jika tidak terpenuhi akan mengembalikan nilai false.

```

int main(){

    int pilih;
    cout<<"=====>>Program Keturunan Gen
Hewan Monohibrid<<"=====\\n";
    cout<<"Nama = Redo"<<endl;
    cout<<"Nim = 193020503017"<<"\\n\\n";
    system("pause");
    do{

```

```

system("cls");

cout<<"MENU PROGRAM\n\n";
cout<<"1. Masukan Gen Warna Induk\n";
cout<<"2. Hasil Keturunan F1\n";
cout<<"3. Tutup\n";
cout<<"pilih = "; cin>>pilih;
system("cls");
switch(pilih){

case 1:{

    input();
    induk gen1(a, b, a1, b1);
    break;

}

case 2:{

    if(IsEmpty()==1){

        cout<<"Data Induk Kosong!!!!";

        break;

    }else{

        third();
        keturunan1 gen2(a, b, a1, b1, c,
c1);

        gen2.cetak();
        break;

    }

}

```

```

    }
    case 3:{
        cout<<"\t\t\tTerima Kasih Telah
Menggunakan Aplikasi Ini";
        break;
    }

    }
    cout<<endl;
    system("pause");

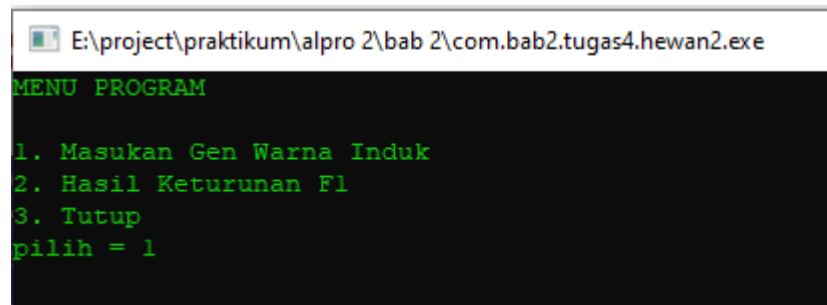
    }while(pilih>0 and pilih<3);
}

```

Code diatas adalah program utama yang akan mengkoordinasikan jalannya program dengan di buat berdasarkan konsep menu.

Algoritma dari code di atas adalah membuat menu dengan memakai perintah pemilihan switch berdasarkan kondisi variabel integer pilih yang sudah di deklarasikan dan di inisiasikan sebelumnya yang di bungkus kedalam perulangan do.

Output program utama :



```

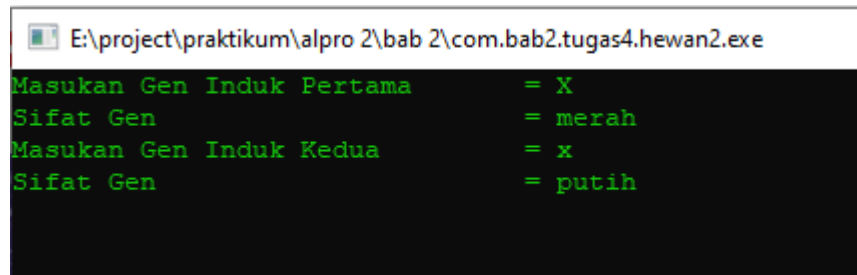
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas4.hewan2.exe
MENU PROGRAM
1. Masukan Gen Warna Induk
2. Hasil Keturunan F1
3. Tutup
pilih = 1

```

Gambar 2.5 Program Utama program ke-2

Kondisi yang akan muncul ketika element dari variabel *pilih* sama dengan 1 adalah dengan menjalankan fungsi *input* yang akan juga di buat ke penamaan untuk membuat konstrukt.

Output keadaan pilih=1 :

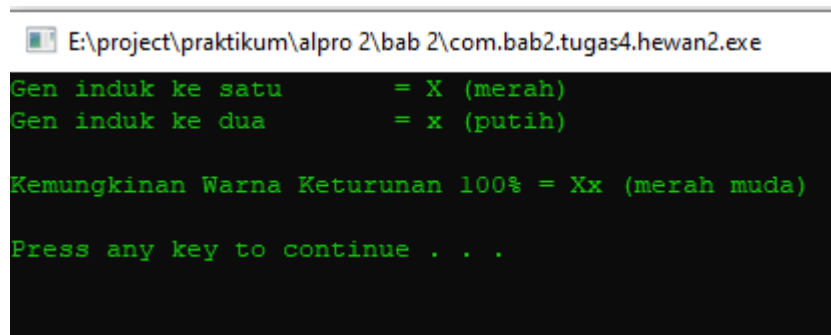


```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas4.hewan2.exe
Masukan Gen Induk Pertama      = X
Sifat Gen                       = merah
Masukan Gen Induk Kedua        = x
Sifat Gen                       = putih
```

Gambar 2.6 keadaan pilih=1 program ke-2

Kondisi yang akan terjadi jika variabel *pilih* = 2 adalah pemilihan dengan kondisi jika nilai balik fungsi *IsEmpty* sama dengan true maka akan mencetak text untuk memperingati bahwa data yang dimiliki adalah kosong, sedangkan selain kondisi di atas akan menjalankan fungsi *thrid* dilanjutkan dengan membuat penamaan untuk membuat konstruktor class *keturunan1* dan di eksekusi dengan fungsi *cetak*.

Output keadaan pilih=2 :



```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas4.hewan2.exe
Gen induk ke satu              = X (merah)
Gen induk ke dua               = x (putih)

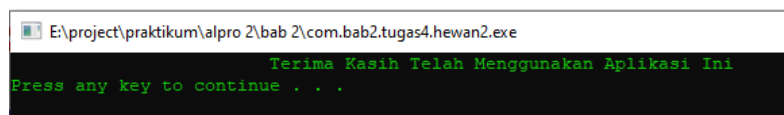
Kemungkinan Warna Keturunan 100% = Xx (merah muda)

Press any key to continue . . .
```

Gambar 2.7 keadaan pilih=2 program ke-2

Kondisi jika variabel *pilih* = 3 adalah akan menjalankan perintah untuk mencetak teks bahwa program akan di tutup.

Output keadaan pilih=3 :



```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas4.hewan2.exe
Terima Kasih Telah Menggunakan Aplikasi Ini
Press any key to continue . . .
```

Gambar 2.8 keadaan pilih=2 program ke-2

Semua perintah di atas di bungkus dengan perulangan `do` yang akan berhenti mengulang jika keadaan `pilih>0 and pilih<3` bernilai `false`.

BAB III

KESIMPULAN

Kesimpulan yang bisa saya ambil dari module ini adalah dengan menggunakan pewarisan akan membuat program akan lebih mudah untuk di buat tanpa harus mengulangi membuat object yang sama untuk di gunakan dengan menggunakan pewarisan kita hanya perlu membuat satu object yang akan di gunakan bisa digunakan oleh turunan.

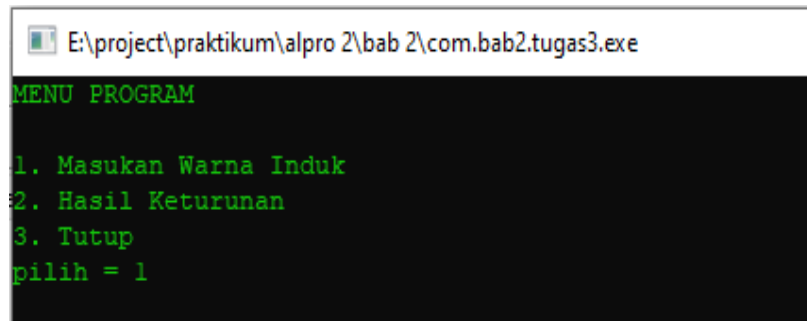
DAFTAR PUSTAKA

Dosen Teknik Informatika. Algoritma dan Pemrograman II. 2020. *Modul Praktikum Algoritma dan pemrograman II*. Universitas Palangka Raya. Fakultas Teknik. Jurusan Teknik Informatika.

Rachmat Santoso. 2017. *C++ - Inheritance (Dasar Pewarisan)*.
<http://www.nblognlife.com/2017/04/c-inheritance-dasar-pewarisan.html>
(diakses pada : rabu, 14 April 2020 pada pukul 19.17)

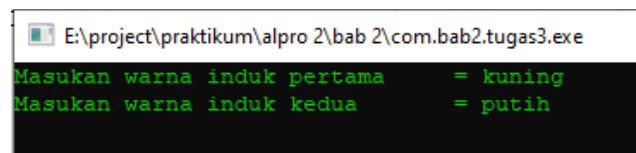
sinauarduino. 2016. *Inheritance Pada Classes C++*.
<https://www.sinauarduino.com/artikel/inheritance-pada-classes-cpp/> (diakses pada : rabu, 14 April 2020 pada pukul 19.48)

LAMPIRAN



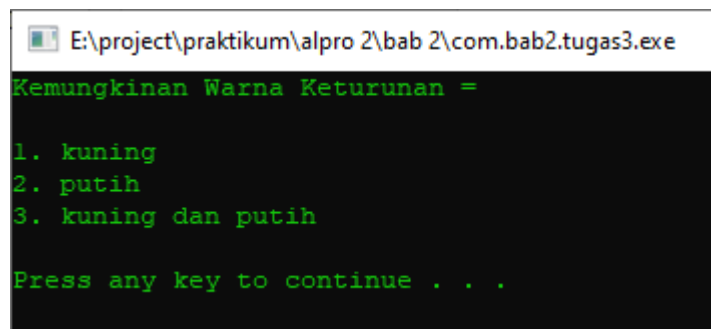
```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas3.exe
MENU PROGRAM
1. Masukan Warna Induk
2. Hasil Keturunan
3. Tutup
pilih = 1
```

Gambar 2.1 Program utama.



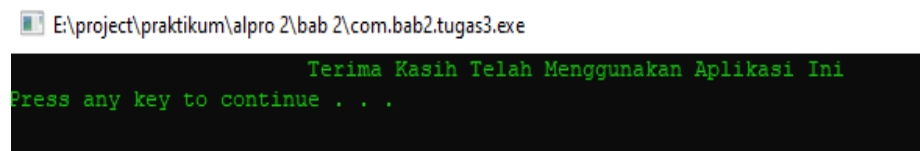
```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas3.exe
Masukan warna induk pertama = kuning
Masukan warna induk kedua = putih
```

Gambar 2.2 keadaan pilih = 1.



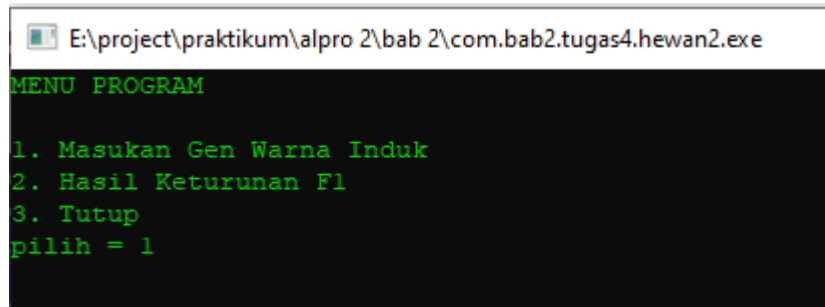
```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas3.exe
Kemungkinan Warna Keturunan =
1. kuning
2. putih
3. kuning dan putih
Press any key to continue . . .
```

Gambar 2.3 keadaan pilih =2.



```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas3.exe
Terima Kasih Telah Menggunakan Aplikasi Ini
Press any key to continue . . .
```

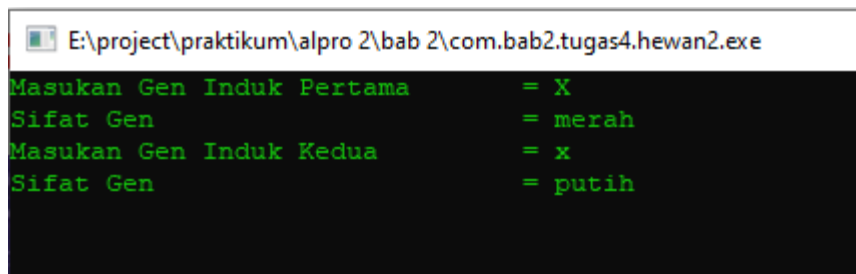
Gambar 2.4 keadaan pilih=3.



```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas4.hewan2.exe
MENU PROGRAM

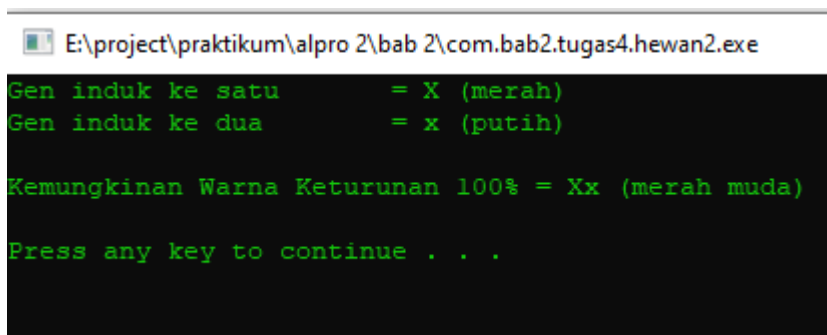
1. Masukan Gen Warna Induk
2. Hasil Keturunan F1
3. Tutup
pilih = 1
```

Gambar 2.5 Program Utama program ke-2



```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas4.hewan2.exe
Masukan Gen Induk Pertama      = X
Sifat Gen                      = merah
Masukan Gen Induk Kedua        = x
Sifat Gen                      = putih
```

Gambar 2.6 keadaan pilih=1 program ke-2

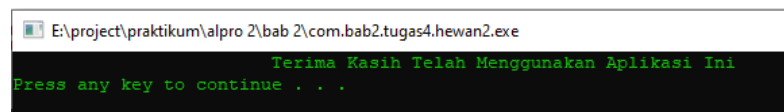


```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas4.hewan2.exe
Gen induk ke satu      = X (merah)
Gen induk ke dua      = x (putih)

Kemungkinan Warna Keturunan 100% = Xx (merah muda)

Press any key to continue . . .
```

Gambar 2.7 keadaan pilih=2 program ke-2



```
E:\project\praktikum\alpro 2\bab 2\com.bab2.tugas4.hewan2.exe
Terima Kasih Telah Menggunakan Aplikasi Ini
Press any key to continue . . .
```

Gambar 2.8 keadaan pilih=2 program ke-2