

**LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : HUSSAIN SYACH NURALAM
NIM : 193020503029
KELAS : A
MODUL : I (DASAR PEMROGRAMAN
BERORIENTASI OBJEK**

**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 : I (Dasar Pemrograman Berorientasi Objek)

Komposisi	MAX	Nilai
BAB I Tujuan dan LandasanTeori	10	7
BAB II Pembahasan	60	53
BAB III Kesimpulan	20	18
Daftar Pustaka	5	4
Lampiran	5	5
Jumlah	100	93

Penilai
AsistenPraktikum

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

31 enter

1.1 Tujuan

Tujuan dari dilaksanakannya praktikum Algoritma dan Pemrograman 2, modul I, Dasar Pemrograman Berorientasi Objek adalah sebagai berikut :

- 1) Memahami dasar-dasar pemrograman berorientasi objek.
- 2) Memahami enkapsulasi.
- 3) Membuat kelas dan objek.

1.2 Landasan Teori

Perbedaan pemrograman tradisional dan berorientasi objek adalah pada cara menyelesaikan suatu permasalahan. Pada pemrograman tradisional dalam memecahkan suatu masalah, masalah akan dibagi menjadi fungsi-fungsi yang lebih kecil, sedangkan pada pemrograman berorientasi objek (PBO) setiap masalah diselesaikan dengan cara dibagi ke dalam objek-objek. Tujuan utama dari pemrograman C++ ialah untuk menambahkan orientasi objek pada bahasa pemrograman C dan kelas-kelas yang dijadikan sebagai fitur dari C++ yang mendukung pemrograman berorientasi objek (PBO) dan sering juga dikenal sebagai user-defined type.

Object Oriented Programming atau disingkat menjadi OOP adalah paradigma pemrograman dalam melakukan pemrograman yang berorientasi kepada object, semua fungsi, data dan pengolahan data akan dibungkus dalam kelas-kelas dan object-object.

Masing-masing object dapat memiliki sifat dan tugasnya. Pada paradigm ini, object-object tersebut dapat berkerja sendiri dan juga dapat saling bekerja sama dengan kemungkinan untuk saling berhubungan, seperti menerima, mengirim data kepada object lainnya dan memproses data.

Paradigma OOP dapat dilihat sebagai interaksi dari object yang saling berhubungan satu sama lain untuk melakukan tugasnya. Pengembangan program OOP dilakukan dengan cara mengikuti model yang telah ada di dalam dunia nyata (Object-Oriented World).

OOP bertujuan untuk memberikan pola pikir dalam mengembangkan program, pola pikir tersebut dipercaya dapat memberikan kemudahan, fleksibilitas, kemudahan pembuatan, pengembangan program, dan perawatan program.

Dan juga banyak pendapat yang mengatakan bahwa pemrograman yang berbasis object lebih mudah dipelajari oleh pemula dibandingkan dengan bahasa pemrograman yang tidak berbasis object.

Class (Kelas) adalah sebuah rancangan (mirip seperti struct) untuk mendefinisikan karakter dan perilaku dari object. yang merupakan kumpulan atas definisi dan fungsi-fungsi dalam suatu unit, untuk suatu tujuan tertentu. Class (Kelas) terdiri dari model objek yang memiliki atribut (data members) dan Behaviors (member functions), dan Member functions yaitu Methods yang dipanggil sebagai response terhadap pesan. Kelas didefinisikan dengan keyword class. Attribute adalah segala sesuatu (data)yang berhubungan dengan karakteristik objek dalam setiap kelas. Tiap atribut memiliki nilai berbeda untuk setiap objek. Object atau instance adalah dasar dari modularitas dan struktur pada OOP. dan merupakan representasi dari class, object akan memiliki sifat dan perilaku dari class yang digunakan.

Enkapsulasi (encapsulation) adalah konsep dalam implementasi untuk membungkus data dan fungsi menjadi satu entitas, dan membatasi akses dari luar class. Enkapsulasi (encapsulation) adalah sebuah metoda untuk mengatur struktur class dengan cara menyembunyikan alur kerja dari class tersebut. Struktur class yang dimaksud adalah property dan method. Dengan enkapsulasi, kita bisa membuat pembatasan akses kepada property dan method, sehingga hanya property dan method tertentu saja yang bisa diakses dari luar class. Enkapsulasi juga dikenal dengan istilah 'information hiding'.

Access Modifier (kadang juga disebut Access Specifier) adalah salah satu fitur penting dalam Object Oriented Programming (OOP) untuk melakukan Data Hiding (Menyembunyikan Data). Fitur ini memungkinkan kita untuk mengatur hak akses dari member class, digunakan agar tidak sembarangan perintah dapat mengakses, atau tidak bisa di akses secara langsung.

Fitur ini memiliki 3 tipe Access Modifier, yaitu:

- 1) Public; menyatakan bahwa kelas/method/attribute tersebut dapat diakses oleh kelas lain dimanapun.
- 2) Protected; menyatakan bahwa kelas/method/attribute tersebut dapat diakses oleh kelas lain yang berada dalam satu package atau kelas lain tersebut merupakan turunannya.
- 3) Private; menyatakan bahwa kelas tersebut tidak dapat diakses sama sekali oleh kelas lain bahkan juga tidak dapat diturunkan. Berarti attribute-attribute yang private hanya dapat diakses oleh

Constructor (konstruktor) adalah method khusus yang akan dijalankan secara otomatis pada saat sebuah objek dibuat (instansiasi), yakni ketika perintah “new” dijalankan. Constructor biasa digunakan untuk membuat proses awal dalam mempersiapkan objek, seperti memberi nilai awal kepada property, memanggil method internal dan beberapa proses lain yang digunakan untuk ‘mempersiapkan’ objek.

BAB II

PEMBAHASAN

2.1 Program Hewan Pertama Untuk Absen Genap

```
#include <iostream>
#include <string>

using namespace std;

class Piaraan {
public :
    string Nama;
    string Hewan;
    string Kelas;

    Piaraan(string Nama, string Hewan, string Kelas) {
        Piaraan::Nama = Nama;
        Piaraan::Hewan = Hewan;
        Piaraan::Kelas = Kelas;
    }

    void tampilData(){
        cout<< "Hi " << Nama << endl;
        cout<< "Your pet is " << Hewan << endl;
        cout<< "This creature is one of " << Kelas << endl;
    }

};
```

Time
new
room

Gambar 2.1 Tugas POB dalam c++ pertama

```
int main()
{
    string masukanNama;
    string masukanHewan;
    string masukanKelas;

    cout<< "What Is Your Name : "; getline(cin,
masukanNama);
    cout<< "What Is Your Pet : "; getline(cin,
masukanHewan);
    cout<< "In What Classification That Your Pet Is Into : ";
getline(cin, masukanKelas);
    cout<< endl << endl;

    Piaraan call = Piaraan(masukanNama, masukanHewan,
masukanKelas);
    call.tampilData();
    return 0;
}
```

Gambar 2.2 Tugas POB dalam c++ pertama

Penjelasan:

```
#include <iostream>

#include <string>

using namespace std;
```

Gambar 2.3 File Header

Kode di atas merupakan file header. File header adalah file yang berisi deklarasi untuk berbagai fungsi yang dibutuhkan oleh program baik itu sebagai standar Input/Output maupun sebagai syntax. Berikut adalah macam file header yang terdapat pada program diatas.

- 1) **iostream.h** : 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) **string** : Digunakan agar program dapat membaca tipe data yang menggunakan string di dalamnya.
- 3) **using namespace std** : File header ini terdiri dari 3 statements yaitu “using”, “namespace” dan “std”. Fungsi-fungsi penggunaan using namespace std adalah sebagai berikut :
 - 1.1) Using adalah deklarasi arahan / panggilan / pemberitahuan kepada kompiler untuk penggunaan deklarasi namespace dan anggota namespace.
 - 1.2) Namespace adalah penyedia metode untuk mencegah komflik 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.

- 1.3) 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.

```
class Piaraan {  
public :  
    string Nama;  
    string Hewan;  
    string Kelas;  
    Piaraan(string Nama, string Hewan,  
string Kelas) {  
        Piaraan::Nama = Nama;  
        Piaraan::Hewan = Hewan;  
        Piaraan::Kelas = Kelas;  
    }  
}
```

Gambar 2.4 Class

Kode program di atas merupakan kode yang berjenis class yang bernama *Piaraan*. Class adalah salah satu dari konsep OOP yang digunakan untuk membungkus data abstraksi procedural sebagai deskripsi tergeneralisir atau rancangan dari sebuah object untuk mendefinisikan atau menggambarkan isi dan tingkah laku sebagai entitas dari object. Untuk mendirikan Class kita membutuhkan keyword class yang dilanjutkan dengan pemberian nama dari deklarasi class tersebut. lalu dilanjutkan dengan meletakkan tanda { dan } untuk mengapit definisi dari class. Class termasuk sebuah pernyataan maka dari itu akhir dari deklarasi class diwajibkan untuk mengakhiri class menggunakan tanda titik-koma (;).

Dalam class program tersebut juga terdapat constructor (konstruktor). Berikut adalah konstruktor yang dimaksud.

```
Piaraan(string Nama, string Hewan,  
string Kelas) {  
    Piaraan::Nama = Nama;  
    Piaraan::Hewan = Hewan;  
    Piaraan::Kelas = Kelas;  
}
```

Gambar 2.5 Constructor

Kode di atas merupakan konstruktor. Konstruktor adalah fungsi anggota dari suatu kelas yang secara otomatis dijalankan ketika suatu objek diciptakan. Konstruktor mempunyai aturan sebagai berikut.

- 1) Nama Konstruktor harus sama dengan nama kelas yang mendefiniskannya.
- 2) Konstruktor tidak memiliki nilai balik.
- 3) Konstruktor harus diletakkan pada bagian public.

Konstruktor biasanya digunakan untuk :

- 1) Melakukan alokasi memori.
- 2) Memberikan nilai awal terhadap anggota kelas (atau disebut inisialisasi), dan melakukan tugas umum lainnya.

```
void tampilData(){  
    cout<< "Hi " << Nama << endl;  
    cout<< "Your pet is " << Hewan << endl;  
    cout<< "This creature is one of " << Kelas << endl;  
}
```

Gambar 2.6 Function

Kode di atas merupakan function dan function yang digunakan adalah 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.

```
int main()
{
    string masukanNama;
    string masukanHewan;
    string masukanKelas;

    cout<< "What Is Your Name : "; getline(cin,
masukanNama);
    cout<< "What Is Your Pet : "; getline(cin,
masukanHewan);
    cout<< "In What Classification That Your Pet Is Into : ";
getline(cin, masukanKelas);
    cout<< endl << endl;

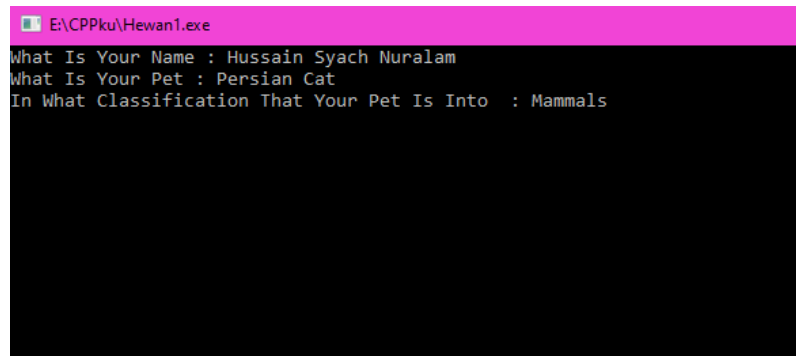
    Piaraan call = Piaraan(masukanNama, masukanHewan,
masukanKelas);
    call.tampilData();

    return 0;
}
```

Gambar 2.7 Main program

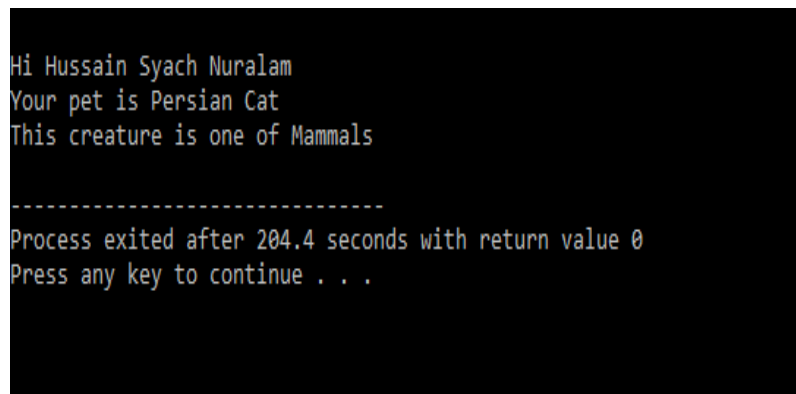
Kode di atas merupakan int main(). Int main() artinya main program mengembalikan nilai int secara default, int main() akan mengembalikan nilai 0, dan fungsi main() tidak memiliki bagan deklarasi lokal, dan hanya memiliki sebuah pernyataan yang dapat dieksekusi, berupa fungsi output printf().

Dari program pertama yang telah dijabarkan di atas, maka didapatkanlah output program sebagai berikut.



```
E:\CPPku\Hewan1.exe
What Is Your Name : Hussain Syach Nuralam
What Is Your Pet : Persian Cat
In What Classification That Your Pet Is Into : Mammals
```

Gambar 2.8 Output untuk isi data



```
Hi Hussain Syach Nuralam
Your pet is Persian Cat
This creature is one of Mammals
-----
Process exited after 204.4 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.9 Output untuk tampil data

Berdasarkan output yang didapat, output dapat ditampilkan karena program memanggil class *Piaraan* yang berisi variabel data string kemudian memanggil tampilData.

2.2 Program Hewan Kedua Untuk Absen Genap

```
#include <iostream>
#include <string>

using namespace std;

class Hewan {
public :
    string Nama;
    string Kelas;
    string Kaki;
    string Spesialisasi;

    Hewan(string Nama, string Kelas, string Kaki, string Spesialisasi) {
        Hewan::Nama = Nama;
        Hewan::Kelas = Kelas;
        Hewan::Kaki = Kaki;
        Hewan::Spesialisasi = Spesialisasi;
    }

    void tampilData(){
        cout<< "Hewan ini adalah " << Nama << endl;
        cout<< "Ia merupakan hewan golongan " << Kelas << endl;
        cout<< "Hewan ini merupakan hewan berkaki " << Kaki << endl;
        cout<< "Hewan ini memiliki spesialisasi yakni " << Spesialisasi <<
endl;
    }

};
```

Gambar 2.10 Tugas POB dalam c++ kedua

```
int main()
{
    string masukanNama;
    string masukanKelas;
    string masukanKaki;
    string masukanSpesialisasi;

    cout<< "Nama Hewan : "; getline(cin, masukanNama);
    cout<< "Kategori Hewan Tersebut : "; getline(cin,
masukanKelas);
    cout<< "Jumlah Kaki Pada Hewan Ini Adalah : "; getline(cin,
masukanKaki);
    cout<< "Spesialisasi Yang Ada Pada Hewan Ini Adalah : ";
getline(cin, masukanSpesialisasi);
    cout<< endl << endl;

    Hewan call = Hewan(masukanNama, masukanKelas,
masukanKaki, masukanSpesialisasi);
    call.tampilData();

    return 0;
}
```

Gambar 2.11 Tugas POB dalam c++ kedua

Penjelasan :

```
#include <iostream>

#include <string>

using namespace std;
```

Gambar 2.12 File Header

Kode di atas merupakan file header. File header adalah file yang berisi deklarasi untuk berbagai fungsi yang dibutuhkan oleh program baik itu sebagai standar Input/Output maupun sebagai syntax.

```
class Hewan {
public :
    string Nama;
    string Kelas;
    string Kaki;
    string Spesialisasi;

    Hewan(string Nama, string Kelas, string Kaki, string Spesialisasi) {
        Hewan::Nama = Nama;
        Hewan::Kelas = Kelas;
        Hewan::Kaki = Kaki;
        Hewan::Spesialisasi = Spesialisasi;
    }
};
```

Gambar 2.13 Class

Kode program di atas merupakan kode yang berjenis class yang bernama *Hewan*. Class adalah salah satu dari konsep OOP yang digunakan untuk membungkus data abstraksi procedural sebagai deskripsi tergeneralisir atau rancangan dari sebuah object untuk mendefinisikan atau menggambarkan isi dan tingkah laku sebagai entitas dari object. Dalam class ini, terdapat konstruktor seperti berikut.

```
Hewan(string Nama, string Kelas, string Kaki, string Spesialisasi) {  
    Hewan::Nama = Nama;  
    Hewan::Kelas = Kelas;  
    Hewan::Kaki = Kaki;  
    Hewan::Spesialisasi = Spesialisasi;  
}
```

Gambar 2.14 Constructor

Kode di atas merupakan konstruktor. Konstruktor adalah fungsi anggota dari suatu kelas yang secara otomatis dijalankan ketika suatu objek diciptakan.

```
void tampilData(){  
    cout<< "Hewan ini adalah " << Nama << endl;  
    cout<< "Ia merupakan hewan golongan " << Kelas << endl;  
    cout<< "Hewan ini merupakan hewan berkaki " << Kaki << endl;  
    cout<< "Hewan ini memiliki spesialisasi yakni " << Spesialisasi <<  
    endl;  
}
```

Gambar 2.15 Function

Kode di atas merupakan function dan function yang digunakan adalah 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.


```

int main()
{
    string masukanNama;
    string masukanKelas;
    string masukanKaki;
    string masukanSpesialisasi;

    cout<< "Nama Hewan : "; getline(cin, masukanNama);
    cout<< "Kategori Hewan Tersebut : "; getline(cin,
masukanKelas);
    cout<< "Jumlah Kaki Pada Hewan Ini Adalah : "; getline(cin,
masukanKaki);
    cout<< "Spesialisasi Yang Ada Pada Hewan Ini Adalah : ";
getline(cin, masukanSpesialisasi);
    cout<< endl << endl;

    Hewan call = Hewan(masukanNama, masukanKelas,
masukanKaki, masukanSpesialisasi);
    call.tampilData();

    return 0;
}


```

Gambar 2.16 Main program

Kode di atas merupakan int main(). Int main() artinya main program mengembalikan nilai int secara default, int main() akan mengembalikan nilai 0,

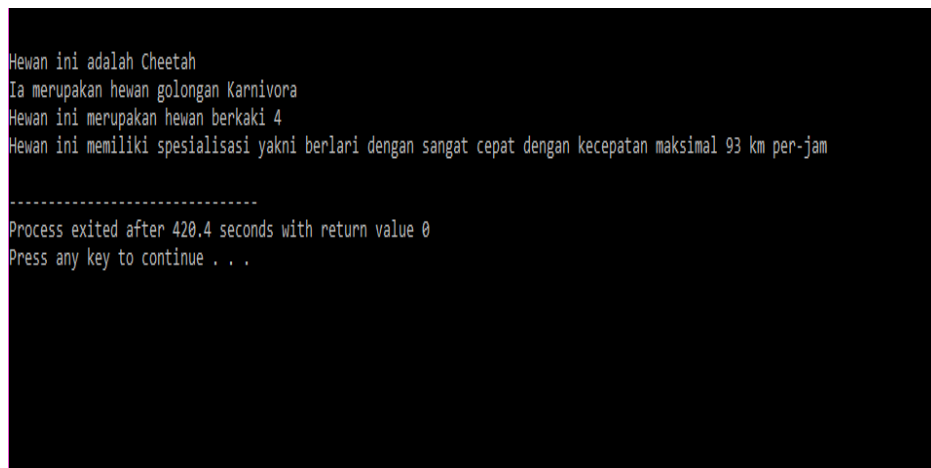
dan fungsi `main()` tidak memiliki bagan deklarasi lokal, dan hanya memiliki sebuah pernyataan yang dapat dieksekusi, berupa fungsi `output printf()`.

Dari program yang telah dijabarkan di atas, didapatlah output sebagai berikut.



```
E:\CPPku\Hewan2.exe
Nama Hewan : Cheetah
Kategori Hewan Tersebut : Karnivora
Jumlah Kaki Pada Hewan Ini Adalah : 4
Spesialisasi Yang Ada Pada Hewan Ini Adalah : berlari dengan sangat cepat dengan kecepatan maksimal 93 km per-jam
```

Gambar 2.17 Output program isi data



```
Hewan ini adalah Cheetah
Ia merupakan hewan golongan Karnivora
Hewan ini merupakan hewan berkaki 4
Hewan ini memiliki spesialisasi yakni berlari dengan sangat cepat dengan kecepatan maksimal 93 km per-jam

-----
Process exited after 420.4 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.18 Output program tampil data

Berdasarkan output yang didapat, output dapat ditampilkan karena program memanggil class *Hewan* yang berisi variabel data string kemudian memanggil `tampilData`.

BAB III

KESIMPULAN

Dapat disimpulkan bahwa class merupakan blueprint (cetak biru) untuk menciptakan suatu instance dari objek dimana terdiri dari sekumpulan objek dengan kemiripan data / properties / attributes, fungsi / behavior / method dan relasi ke objek lain. Pemrograman C++ memungkinkan pembuatan class lebih dari 1. Ketika data dan fungsi yang terkait disimpan di dalam sebuah class mampu membantu mem-visualisasikan permasalahan yang kompleks dengan efisien dan efektif. Sehingga, dapat dikatakan bahwa kelas adalah representasi benda (objek) dari dunia nyata.

DAFTAR PUSTAKA

mis mirin

- Andre. Tutorial Belajar OOP PHP Part 4: Pengertian Enkapsulasi Objek (Public, Protected dan Private). 2014.<https://www.duniailkom.com/tutorial-belajar-oop-php-pengertian-enkapsulasi-objek-public-protected-dan-private/> (Diakses pada : Selasa, 07 April 2020 Pukul 18.00 WIB)
- Andre. Tutorial Belajar OOP PHP Part 8: Pengertian Constructor dan Destructor. 2014.<https://www.duniailkom.com/tutorial-belajar-oop-php-pengertian-structor-dan-destructor/> (Diakses pada : Selasa, 07 April 2020 Pukul 18.00 WIB)
- Andrians, Rendy. Perbedaan Int main dan void main. 2014. <http://rendyandrians.blogspot.com/2014/08/perbedaan-int-main-dan-void-main.html> (Diakses pada : Rabu, 08 April 2020 Pukul 21.30 WIB)
- Dosen Teknik Informatika. Algoritma dan Pemrograman I. 2019. Modul Praktikum Algoritma dan Pemrograman II. Universitas Palangka Raya. Fakultas Teknik. Jurusan Teknik Informatika.
- Fajar. Penjelasan dan Cara Pembuatan Class dan Object. 2018. <https://www.belajarcpp.com/tutorial/cpp/class/> (Diakses pada : Selasa, 07 April 2020 Pukul 18.30 WIB)
- Fajar. Penjelasan dan Konsep OOP (Object Oriented Programming). 2019.<https://www.belajarcpp.com/tutorial/cpp/penjelasan-oop/> (Diakses pada : Selasa, 07 April 2020 Pukul 18.30 WIB)
- Fajar. Penjelasan dan Macam-macam Access Modifier (Akses pengubah). 2019. <https://www.belajarcpp.com/tutorial/cpp/access-modifier/> (Diakses pada : Selasa, 07 April 2020 Pukul 18.30 WIB)
- Fathvizard. Class, Method, Attribute (Tugas OOP1). 2012. <https://fvizard.wordpress.com/2012/10/17/class-method-attribute/> (Diakses pada : Selasa, 07 April 2020 Pukul 19.40 WIB)
- Fatturahman, Rizky. Kegunaan Using namespace std. 2017.<http://rzkyiff.blogspot.com/2017/03/kegunaan-using-namespace-std.html> (Diakses pada : Selasa, 07 April 2020 Pukul 19.40 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 (Diakses pada : Rabu, 08 April 2020 Pukul 23.30 WIB)

Pratama, Ari. Apa Itu Void Dan Ciri-Cirinya. 2015. <https://aripratama.com/apa-itu-void-dan-ciri-cirinya/> (Diakses pada : Selasa, 07 April 2020 Pukul 20.30 WIB)

LAMPIRAN

```
#include <iostream>
#include <string>

using namespace std;

class Piaraan {
public :
    string Nama;
    string Hewan;
    string Kelas;

    Piaraan(string Nama, string Hewan, string Kelas) {
        Piaraan::Nama = Nama;
        Piaraan::Hewan = Hewan;
        Piaraan::Kelas = Kelas;
    }

    void tampilData(){
        cout<< "Hi " << Nama << endl;
        cout<< "Your pet is " << Hewan << endl;
        cout<< "This creature is one of " << Kelas << endl;
    }

};
```

Gambar 2.1 Tugas POB dalam c++ pertama

```
int main()
{
    string masukanNama;
    string masukanHewan;
    string masukanKelas;

    cout<< "What Is Your Name : "; getline(cin,
masukanNama);
    cout<< "What Is Your Pet : "; getline(cin,
masukanHewan);
    cout<< "In What Classification That Your Pet Is Into : ";
getline(cin, masukanKelas);
    cout<< endl << endl;

    Piaraan call = Piaraan(masukanNama, masukanHewan,
masukanKelas);
    call.tampilData();

    return 0;
}
```

Gambar 2.2 Tugas POB dalam c++ pertama

```
#include <iostream>

#include <string>

using namespace std;
```

Gambar 2.3 File Header

```
class Piaraan {
public :
    string Nama;
    string Hewan;
    string Kelas;
    Piaraan(string Nama, string Hewan,
string Kelas) {
        Piaraan::Nama = Nama;
        Piaraan::Hewan = Hewan;
        Piaraan::Kelas = Kelas;
    }
}
```

Gambar 2.4 Class

```
Piaraan(string Nama, string Hewan,
string Kelas) {
    Piaraan::Nama = Nama;
    Piaraan::Hewan = Hewan;
    Piaraan::Kelas = Kelas;
}
```

Gambar 2.5 Constructor


```

void tampilData(){
    cout<< "Hi " << Nama << endl;
    cout<< "Your pet is " << Hewan << endl;
    cout<< "This creature is one of " << Kelas << endl;
}

```

Gambar 2.6 Function

```

int main()
{
    string masukanNama;
    string masukanHewan;
    string masukanKelas;

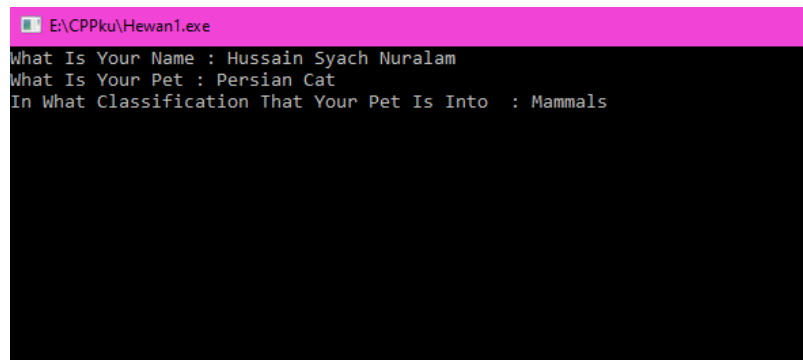
    cout<< "What Is Your Name : "; getline(cin,
masukanNama);
    cout<< "What Is Your Pet : "; getline(cin,
masukanHewan);
    cout<< "In What Classification That Your Pet Is Into : ";
getline(cin, masukanKelas);
    cout<< endl << endl;

    Piaraan call = Piaraan(masukanNama, masukanHewan,
masukanKelas);
    call.tampilData();

    return 0;
}

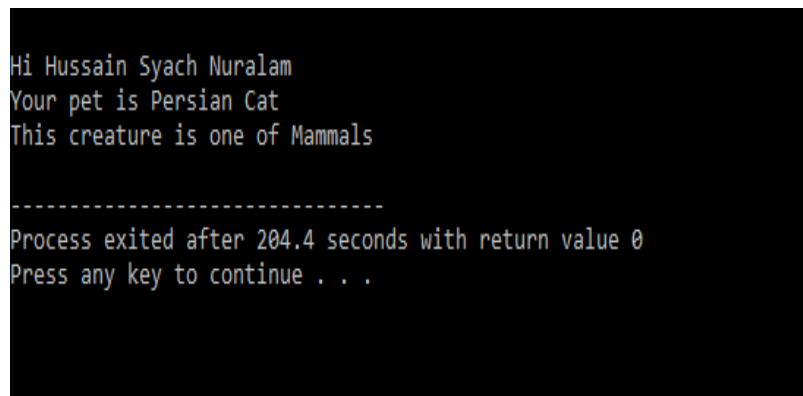
```

Gambar 2.7 Main program



```
E:\CPPku\Hewan1.exe
What Is Your Name : Hussain Syach Nuralam
What Is Your Pet : Persian Cat
In What Classification That Your Pet Is Into : Mammals
```

Gambar 2.8 Output untuk isi data



```
Hi Hussain Syach Nuralam
Your pet is Persian Cat
This creature is one of Mammals

-----
Process exited after 204.4 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.9 Output untuk tampil data

```

#include <iostream>
#include <string>

using namespace std;

class Hewan {
public :
    string Nama;
    string Kelas;
    string Kaki;
    string Spesialisasi;

    Hewan(string Nama, string Kelas, string Kaki, string Spesialisasi) {
        Hewan::Nama = Nama;
        Hewan::Kelas = Kelas;
        Hewan::Kaki = Kaki;
        Hewan::Spesialisasi = Spesialisasi;
    }

    void tampilData(){
        cout<< "Hewan ini adalah " << Nama << endl;
        cout<< "Ia merupakan hewan golongan " << Kelas << endl;
        cout<< "Hewan ini merupakan hewan berkaki " << Kaki << endl;
        cout<< "Hewan ini memiliki spesialisasi yakni " << Spesialisasi <<
endl;
    }

};

```

Gambar 2.10 Tugas POB dalam c++ kedua

```
int main()
{
    string masukanNama;
    string masukanKelas;
    string masukanKaki;
    string masukanSpesialisasi;

    cout<< "Nama Hewan : "; getline(cin, masukanNama);
    cout<< "Kategori Hewan Tersebut : "; getline(cin,
masukanKelas);
    cout<< "Jumlah Kaki Pada Hewan Ini Adalah : "; getline(cin,
masukanKaki);
    cout<< "Spesialisasi Yang Ada Pada Hewan Ini Adalah : ";
getline(cin, masukanSpesialisasi);
    cout<< endl << endl;

    Hewan call = Hewan(masukanNama, masukanKelas,
masukanKaki, masukanSpesialisasi);
    call.tampilData();

    return 0;
}
```

Gambar 2.11 Tugas POB dalam c++ kedua

```
#include <iostream>

#include <string>

using namespace std;
```

Gambar 2.12 File Header

```
class Hewan {
public :
    string Nama;
    string Kelas;
    string Kaki;
    string Spesialisasi;

    Hewan(string Nama, string Kelas, string Kaki, string Spesialisasi) {
        Hewan::Nama = Nama;
        Hewan::Kelas = Kelas;
        Hewan::Kaki = Kaki;
        Hewan::Spesialisasi = Spesialisasi;
    }
};
```

Gambar 2.13 Class

```
Hewan(string Nama, string Kelas, string Kaki, string Spesialisasi) {
    Hewan::Nama = Nama;
    Hewan::Kelas = Kelas;
    Hewan::Kaki = Kaki;
    Hewan::Spesialisasi = Spesialisasi;
}
```

Gambar 2.14 Constructor

```
void tampilData(){  
    cout<< "Hewan ini adalah " << Nama << endl;  
    cout<< "Ia merupakan hewan golongan " << Kelas << endl;  
    cout<< "Hewan ini merupakan hewan berkaki " << Kaki << endl;  
    cout<< "Hewan ini memiliki spesialisasi yakni " << Spesialisasi <<  
    endl;  
}
```

Gambar 2.15 Function

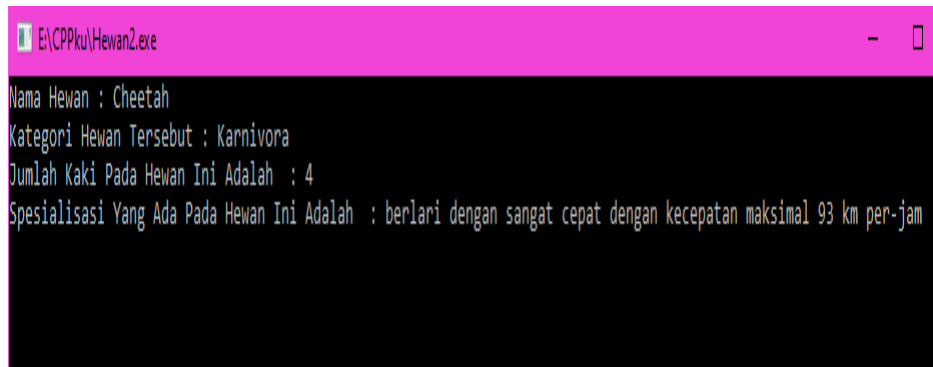
```
int main()
{
    string masukanNama;
    string masukanKelas;
    string masukanKaki;
    string masukanSpesialisasi;

    cout<< "Nama Hewan : "; getline(cin, masukanNama);
    cout<< "Kategori Hewan Tersebut : "; getline(cin,
masukanKelas);
    cout<< "Jumlah Kaki Pada Hewan Ini Adalah : "; getline(cin,
masukanKaki);
    cout<< "Spesialisasi Yang Ada Pada Hewan Ini Adalah : ";
getline(cin, masukanSpesialisasi);
    cout<< endl << endl;

    Hewan call = Hewan(masukanNama, masukanKelas,
masukanKaki, masukanSpesialisasi);
    call.tampilData();

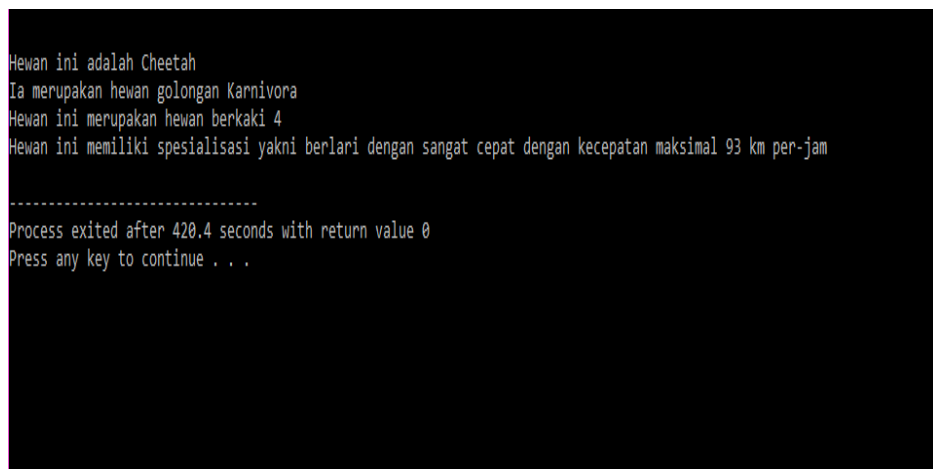
    return 0;
}
```

Gambar 2.16 Main program



```
E:\CPPku\Hewan2.exe
Nama Hewan : Cheetah
Kategori Hewan Tersebut : Karnivora
Jumlah Kaki Pada Hewan Ini Adalah : 4
Spesialisasi Yang Ada Pada Hewan Ini Adalah : berlari dengan sangat cepat dengan kecepatan maksimal 93 km per-jam
```

Gambar 2.17 Output program isi data



```
Hewan ini adalah Cheetah
Ia merupakan hewan golongan Karnivora
Hewan ini merupakan hewan berkaki 4
Hewan ini memiliki spesialisasi yakni berlari dengan sangat cepat dengan kecepatan maksimal 93 km per-jam

-----
Process exited after 420.4 seconds with return value 0
Press any key to continue . . .
```

Gambar 2.18 Output program tampil data