

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : HUSSAIN SYACH NURALAM**  
**NIM : 193020503029**  
**KELAS : A**  
**MODUL : III (POLIMORFISME)**

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

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



Nama : Hussain Syach Nuralam  
NIM : 193020503029  
Kelas : A  
Modul : III (Polimorfisme)

| Komposisi                      | MAX | Nilai |
|--------------------------------|-----|-------|
| BAB I Tujuan dan LandasanTeori | 10  | 9     |
| BAB II Pembahasan              | 60  | 50    |
| BAB III Kesimpulan             | 20  | 13    |
| Daftar Pustaka                 | 5   | 5     |
| Lampiran                       | 5   | 5     |
| Jumlah                         | 100 |       |

Penilai  
Asisten Praktikum

Diana

# BAB I

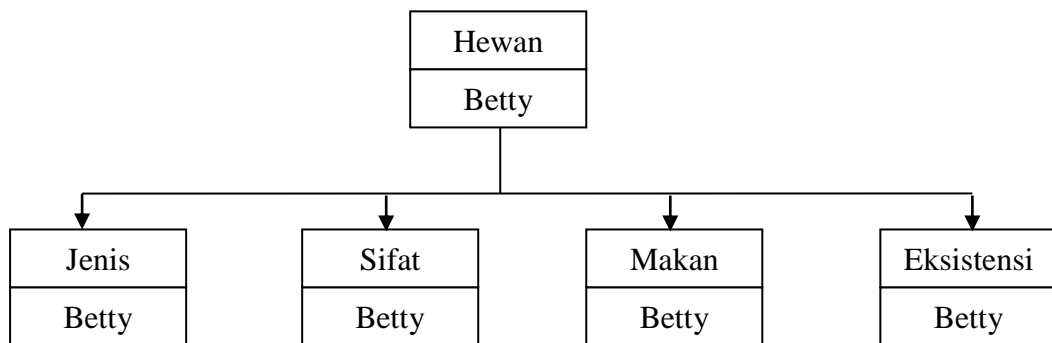
## TUJUAN DAN LANDASAN TEORI

### 1.1 Tujuan

Tujuan dari dilaksanakannya praktikum Algoritma dan Pemrograman 2, modul III, Polimorfisme adalah agar mahasiswa mampu membuat polimorfisme menggunakan media Dev C++.

### 1.2 Landasan Teori

Polimorfisme Memiliki arti “banyak bentuk”, melakukan hal yang sama untuk berbagai data yang berbeda, mengirimkan pesan yang sama ke berbagai objek yang berbeda karena tiap objek memberi respons dengan cara yang berbeda. Berikut ini merupakan contoh polimorfisme.



Gambar 1.1 Bagan Polimorfisme

Terdapat dua tipe polimorfisme, yaitu saat pengkompilasian (compile time) dan saat runtime. Polimorfisme saat pengkompilasian (compile time) dalam hal ini adalah overloading pada fungsi dan overloading pada operator. Pada polimorfisme saat runtime, fungsi pada kelas dasar di deklarasikan untuk umum, selanjutnya secara spesifiknya akan diterangkan kembali pada kelas turunan.

Analoginya, semua binatang memiliki suara maka pada kelas dasar diciptakan variabel dan fungsi untuk menampilkan suara. Namun seperti yang kita ketahui setiap jenis binatang memiliki suara yang berbeda, sehingga pada kelas turunan ditulis kembali suara yang khas dari jenis binatang tersebut.

Contoh lain, seorang raja (kelas dasar) mempunyai uang yang akan diberikan (diwariskan/diturunkan) ke anak kecil (kelas turunan) dan orang tua (kelas turunan). Uang beserta jumlahnya sama, namun perlakukannya dapat berbeda-beda (memiliki banyak bentuk). Oleh anak kecil uang tersebut digunakan untuk membeli mainan dan atau semua yang diinginkan. Oleh orang tua uang tersebut digunakan untuk membeli semua yang dibutuhkan termasuk kebutuhan anaknya.

Polimorfisme memiliki syarat-syarat sebagai berikut:

1. Ada hirarki pewarisan.
2. Kelas dalam hirarki pewarisan harus memiliki fungsi virtual (virtual method) dengan signature yang sama.
3. Menggunakan pointer atau rujukan ke kelas induk. Pointer digunakan untuk memanggil fungsi virtual.

Polimorfisme dapat diimplementasikan dengan menggunakan dasar function overriding (melakukan redefinisi suatu fungsi di kelas anak, fungsi yang di override memiliki signature sama, signature sama : tipe balik, nama fungsi, parameter sama) dan pewarisan.

Suatu kelas disebut abstrak apabila memiliki minimal satu fungsi abstrak. Fungsi abstrak merupakan fungsi yang tidak memiliki definisi (hanya deklarasi fungsi) atau menggunakan fungsi virtual (pure virtual).



```
virtual balikan namaFungsi (parameter) = 0
```

Gambar 1.2 Bentuk Fungsi Virtual

Konstruksi polimorfisme dalam pemrograman berorientasi obyek memungkinkan untuk mengadakan ikatan dinamis (juga disebut ikatan tunda, atau ikatan akhir). Apabila fungsi-fungsi dari suatu kelas dasar didefinisikan ulang atau ditindih pada kelas turunan, maka obyek-obyek yang dihasilkan hirarki kelas berupa obyek polimorfik. Polimorfik artinya mempunyai banyak bentuk atau punya kemampuan untuk mendefinisikan banyak bentuk.

Polimorfisme merupakan suatu konsep yang menyatakan bahwa sesuatu yang sama dapat memiliki berbagai bentuk dan perilaku yang berbeda. Dalam hal ini

polimorfisme merupakan suatu sifat menyandarkan pada kesamaan nama dalam program. Pengenal data, instans, dan metode, bahkan nama fungsi dapat dibuat dengan nama yang sama untuk kegunaan yang berbeda.

```
class induk
{
    //member
};
class anak: public induk
{
    //member
};
int main(){
    induk *pointer_anak;
    anak objek_anak;
    pointer_anak = &objek_anak
}
```

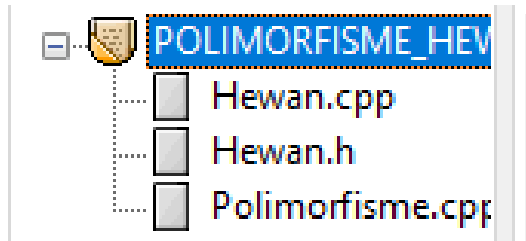
Gambar 1.3 Sintaks Sederhana Polimorfisme

Dari permisalan di atas, kita ketahui bahwa terdapat kelas anak yang menginduk pada kelas induk. Kemudian pada fungsi main(), terdapat deklarasi pointer bertipe kelas induk yang digunakan sebagai penunjuk objek kelas anak.

## BAB II

### PEMBAHASAN

#### 2.1 Program Polimorfisme Menggunakan Project



Gambar 2.1 Project pada Polimorfisme

Project pada program polimorfisme sangatlah berfungsi. Ia bisa memuat beberapa file coding yang nantinya akan menghasilkan 1 program project. Berikut adalah beberapa program yang terdapat pada project program polimorfisme hewan.

```
#ifndef HEWAN_H
#define HEWAN_H

class Hewan {
public :
    virtual void Hello();
};

class Jenis : public Hewan {
public :
    virtual void Hello();
};

class Sifat : public Hewan {
public :
    virtual void Hello();
};
```

Gambar 2.2 Coding Hewan.h

```

class Makan : public Hewan {
    public :
        virtual void Hello();
};
class Eksistensi : public Hewan {
    public :
        virtual void Hello();
};
#endif

```

Gambar 2.3 Coding Hewan.h

Pada file Hewan.h yang merupakan sub program, terdapat fungsi virtual di dalamnya. Fungsi virtual atau virtual function adalah sebuah member function yang didirikan di dalam base class dan overridden oleh derived class. Ketika anda menunjuk ke suatu object yang dibuat dari derived class dengan menggunakan sebuah object pointer yang dibuat menggunakan base class, anda dapat memanggil sebuah function yang dibuat sebagai virtual function. setiap pemanggilan function, akan mengeksekusi masing-masing function berdasarkan yang dimiliki oleh class yang ditunjuk.

```

#include "Hewan.h"
#include <iostream>

using namespace std;

void Hewan::Hello() {
    cout << "Betty adalah seekor ayam" << endl;
}

```

Gambar 2.4 Coding Hewan.cpp

```

void Jenis::Hello() {
    cout <<"Jenis ayam ini adalah Ayam Silkie"<< endl;
}
void Sifat::Hello() {
    cout <<"Ayam ini memiliki sifat yang tenang dan jinak"<< endl;
}
void Makan::Hello() {
    cout <<"Hewan ini termasuk Omnivora"<< endl;
}
void Eksistensi::Hello() {
    cout <<"Termasuk hewan langka dan unik"<< endl;
}

```

Gambar 2.5 Coding Hewan.cpp

Pada file Hewan.cpp, coding diatas memiliki fungsi sebagai sub program. Sub program ini memiliki fungsi void. Fungsi tersebut nantinya berguna untuk menampilkan data yang diinginkan pada output program utama.

```

#include <stdlib.h>
#include "Hewan.h"
#include <iostream>

using namespace std;

int main() {
    Hewan* h;

```

Gambar 2.6 Coding Polimorfisme.cpp



```

int pilihan;

do {
    ulang:
    system("cls");
    cout <<"-----SELAMAT DATANG-----\n"
    <<"-----PROFIL BETTY PELIHARAAN HUSIN-----\n"
    <<" 1: Hewan \n 2: Jenis \n 3: Sifat \n 4: Makan \n 5: Eksistensi \n"
    <<"\nMasukan Pilihan Anda : ";
    cin >> pilihan;
}
while ( pilihan <0 || pilihan >5 );

switch ( pilihan ) {
    case 1:
        h = new Hewan;
        break;

    case 2:
        h = new Jenis;
        break;

    case 3:
        h = new Sifat;
        break;

```

Gambar 2.7 Coding Polimorfisme.cpp

```

        case 4:
            h = new Makan;
            break;

        case 5:
            h = new Eksistensi;
            break;
    }
    h->Hello();
    delete h;

    return 0;
}

```

Gambar 2.8 Coding Polimorfisme.cpp

Coding pada gambar 2.6 sampai 2.8 merupakan main program pada polimorfisme hewan. Coding-coding diatas memegang peranan utama pada polimorfisme hewan yang mana nantinya akan menggunakan fungsi yang dapat dihubungkan pada Hewan.h dan Hewan.cpp.

Berikut ini adalah beberapa penjelasan tambahan terkait coding-coding yang ada di atas.

1) **iostream.h (iostream)** : Merupakan singkatan dari input outout stream header yang digunakan sebagai standar input output operasi yang digunakan oleh bahasa C++. Fungsi-fungsi yang ada didalam iostream.h antara lain sebagai berikut :

1.1) **cout** : Merupakan fungsi keluaran pada C++ yang menampilkan data dengan tipe data apapun kelayar.

1.2) **cin** : Merupakan fungsi masukan pada C++ yang bisa memasukan data berupa numerik dan karakter.

1.3) **endl** : Merupakan suatu fungsi yang manipulator yang digunakan untuk melakukan perintah Newline atau pindah baris

2) **#include "Hewan.h"**

Preprocessor **#include** berfungsi untuk memasukkan atau menyertakan file-file header ke dalam program yang akan dibuat. Pertama kali kompiler akan mencari file header yang disebutkan pada direktori yang sedang aktif dan apabila tidak ditemukan akan mencari pada direktori dimana file header tersebut berada.

3) **#define**

Digunakan untuk mendefinisikan suatu nilai tertentu kepada suatu nama konstanta.

4) **#ifdef - #ifndef**

Preprocessor **#ifdef** dan **#ifndef** memberikan bagian dari program yang akan dikompilasi, hal ini dapat dilakukan jika sudah konstanta didefinisikan pada bagian **#define**, hal ini merupakan parameter yang khusus yang harus terdefinisi.

5) **using namespace std** : File header ini terdiri dari 3 statements yaitu “using”, “namespace” dan “std”. Fungsi-fungsi penggunaan using namespace std adalah sebagai berikut :

5.1) Using adalah deklarasi arahan / panggilan / pemberitahuan kepada kompiler untuk penggunaan deklarasi namespace dan anggota namespace

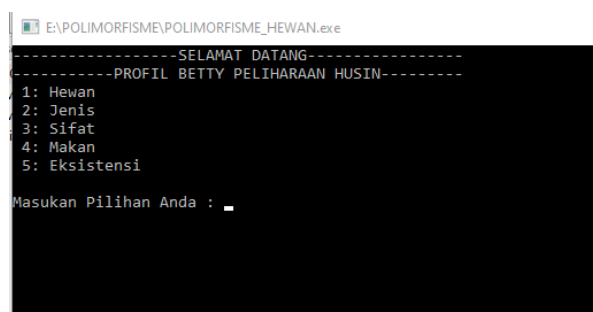
5.2) Namespace adalah penyedia metode untuk mencegah konflik nama dalam proyek-proyek besar. Simbol pendeklarasian di dalam blok namespace ditempatkan di lingkup bernama yang akan mencegah mereka pada keliruan pengenalan pada scope yang lainnya. Jika pendeklarasiannya seperti diatas, itu berfungsi untuk memanggil namespace yang telah dibuat.

5.3) Std adalah nama namespace tersebut yang sudah tersedia dan tidak perlu dibuat ulang menggunakan fungsi namespace, cukup dipanggil. Std merupakan wadah urutan yang merangkum ukuran data dan array

dinamis. Di dalam std terdapat halnya seperti cout, cin, endl dan lain-lain.

- 6) **Void** : Void adalah sebuah fungsi ( function ) yang ada dalam sebuah bahasa pemrograman C, entah itu C++ atau C#. Fungsi ini juga disebut sebagai prosedur ( procedure ). Fungsi ini tidak mengembalikan nilai keluaran ( return output ) yang didapat dari hasil proses tersebut, ini kenapa fungsi ini disebut void, secara harfiah berarti kosong.
- 7) **int main()** : Int main() artinya main program mengembalikan nilai int secara default, int main() akan mengembalikan nilai 0, dan fungsi main() tidak memiliki bagan deklarasi lokal, dan hanya memiliki sebuah pernyataan yang dapat dieksekusi, berupa fungsi output printf().
- 8) **Switch Case** : Pernyataan switch adalah pernyataan penyeleksian untuk memanipulasi jalanya aliran program berdasarkan conditional expression, Mengendalikan aliran program agar dapat berjalan secara fleksibel atas keadaan program dan keinginan pengguna. Pernyataan ini akan memilih pilihan yang sesuai dengan ekspresi kondisi.  
Pada umumnya pernyataan switch tidak jauh berbeda seperti pernyataan penyeleksian IF, yang berbeda adalah; jika pada if, if mengevaluasi kebenaran dari sebuah ekspresi (conditional expression) seperti operasi logika, relasi dan perbandingan yang menghasilkan bilangan boolean 1 (benar) atau 0 (salah).

Dari coding-coding diatas, dihasilkan output utama. Berikut adalah outputnya.



```
E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi
Masukan Pilihan Anda : _
```

Gambar 2.9 Output Main Program

```
E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi

Masukan Pilihan Anda : 1
Betty adalah seekor ayam

-----
Process exited after 137.7 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.10 Output Menu 1

```
E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi

Masukan Pilihan Anda : 2
Jenis ayam ini adalah Ayam Silkie

-----
Process exited after 101.4 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.10 Output Menu 2

```
E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi

Masukan Pilihan Anda : 3
Ayam ini memiliki sifat yang tenang dan jinak

-----
Process exited after 7.099 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.11 Output Menu 3

```
E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi

Masukan Pilihan Anda : 4
Hewan ini termasuk Omnivora

-----
Process exited after 6.883 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.12 Output Menu 4

```
E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi

Masukan Pilihan Anda : 5
Termasuk hewan langka dan unik

-----
Process exited after 3.611 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.13 Output Menu 5

### **BAB III**

### **KESIMPULAN**

Polimorfisme merupakan suatu konsep yang menyatakan bahwa sesuatu yang sama dapat memiliki berbagai bentuk dan perilaku yang berbeda. Dalam hal ini polimorfisme merupakan suatu sifat menyandarkan pada kesamaan nama dalam program. Pengenal data, instans, dan metode, bahkan nama fungsi dapat dibuat dengan nama yang sama untuk kegunaan yang berbeda.

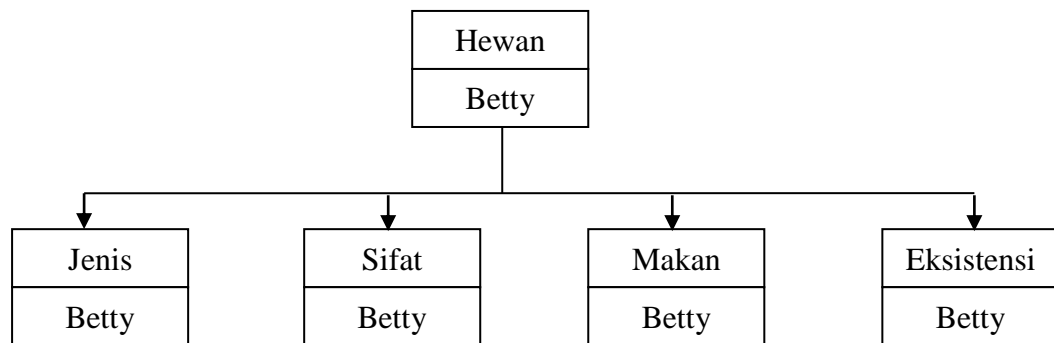
Dalam OOP, konsep polimorfisme memungkinkan digunakannya suatu interface yang sama untuk memerintah suatu objek agar melakukan suatu aksi/tindakan yang mungkin secara prinsip sama tetapi secara proses berbeda. Sehingga, dapat dikatakan bahwa polimorfisme adalah kemampuan 2 buah object yang berbeda untuk merespon pesan permintaan yang sama dalam suatu cara yang unik.

## DAFTAR PUSTAKA

- Dosen Teknik Informatika. *Algoritma dan Pemrograman II*. 2020. Modul Praktikum Algoritma dan Pemrograman II. Universitas Palangka Raya. Fakultas Teknik. Jurusan Teknik Informatika.
- Fajar. *Penjelasan Virtual Function untuk Polymorphism*. 2019. <https://www.belajarcpp.com/tutorial/cpp/virtual-polymorphism/> (Diakses pada : Rabu, 22 April 2020. Pukul 18.30 WIB)
- Fatturahman, Rizky. *Kegunaan Using namespace std*. 2017. <http://rzkyiff.blogspot.com/2017/03/kegunaan-using-namespace-std.html> (Diakses pada : Rabu, 22 April 2020. Pukul 19.00 WIB)
- Pascal, Dhika. *Macam-macam file header dan fungsinya di bahasa c++*. 2014. [http://pascaldhika.blogspot.com/2014/03/macam-macam-file-header-dan-fungsinya\\_22.html](http://pascaldhika.blogspot.com/2014/03/macam-macam-file-header-dan-fungsinya_22.html) (Diakses pada : Rabu, 22 April 2020. Pukul 20.30 WIB)
- Pratama, Ari. *Apa Itu Void Dan Ciri-Cirinya*. 2015. <https://aripratama.com/apa-itu-void-dan-ciri-cirinya/> (Diakses pada : Rabu, 22 April 2020. Pukul 21.00 WIB)
- Rizki, Aditya. *Tutorial Pemrograman Berorientasi Obyek dengan C++ : Polimorfisme (Studi Kasus)*. 2011. <https://adityarizki.net/tutorial-pemrograman-berorientasi-obyek-dengan-c-polimorfisme-studi-kasus/> (Diakses pada : Rabu, 22 April 2020. Pukul 22.30 WIB)
- Santoso, Rachmat. *Polimorfisme C++*. 2017. <http://www.nblognlife.com/2017/07/polimorfisme-pada-c.html> (Diakses pada : Rabu, 22 April 2020. Pukul 23.00 WIB)
- Sinauarduino. *Polymorphism dan Virtual Function Pada Classes C++*. 2016. <https://www.sinauarduino.com/artikel/polymorphism-dan-virtual-function-pada-classes-cpp/> (Diakses pada : Rabu, 22 April 2020. Pukul 23.30 WIB)



## LAMPIRAN



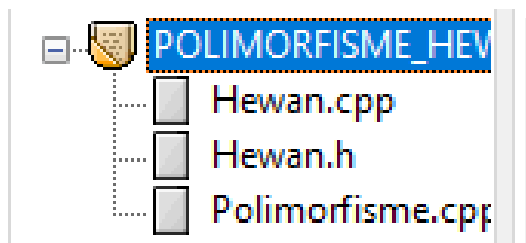
Gambar 1.1 Bagan Polimorfisme

`virtual balik namaFungsi (parameter) = 0`

Gambar 1.2 Bentuk Fungsi Virtual

```
class induk
{
    //member
};
class anak: public induk
{
    //member
};
int main(){
    induk *pointer_anak;
    anak objek_anak;
    pointer_anak = &objek_anak
}
```

Gambar 1.3 Sintaks Sederhana Polimorfisme



Gambar 2.1 Project pada Polimorfisme

```
#ifndef HEWAN_H
#define HEWAN_H

class Hewan {
    public :
        virtual void Hello();
};

class Jenis : public Hewan {
    public :
        virtual void Hello();
};

class Sifat : public Hewan {
    public :
        virtual void Hello();
};
```

Gambar 2.2 Coding Hewan.h

```
class Makan : public Hewan {
    public :
        virtual void Hello();
};

class Eksistensi : public Hewan {
    public :
        virtual void Hello();
};

#endif
```

Gambar 2.3 Coding Hewan.h

```
#include "Hewan.h"
#include <iostream>

using namespace std;

void Hewan::Hello() {
    cout <<"Betty adalah seekor ayam"<< endl;
}
```

Gambar 2.4 Coding Hewan.cpp

```
void Jenis::Hello() {
    cout <<"Jenis ayam ini adalah Ayam Silkie"<< endl;
}
void Sifat::Hello() {
    cout <<"Ayam ini memiliki sifat yang tenang dan jinak"<< endl;
}
void Makan::Hello() {
    cout <<"Hewan ini termasuk Omnivora"<< endl;
}
void Eksistensi::Hello() {
    cout <<"Termasuk hewan langka dan unik"<< endl;
}
```

Gambar 2.5 Coding Hewan.cpp

```
#include <stdlib.h>

#include "Hewan.h"

#include <iostream>

using namespace std;

int main() {
    Hewan* h;
```

Gambar 2.6 Coding Polimorfisme.cpp

```

int pilihan;

do {
    ulang:
    system("cls");
    cout <<"-----SELAMAT DATANG-----\n"
    <<"-----PROFIL BETTY PELIHARAAN HUSIN-----\n"
    <<" 1: Hewan \n 2: Jenis \n 3: Sifat \n 4: Makan \n 5: Eksistensi \n"
    <<"\nMasukan Pilihan Anda : ";
    cin >> pilihan;
}
while ( pilihan <0 || pilihan >5 );

switch ( pilihan ) {
    case 1:
        h = new Hewan;
        break;

    case 2:
        h = new Jenis;
        break;

    case 3:
        h = new Sifat;
        break;

```

Gambar 2.7 Coding Polimorfisme.cpp

```


        case 4:
            h = new Makan;
            break;

        case 5:
            h = new Eksistensi;
            break;
    }
    h->Hello();
    delete h;

    return 0;
}

```

Gambar 2.8 Coding Polimorfisme.cpp



```

E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi
Masukan Pilihan Anda : 

```

Gambar 2.9 Output Main Program

```
E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi

Masukan Pilihan Anda : 1
Betty adalah seekor ayam

-----
Process exited after 137.7 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.10 Output Menu 1

```
E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi

Masukan Pilihan Anda : 2
Jenis ayam ini adalah Ayam Silkie

-----
Process exited after 101.4 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.10 Output Menu 2

```
E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi

Masukan Pilihan Anda : 3
Ayam ini memiliki sifat yang tenang dan jinak

-----
Process exited after 7.099 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.11 Output Menu 3

```
E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi

Masukan Pilihan Anda : 4
Hewan ini termasuk Omnivora

-----
Process exited after 6.883 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.12 Output Menu 4

```
E:\POLIMORFISME\POLIMORFISME_HEWAN.exe
-----SELAMAT DATANG-----
-----PROFIL BETTY PELIHARAAN HUSIN-----
1: Hewan
2: Jenis
3: Sifat
4: Makan
5: Eksistensi

Masukan Pilihan Anda : 5
Termasuk hewan langka dan unik

-----
Process exited after 3.611 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.13 Output Menu 5