

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



NAMA : REMEMBER SITOMPUL
NIM : 193020503022
KELAS : A
MODUL : PEWARISAN

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

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



Nama : Remember Sitompul
NIM : 193020503022
Kelas : A
Modul : II (PEWARISAN)

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	7
BAB II Pembahasan	60	48
BAB III Kesimpulan	20	10
Daftar Pustaka	5	3
Lampiran	5	3
Jumlah	100	71

Penilai
Asisten Praktikum

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

A. Tujuan

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

B. Landasan Teori

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

Selain melakukan kategorisasi terhadap objek yang memiliki sekumpulan atribut dan perilaku yang sama, manusia sering melakukan pengelompokan terhadap objek yang memiliki kesamaan atas beberapa (**tidak semua**) atribut/perilaku. Contoh : pengelompokan atas kendaraan bermotor, kemudian menggrupkannya berdasarkan suatu tipe atau jenis (mobil, truk, sepeda motor, dll.). Setiap subkategori merupakan kelas atas objek-objek yang serupa.

Ada beberapa karakteristik yang di-share oleh semua kelompok. Relasi antar kelas-kelas ini disebut dengan **relasi “is - a”**. dalam setiap kasus, objek yang dikelompokkan bersama dalam satu sub-kategori merupakan anggota dari kategori yang lebih umum. Contohnya adalah seperti di bawah ini.

- a) Mobil adalah (“is – a”) kendaraan bermotor.
- b) Truk adalah (“is – a”) kendaraan bermotor.
- c) 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 nama atribut yang diakses.

Memahami Konsep Inheritance (Pewarisan) Pada OOP

1. KONSEP PEWARISAN (INHERITANCE) PADA OOP

Sebagai Informasi bahwa Ciri khas Pemrograman Berorientasi Objek ada tiga hal yaitu:

- a. Enkaptulasi (Pembungkusan)
- b. Inheritance (Pewarisan)
- c. Polymorphism (Perbedaan Bentuk)

A. PENGERTIAN PEWARISAN (INHERITANCE)

Konsep inheritance ini mengadopsi dunia riil dimana suatu entitas/obyek dapat mempunyai entitas/obyek turunan. Dengan konsep inheritance, sebuah Class dapat mempunyai class turunan. Suatu class yang mempunyai class turunan dinamakan parent class atau base class. Sedangkan class turunan itu sendiri seringkali disebut subclass atau child class. Suatu subclass dapat mewarisi apa-apa yang dipunyai oleh parent class. Karena suatu subclass dapat mewarisi apa apa yang dipunyai oleh parent class-nya, maka member dari suatu subclass adalah terdiri dari apa-apa yang ia punyai dan juga apa-apa yang ia warisi dari class parent-nya. Kesimpulannya, boleh dikatakan bahwa suatu subclass adalah tidak lain hanya Memperluas (extend) parent class-nya.

Dari hirarki diatas dapat dilihat bahwa, semakin kebawah,

class akan semakin bersifat spesifik. Class mamalia memiliki seluruh sifat yang dimiliki oleh binatang, demikian halnya juga macan , kucing, Paus dan Monyet memiliki seluruh sifat yang diturunkan dari class mamalia. Dengan konsep ini, karakteristik yang dimiliki oleh class binatang cukup didefinisikan dalam class binatang saja. Class mamalia tidak perlu mendefinisikan ulang apa yang telah dimiliki oleh class binatang, karena sebagai class turunannya, ia akan mendapatkan karakteristik dari class binatang secara otomatis. Demikian juga dengan class macan, kucing, Paus dan monyet, hanya perlu mendefinisikan karakteristik yang spesifik dimiliki oleh classnya masing-masing. Dengan memanfaatkan konsep pewarisan ini dalam pemrograman, maka hanya perlu mendefinisikan karakteristik yang lebih umum akan didapatkan dari class darimana ia diturunkan.

B. DEKLARASI PEWARISAN (INHERITANCE)

Deklarasi yang digunakan adalah dengan menambahkan kata kunci `extends` setelah deklarasi nama class, kemudian diikuti dengan nama parent class-nya. Kata kunci `extends` tersebut memberitahu kompiler Java bahwa kita ingin melakukan perluasan class.

```
Public class B extends A {...}.
```

Pada saat dikompilasi, Kompiler Java akan membaca sebagai subclass dari class object.

```
Public class A extends Object {...}
```

C. KAPAN KITA MENERAPKAN INHERITANCE ?

Kita baru perlu menerapkan inheritance pada saat kita jumpai ada suatu class yang dapat diperluas dari class lain. Misal terdapat class Pegawai

```
public class Pegawai {public String nama;public double gaji;}Misal terdapat class Manajerp
```

```
public class Manajer {public String nama;public double gaji;public String departemen;} Dari 2 buah class diatas, kita lihat class Manajer mempunyai data member yang identik sama dengan class Pegawai, hanya saja ada tambahan data member departemen.
```

Sebenarnya yang terjadi disana adalah class Manajer merupakan perluasan dari class pegawai dengan tambahan data member departemen. Disini perlu memakai konsep inheritance, sehingga class Manajer dapat kita tuliskan Seperti berikut :

```
public class Manajer extends Pegawai {public String departemen;}
```

D. KEUNTUNGAN INHERITANCE

- a. Subclass menyediakan state/behaviour yang spesifik yang membedakannya dengan superclass, hal ini akan memungkinkan programmer Java untuk menggunakan ulang source code dari superclass yang telah ada.
- b. Programmer Java dapat mendefinisikan superclass khusus yang bersifat generik, yang disebut abstract class, untuk mendefinisikan class dengan behaviour dan state secara umum.

E. ISTILAH INHERITANCE

Extends

Keyword ini harus kita tambahkan pada definisi class yang menjadi subclass.

Superclass

Superclass digunakan untuk menunjukkan hirarki class yang berarti class dasar dari subclass/class anak.

Subclass

Subclass adalah class anak atau turunan secara hirarki dari superclass.

Super

Keyword ini digunakan untuk memanggil konstruktor dari superclass atau menjadi variabel yang mengacu pada superclass.

Method Overriding

Pendefinisian ulang method yang sama pada subclass. Dalam inheritance, method overriding berbeda dengan method overloading. Kalau method overriding adalah mendefinisikan kembali method yang sama, baik nama method maupun signature atau parameter yang diperlukan dalam subclass, kalau method overloading adalah mendefinisikan method yang memiliki nama yang sama, tetapi dengan signature yang berbeda dalam definisi class yang sama.

PEMBAHASAN

1. Program pertama

Program untuk menghasilkan ciri-ciri hewan di bawah:

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

//Kelas Dasar "Hewan"
class Hewan
{
protected:
    char nama[15];
    char habitat[15];
    char namaHewan[15];
    char habitatHewan[15];

public:
    void infoHewan()
    {
        cout<<endl;
        cout<<"Info salah satu jenis hewan"<<endl;
    }
};

class Buaya : public Hewan
{
protected:
    char ciri_khusus[20];

public:
    //konstruktor
    Buaya(char nm[15], char *hbt, char *c_kh)
    {
        strcpy (nama, nm);
        strcpy (habitat, hbt);
        strcpy (ciri_khusus, c_kh);
    }

    void infoBuaya()
    {
        cout<<"Nama          : "<<nama<<endl;
        cout<<"Habitat        : "<<habitat<<endl;
        cout<<"Ciri Khusus   : "<<ciri_khusus<<endl;
    }
};

int main()
{
    //menciptakan objek "bya" sekaligus melakukan
    inisialisasi melalui Konstruktor dari Kelas Gajah
    Buaya bya("BUAYA", "RAWA RAWA", "hidup di darat
    dan di air");

    cout<<"INHERITANCE PADA C++"<<endl;
    cout<<"-----"<<endl;

    bya.infoHewan();
```

```
bya.infoBuaya();

    getch();
    return 0;
}
```

Pejelasan:

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

Ini adalah header file, atau bisa juga disebut library. File ini sebenarnya kumpulan fungsi-fungsi yang sering digunakan untuk pembuatan program. Fungsi #include adalah untuk memasukkan file library. Sedangkan conio.h dan iostream.h berisi library untuk memanggil fungsi yang kita butuhkan yaitu getch();

```
class Hewan
{
protected:
    char nama[15];
    char habitat[15];
    char namaHewan[15];
    char habitatHewan[15];

public:
    void infoHewan()
    {
        cout<<endl;
        cout<<"Info salah satu jenis hewan"<<endl;
    }
};
```

Ini merupakan class Induk yang nantinya akan di turukan ke kelas anak. Class ini menggunakan protected dan public supaya programnya bisa di turunkan ke kelas anak.

Pada bagian protected ini memiliki variabel antara lain:

1. char nama[15];
2. char habitat[15];
3. char namaHewan[15];
4. char habitatHewan[15];

Pada bagian public ini mengakses program:

```
void infoHewan()
{
    cout<<endl;
    cout<<"Info salah satu jenis hewan"<<endl;
```



```

    }
};

```

Yang nantinya akan menampilkan output pada inti program yaitu “Info salah satu jenis hewan”.

```

class Buaya : public Hewan
{
protected:
    char ciri_khusus[20];

public:
    //konstruktor
    Buaya(char nm[15], char *hbt, char *c_kh)
    {
        strcpy (nama, nm);
        strcpy (habitat, hbt);
        strcpy (ciri_khusus, c_kh);
    }

    void infoBuaya()
    {
        cout<<"Nama          : "<<nama<<endl;
        cout<<"Habitat       : "<<habitat<<endl;
        cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
    }
};

```

Ini merupakan turunan dari class induk (class Hewan) dengan tambahan konstruktor pada public program anak.

Pada Bagian program baris yang selanjutnya merupakan **class Buaya: public Hewan{** yang berfungsi untuk mendeklarasikan variabel dan fungsi yang terdapat pada **class Buaya**. Kemudian pada baris selanjutnya adalah **void infoBuaya()** yang tipe datanya void, digunakan untuk menampilkan/mencetak nilai dari fungsi area.

Kemudian pada mode akses protectednya, terdapat variabel radius dan l yang tipe datanya char. Kemudian diakhiri dengan tanda kurung kurawal tutup (}) untuk menandakan akhir dari program.

```

int main()
{
    //menciptakan objek "bya" sekaligus
    melakukan inisialisasi melalui Konstruktor dari
    Kelas Gajah
    Buaya bya("BUAYA", "RAWA RAWA", "hidup di
    darat dan di air");

    cout<<"INHERITANCE PADA C++"<<endl;
    cout<<"-----"<<endl;
}

```

```

        bya.infoHewan();
        bya.infoBuaya();

        getch();
        return 0;
    }

```

Pada bagian ini merupakan bagian utama program. **int main()** Pada bagian ini, dilakukan pendeklarasian/penginputan hasil dari program Buaya bya("BUAYA", "RAWA RAWA", "hidup di darat dan di air");. Yang nantinya akan menampilkan hasil output.

31.5

Kemudian pada baris selanjutnya terdapat fungsi yaitu **getch()**; yang berfungsi untuk menangkap tampilan pada layar dan program tidak akan tertutup sendiri sebelum tombol pada keyboard ditekan. Fungsi ini mirip dengan fungsi **readkey** pada pascal, fungsi ini diambil dari file **conio.h** yang merupakan header yang berguna untuk. Memanggil fungsi standar yang sudah ada pada C++. Pada bagian akhir dari fungsi **main** ini ada tertulis sebuah kode yaitu **return 0;**, yang berarti apabila program ini dieksekusi maka akan memberikan nilai kembalian 0 yang artinya program dijalankan tanpa error. inilah alasan kenapa fungsi ini menggunakan **int** (integer) sebagai nilai kembalinya.

HASIL OUTPUT PROGRAM

```

4G 21:09 0.5 KB/s 55%
← TAB ⋮
INHERITANCE PADA C++
-----
Info salah satu jenis hewan
Nama      : BUAYA
Habitat   : RAWA RAWA
Ciri Khusus : hidup di darat dan di air

```

2. Program kedua

```

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

```

```

#include <string.h>
using namespace std;

//Kelas Dasar "Hewan"
class Hewan
{
protected:
    char nama[15];
    char habitat[15];
    char namaHewan[15];
    char habitatHewan[15];

public:
    void infoHewan()
    {
        cout<<endl;
        cout<<"Info salah satu jenis
hewan"<<endl;
    }
};

class Gajah : public Hewan
{
private:
    char ciri_khusus[20];

public:
    //konstruktor
    Gajah(char nm[15], char *hbt, char *c_kh)
    {
        strcpy (nama, nm);
        strcpy (habitat, hbt);
        strcpy (ciri_khusus, c_kh);
    }

    void infoGajah()
    {
        cout<<"Nama          : "<<nama<<endl;
        cout<<"Habitat       : "<<habitat<<endl;
        cout<<"Ciri Khusus :
"<<ciri_khusus<<endl;
    }
};

int main()
{
    //menciptakan objek "gjh" sekaligus melakukan
    inisialisasi melalui Konstruktor dari Kelas Gajah
    Gajah gjh("Gajah", "Padang rumput", "Belalai dan
    Gading");

    cout<<"INHERITANCE PADA C++"<<endl;
    cout<<"-----"<<endl;

    gjh.infoHewan();
    gjh.infoGajah();

    getch();
    return 0;
}

```

Penjelasan :

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

Ini adalah header file, atau bisa juga disebut library. File ini sebenarnya kumpulan fungsi-fungsi yang sering digunakan untuk pembuatan program. Fungsi #include adalah untuk memasukkan file library. Sedangkan conio.h dan iostream.h berisi library untuk memanggil fungsi yang kita butuhkan yaitu getch();

```
class Hewan
{
protected:
    char nama[15];
    char habitat[15];
    char namaHewan[15];
    char habitatHewan[15];

public:
    void infoHewan()
    {
        cout<<endl;
        cout<<"Info salah satu jenis hewan"<<endl;
    }
};
```

Ini merupakan class Induk yang nantinya akan di turunkan ke kelas anak. Class ini menggunakan protected dan public supaya programnya bisa di turunkan ke kelas anak.

Pada bagian protected ini memiliki variabel antara lain:

1. char nama[15];
2. char habitat[15];
3. char namaHewan[15];
4. char habitatHewan[15];

Pada bagian public ini mengakses program:

```
void infoHewan()
{
    cout<<endl;
    cout<<"Info salah satu jenis hewan"<<endl;
}
};
```

Yang nantinya akan menampilkan output pada inti program yaitu "Info salah satu jenis hewan".

```
class Gajah : public Hewan
{
```

```

private:
    char ciri_khusus[20];

public:
    //konstruktor
    Gajah(char nm[15], char *hbt, char *c_kh)
    {
        strcpy (nama, nm);
        strcpy (habitat, hbt);
        strcpy (ciri_khusus, c_kh);
    }

    void infoGajah()
    {
        cout<<"Nama          : "<<nama<<endl;
        cout<<"Habitat       : "<<habitat<<endl;
        cout<<"Ciri Khusus : "<<ciri_khusus<<endl;
    }
};

```

Ini merupakan turunan dari class induk (class Hewan) dengan tambahan konstruktor pada public program anak.

Pada Bagian program baris yang selanjutnya merupakan **class Gajah : public Hewan{** yang berfungsi untuk mendeklarasikan variabel dan fungsi yang terdapat pada **class Gajah**. Kemudian pada baris selanjutnya adalah **void infoGajah()** yang tipe datanya void, digunakan untuk menampilkan/mencetak nilai dari fungsi area.

Kemudian pada mode akses protectednya, terdapat variabel radius dan l yang tipe datanya char. Kemudian diakhiri dengan tanda kurung kurawal tutup (}) untuk menandakan akhir dari program.

```

int main()
{
    //menciptakan objek "gjh" sekaligus melakukan inisialisasi
    melalui Konstruktor dari Kelas Gajah
    Gajah gjh("Gajah", "Padang rumput", "Belalai dan Gading");

    cout<<"INHERITANCE PADA C++"<<endl;
    cout<<"-----"<<endl;

    gjh.infoHewan();
    gjh.infoGajah();

    getch();
    return 0;
}

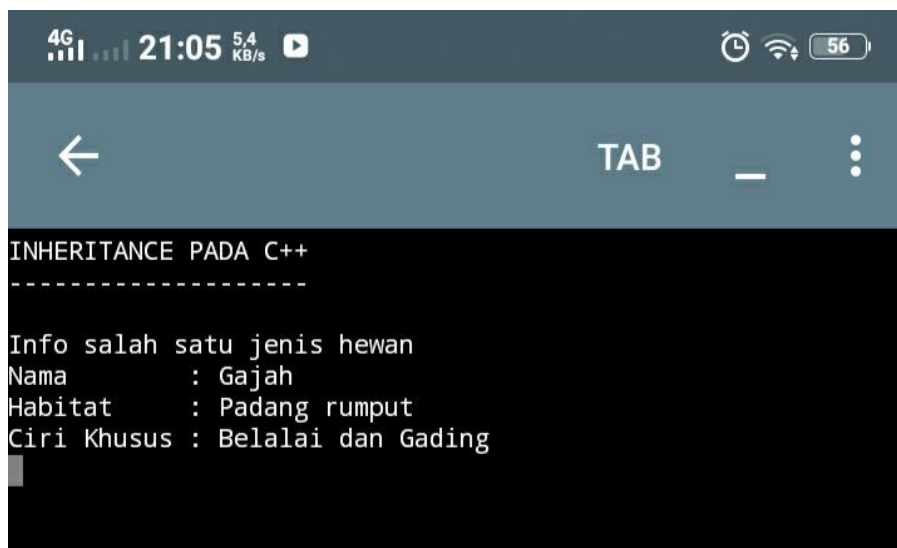
```

```
}
```

Pada bagian ini merupakan bagian utama program. **int main()** Pada bagian ini, dilakukan pendeklarasian/penginputan hasil dari program Gajah gjh ("Gajah", "Padang rumput", "Belalai dan Gading");. Yang nantinya akan menampilkan hasil output.

Kemudian pada baris selanjutnya terdapat fungsi yaitu **getch()**; yang berfungsi untuk menangkap tampilan pada layar dan program tidak akan tertutup sendiri sebelum tombol pada keyboard ditekan. Fungsi ini mirip dengan fungsi `readkey` pada pascal, fungsi ini diambil dari file `conio.h` yang merupakan header yang berguna untuk. Memanggil fungsi standar yang sudah ada pada C++. Pada bagian akhir dari fungsi `main` ini ada tertulis sebuah kode yaitu **return 0;**, yang berarti apabila program ini dieksekusi maka akan memberikan nilai kembalian 0 yang artinya program dijalankan tanpa error. Inilah alasan kenapa fungsi ini menggunakan `int` (integer) sebagai nilai kembalinya.

HASIL OUPUT PROGRAM

A screenshot of a mobile application interface. At the top, there is a status bar with '4G' signal, time '21:05', and battery level '56'. Below the status bar is a dark blue header with a back arrow on the left, the word 'TAB' in the center, and a menu icon on the right. The main content area is black with white text. It starts with 'INHERITANCE PADA C++' followed by a dashed line. Then, it says 'Info salah satu jenis hewan' followed by three lines of text: 'Nama : Gajah', 'Habitat : Padang rumput', and 'Ciri Khusus : Belalai dan Gading'. There is a small grey square at the bottom left of the text area.

BAB III

KESIMPULAN

Pewarisan merupakan sebuah Kelas yang menurunkan sifat-sifat class disebut kelas dasar (*base class*), sedangkan yang kelas baru yang mewarisi sifat kelas dasar disebut kelas turunan (*derived class*).

Float merupakan tipe data yang menyatakan bilangan pecahan.

fungsi *strcpy* adalah untuk penyalinan(copy) string.

KONSTRUKTOR adalah fungsi anggota yang mempunyai nama sama dengan nama kelasnya, sebagai fungsi yang tidak mempunyai nilai balik, harus diletakkan pada bagian public, dijalankan secara otomatis saat objek diciptakan , dapat memiliki argument dan argument dengan nilai default, dapat diletakkan overloading . Kegunaannya : mengalokasikan ruang bagi sebuah objek , memberikan nilai awal terhadap anggota data suatu objek , membentuk tugas- tugas umum lainnya , melakukan tugasnya pada saat objek diciptakan . (//) merupakan sebuah komentar

DAFTAR PUSTAKA

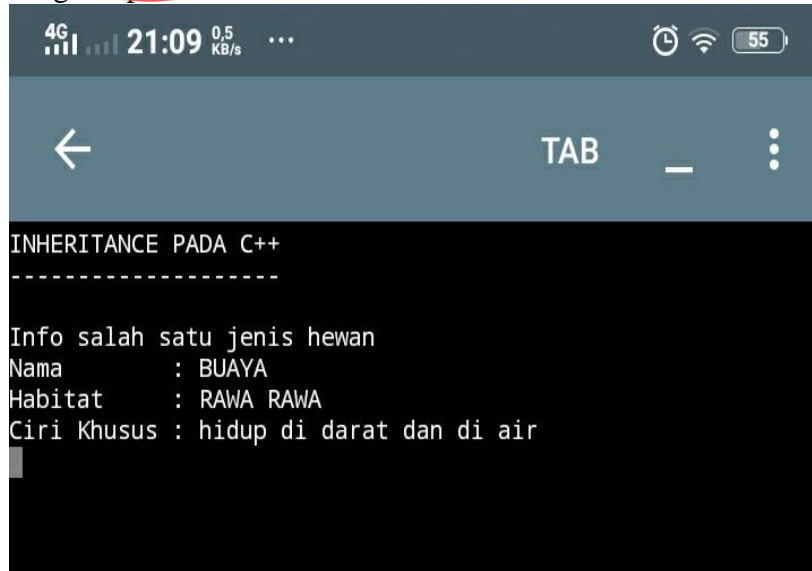
Modul Praktikum Algoritma dan Pemrograman II, Jurusan Teknik Informatika
Universitas Palangkaraya, 2020.

<http://mungjava.blogspot.com/2009/05/memahami-konsep-inheritance>
pewarisan.html (diakses pada hari rabu tanggal 15 april 2020 pukul 13.35
WIB)

[http://www.findamo.com/search.html?q=pewarisan+pada+pemrogaman+c++
®ion=id&lang=en](http://www.findamo.com/search.html?q=pewarisan+pada+pemrogaman+c++®ion=id&lang=en) (diakses pada hari rabu tanggal 15 april 2020 pukul
14:00 WIB)

LAMPIRAN

1. Program pertama



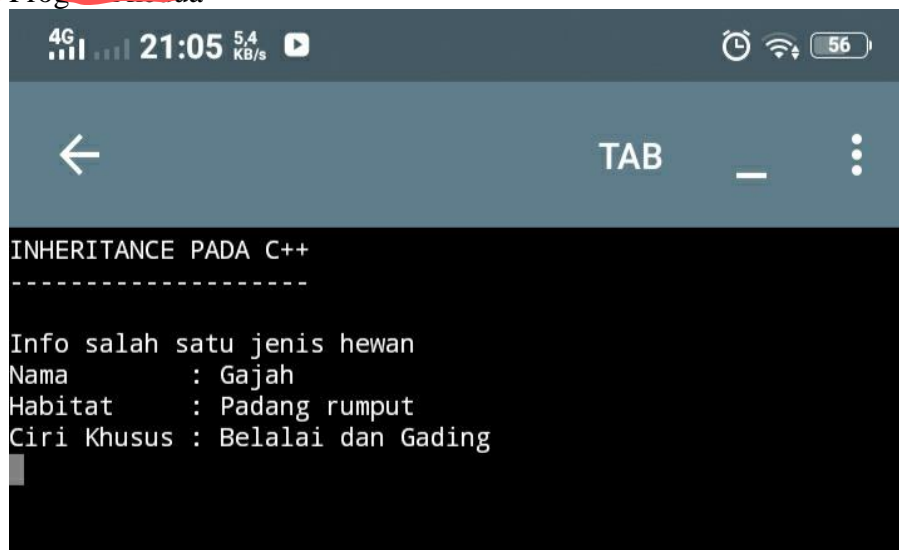
The screenshot shows a mobile application interface with a dark blue header bar containing a back arrow, the word 'TAB', and a menu icon. The main content area is black with white text. The text reads: 'INHERITANCE PADA C++', followed by a dashed line separator, then 'Info salah satu jenis hewan'. Below this, three lines of data are displayed: 'Nama : BUAYA', 'Habitat : RAWA RAWA', and 'Ciri Khusus : hidup di darat dan di air'.

```
INHERITANCE PADA C++
-----

Info salah satu jenis hewan
Nama      : BUAYA
Habitat   : RAWA RAWA
Ciri Khusus : hidup di darat dan di air
```

Gambar 1.1

2. Program kedua



The screenshot shows a mobile application interface with a dark blue header bar containing a back arrow, the word 'TAB', and a menu icon. The main content area is black with white text. The text reads: 'INHERITANCE PADA C++', followed by a dashed line separator, then 'Info salah satu jenis hewan'. Below this, three lines of data are displayed: 'Nama : Gajah', 'Habitat : Padang rumput', and 'Ciri Khusus : Belalai dan Gading'.

```
INHERITANCE PADA C++
-----

Info salah satu jenis hewan
Nama      : Gajah
Habitat   : Padang rumput
Ciri Khusus : Belalai dan Gading
```

Gambar 2.1

