

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II



NAMA : ERVINA AYU ROSAMADA
NIM : 193020503020
KELAS : A
MODUL : I (DASAR PEMROGRAMAN
BERORIENTASI OBJEK)

g. 1.5

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA

2020

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II



NAMA : ERVINA AYU ROSAMADA

NIM : 193020503020

KELAS : A

**MODUL : I (DASAR PEMROGRAMAN
BERORIENTASI OBJEK)**

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	9
BAB II Pembahasan	60	55
BAB III Kesimpulan	20	18
Daftar Pustaka	5	3
Lampiran	5	3
Jumlah	100	88

Penilai
Asisten Praktikum

Diana

BAB I

TUJUAN DAN LANDASAN TEORI

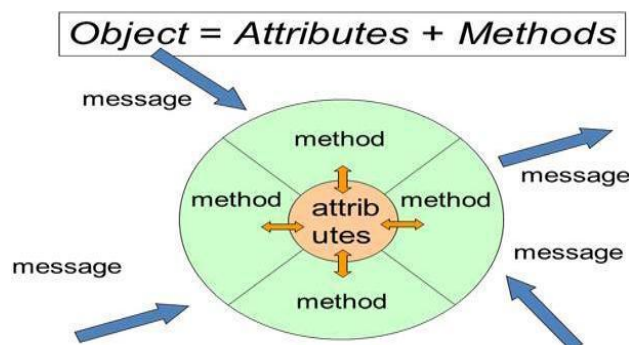
I. TUJUAN

Setelah menyelesaikan modul ini, mahasiswa diharapkan mampu:

1. memahami dasar-dasar pemrograman berorientasi obyek
2. Memahami enkapsulasi
3. membuat kelas dan objek

II. DASAR 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. Pada PBO dilakukan pembungkusan data (attributes) dan fungsi (behavior) ke paket yang disebut kelas. Attributes merupakan data yang menggambarkan status internal sebuah objek dan biasanya merupakan “member variables” pada C++, tidak dapat diakses dari luar (enkapsulasi), dan juga sebagai “state”. Methods merupakan fungsi yang mengakses status internal sebuah objek dan biasanya merupakan “member functions” pada C++, dapat diakses dari luar, memanipulasi atribut, dan disebut juga “behavior”. Berikut ini merupakan gambaran mengenai objek.



Kelas (Class) 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.

Mode Akses akses yang ada pada kelas ada tiga yaitu private yang merupakan default mode akses dan dapat diakses oleh member functions, public yang dapat diakses oleh setiap Accessible fungsi dalam program, dan protected yang biasanya digunakan untuk pewarisan . Fungsi Constructor merupakan member function khusus yang menginisialisasi data members dan memiliki nama yang sama dengan nama kelas. Fungsi Constructor dipanggil saat membuat objek dari kelas dan tidak memiliki tipe balikan. Member functions yang didefinisikan di luar kelas dilakukan dengan menggunakan binary scope resolution operator (::) yang berfungsi untuk “mengikat” nama fungsi ke nama kelas dan mengidentifikasi fungsi dari suatu kelas tertentu. Berikut ini merupakan format dari member functions. NilaiBalikan NamaKelas::NamaFungsi() { ... }

Member functions yang didefinisikan di dalam kelas tidak membutuhkan scope resolution operator dan nama kelas

Istilah – istilah Pemrograman Berorientasi Objek (PBO)

a. Object

Untuk mempermudah pemahaman, maka disini akan dijelaskan melalui analogi. Pada dasarnya semua benda yang ada di dunia nyata dapat dianggap sebagai object. Misalnya rumah, mobil, sepeda, motor, gelas, komputer, meja, sepatu, dll. Setiap object memiliki atribut sebagai status (state) dan tingkah laku sebagai behavior. Contoh objek : Motor. Maka attribute (state) nya adalah pedal, roda, jeruji, speedometer, warna, jumlah roda. Sedangkan tingkah laku (behavior) nya adalah kecepatan menaik, kecepatan menurun, dan perpindahan gigi motor. Analogi pemograman berorientasi objek sama dengan penggambaran pada dunia nyata seperti

contoh diatas. Dalam PBO, state disimpan pada variabel dan tingkah laku disimpan pada method. Dalam bahasa teoretis PBO, object berfungsi untuk membungkus data dan fungsi bersama menjadi satu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

b. Class

Definisi class yaitu template untuk membuat objek. Class merupakan prototype atau blue prints yang mendefinisikan variabel-variabel dan method-method secara umum. Objek merupakan hasil instansiasi dari suatu class. Proses pembentukan objek dari suatu kelas disebut sebagai instantiation. Objek disebut juga sebagai instances.

Dalam bahasa teoretis PBO, class merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemograman berorientasi object.

Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan PBO). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

c. Attributes

Atribut adalah data yang membedakan antara objek satu dengan yang lainnya. Dalam class, atribut sering disebut dengan variabel. Atribut dibedakan menjadi dua jenis yaitu Instance Variabel dan Class Variabel. Instance Variabel adalah atribut untuk tiap objek dari kelas yang sama. Tiap objek mempunyai dan menyimpan nilai atributnya sendiri. Jadi, tiap

objek dari class yang sama boleh mempunyai nilai yang sama atau berbeda. Class Variabel adalah atribut untuk semua objek yang dibuat dari class yang sama. Semua objek mempunyai nilai atribut yang sama. Jadi semua objek dari class yang sama mempunyai hanya satu nilai yang value nya sama.

d. Behavior

Behavior/tingkah laku adalah hal-hal yang bisa dilakukan oleh objek dari suatu class. Behavior dapat digunakan untuk mengubah nilai atribut suatu objek, menerima informasi dari objek lain, dan mengirim informasi ke objek lain untuk melakukan suatu tugas (task).

Dalam class, behavior disebut juga sebagai methods. Methods sendiri adalah serangkaian statements dalam suatu class yang handle suatu task tertentu. Cara objek berkomunikasi dengan objek yang lain adalah dengan menggunakan methods.

BAB II

PEMBAHASAN



Program I

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

class ayam {
    public:
        ayam (int);
        int jumlahkaki();
        void data();
        ayam (const char *makanan, const char *kembangbiak);
    private:
        char makan[30];
        char kbiak[30];
        int kaki;
};

ayam::ayam(int t)
{
    kaki=t;
};

ayam::ayam(const char *makanan, const char *kembangbiak)
{
    strcpy (makan, makanan);
    strcpy (kbiak, kembangbiak);
}

int ayam::jumlahkaki()
{
```

```

        return kaki;
    }
    void ayam::data()
    {
        cout<<"berdasarkan jenis makanan      : "<<makan<<endl;
        cout<<"-----:"<<endl;
        cout<<"berdasarkan cara berkembang biak: "<<kbiak<<endl;
    }
    int main ()
    {
        ayam a ("omnivora", "ovipar");
        ayam a1 (2);
        cout<<"=====:"<<endl;
        cout<<"KLARIFIKASI HEWAN AYAM"<<endl;
        cout<<"=====:"<<endl;
        cout<<"jumlah kaki          : "<<a1.jumlahkaki()<<endl;
        cout<<"-----:"<<endl;
        a.data();
        cout<<"-----:"<<endl;

        return 0;
    }

```

```
#include<iostream>
```

Tanda # disebut sebagai preprocessor directive, #include adalah sebuah professor pengarah yang menunjukkan kepada compiler untuk meletakkan kode dari header file iostream.

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


header yang berisi fungsi – fungsi makro dan tipe yang digunakan untuk pengoperasian string.

```
Using namespace std;
```

Menyatakan bahwa program akan menggunakan namespace yang bernama std, sehingga tidak perlu untuk menyertakan identitas std:: pada anggota dari namespace std salah satunya adalah cout.

```
class ayam {
```

Class adalah rancangan yang mendefinisikan variabel dan method-method pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program yang akan dijalankan suatu program. Class juga adalah sebuah konsep, konsep tersebut ada dua yaitu : method dan attribute.

```
public:
```

Public berfungsi untuk memberikan hak akses secara umum (public) kepada kelas-kelas turunannya maupun terhadap lingkungan luar didalam program. Bagian ini biasanya berisi fungsi-fungsi yang akan dijadikan sebagai penghubung (interface).

```
ayam (int);
```

Int dalam fungsi utama adalah sebuah return type integer. Menggunakan tipe data integer dikarenakan dalam class ayam ini akan ada jumlah kaki, dan jumlah kaki akan memberikan output bernilai angka sehingga salah satu tipe data yang dipakai adalah integer.

```
int jumlahkaki();
```

Int dalam fungsi utama adalah sebuah return type integer. Karena pada saat menampilkan jumlah kaki menggunakan angka, maka tipe data yang digunakan dalam jumlahkaki adalah integer.

```
void data();
```

Fungsi pada tipe void tidak akan mengembalikan nilai apapun, void biasanya digunakan untuk menjalankan rutin program.

```
ayam (const char *makanan, const char *kembangbiak);
```

Char adalah tipe data yang hanya bisa diisi dengan 1 karakter saja. Karakter ini harus ditulis diantara tanda kutip. Karena hanya bisa diisi dengan satu karakter saja, tipe data char hanya membutuhkan 1 byte memori (1 byte = 8 bit).

```
private:
```

Private digunakan untuk memberikan hak akses data hanya kepada kelas yang bersangkutan saja. Artinya kelas-kelas turunan atau lingkungan luar didalam program tidak diizinkan untuk mengakses data tersebut.

```
char makan[30];
```

Char adalah tipe data yang hanya bisa diisi dengan 1 karakter saja. Karakter ini harus ditulis diantara tanda kutip. Karena hanya bisa diisi dengan satu karakter saja, tipe data char hanya membutuhkan 1 byte memori (1 byte = 8 bit).

```
char kbiak[30];
```

Char adalah tipe data yang hanya bisa diisi dengan 1 karakter saja. Karakter ini harus ditulis diantara tanda kutip. Karena hanya bisa diisi dengan satu karakter saja, tipe data char hanya membutuhkan 1 byte memori (1 byte = 8 bit).

```
int kaki;
```

Int dalam fungsi utama adalah sebuah return type integer. Karena pada saat menampilkan jumlah kaki menggunakan angka, maka tipe data yang digunakan dalam jumlahkaki adalah integer.

```
};
```

Blok penutup program.

```
ayam::ayam(int t)
```

Berfungsi sebagai pemanggil constructor yang bertipe data integer.

```
{
```

Sebagai blok untuk mengawali program.

```
kaki=t;
```

Berfungsi untuk memudahkan penampilan jumlah kaki di layar output.

```
};
```

Sebagai blok untuk mengakhiri program.

```
ayam::ayam(const char *makanan, const char *kembangbiak)
```

Berfungsi untuk memanggil constructor yang bertipe data char.

```
{
```

Sebagai blok untuk membuka program.

```
strcpy (makan, makanan);
```

Perintah strcpy bisa dipakai untuk mengcopy sebuah string ke dalam string lain, atau sebuah string ke dalam variabel. Strcpy adalah sebuah fungsi atau function.

```
strcpy (kbiak, kembangbiak);
```

Perintah strcpy bisa dipakai untuk mengcopy sebuah string ke dalam string lain, atau sebuah string ke dalam variabel. Strcpy adalah sebuah fungsi atau function.

```
}
```

Berfungsi untuk mengakhiri program.

```
int ayam::jumlahkaki()
```

Int dalam fungsi utama adalah sebuah return type integer.

```
{
```

Blok untuk memulai program

```
return kaki;
```

Mengakhiri program perhitungan jumlah kaki.

```
}
```

Blok untuk mengakhiri program.

```
void ayam::data()
```

Perintah void, void sering disebut juga prosedur, disebut void karena fungsi tersebut tidak mengembalikan suatu nilai keluaran yang didapat dari hasil proses fungsi tersebut, void digunakan untuk sebuah function atau procedure yang tidak membutuhkan nilai balik.

```
{
```

Blok untuk memulai program.

```
cout<<"berdasarkan jenis makanan    : "<<makan<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda " untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"-----:"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda " untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"berdasarkan cara berkembang biak: "<<kbiak<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda " untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
}
```

Blok untuk mengakhiri program.

```
int main ()
```

Int main() merupakan fungsi (function) utama, fungsi yang akan dibaca pertama kali oleh kompilator secara otomatis, tidak akan function yang dibaca otomatis oleh kompilator kecuali function utama *int main()*, fungsi utama adalah program yang mengatur arah kompilator. *Int* dalam fungsi utama adalah sebuah return type integer. Tanda () merupakan tempat mengisi parameter untuk function, tetapi dalam fungsi utama, hal itu tidak dibutuhkan dan dibiarkan dalam keadaan kosong.

```
{
```

Blok untuk memulai program.

```
ayam a ("omnivora", "ovipar");
```

Bagian ini berfungsi untuk menampilkan di bagian void data, dengan inisial *a* bisa dengan mudah untuk memanggil sebuah fungsi.

```
ayam a1 (2);
```

Bagian ini untuk menampilkan sebuah data yang terdapat pada *cout*.

```
cout<<"=====:"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"KLARIFIKASI HEWAN AYAM"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"=====:"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"jumlah kaki          : "<<a1.jumlahkaki()<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
cout<<"-----:"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota namespace *std* yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
a.data();
```

Berfungsi untuk memanggil fungsi yaitu void data.

```
cout<<"-----:"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. *Endl* merupakan anggota dari *iostream* dan masuk sebagai anggota

namespace std yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
return 0;
```

Return 0; merupakan pernyataan pengembalian, untuk memberi tahu kepada system operasi bahwa program telah berakhir secara normal dengan bernilai 0 yang berarti tanpa kesalahan.

```
}
```

Berfungsi untuk mengakhiri program.

Program II

```
#include<iostream>
using namespace std;

class kucing {
    public:
        kucing (int);
        int jumlahkaki();
        void data();
    private:
        int kaki;
};

kucing::kucing(int t) {

    kaki=t;
};

int kucing::jumlahkaki()
{
```



```

        return kaki;
    }
    void kucing::data()
    {
        cout<<"Kucing adalah hewan yang menggemaskan"<<endl;
    }

    int main ()
    {
        kucing k (4);
        cout<<"Kucing ";
        cout<<"jumlah kakinya ada "<<k.jumlahkaki()<<endl;
        k.data();
        return 0;
    }

```

```
#include<iostream>
```

Tanda # disebut sebagai preprocessor directive, #include adalah sebuah professor pengarah yang menunjukkan kepada compiler untuk meletakkan kode dari header file iostream.

```
using namespace std;
```

Menyatakan bahwa program akan menggunakan namespace yang bernama std, sehingga tidak perlu untuk menyertakan identitas std:: pada anggota dari namespace std salah satunya adalah cout.

```
class kucing {
```

Class adalah rancangan yang mendefinisikan variabel dan method-method pada seluruh objek tertentu. Class berfungsi untuk menampung isi dari program

yang akan dijalankan suatu program. Class juga adalah sebuah konsep, konsep tersebut ada dua yaitu : method dan attribute.

public:

Public berfungsi untuk memberikan hak akses secara umum (public) kepada kelas-kelas turunannya maupun terhadap lingkungan luar didalam program. Bagian ini biasanya berisi fungsi-fungsi yang akan dijadikan sebagai penghubung (interface).

kucing (int);

Int (integer) adalah tipe data untuk angka numerik dan tidak menggunakan koma, tipe data int membutuhkan 32 bit. Karena pada saat menampilkan jumlah kaki menggunakan angka, maka tipe data yang digunakan dalam jumlahkaki adalah integer.

void data();

Perintah void, void sering disebut juga prosedur, disebut void karena fungsi tersebut tidak mengembalikan suatu nilai keluaran yang didapat dari hasil proses fungsi tersebut, void digunakan untuk sebuah function atau procedure yang tidak membutuhkan nilai balik.

private:

Private digunakan untuk memberikan hak akses data hanya kepada kelas yang bersangkutan saja. Artinya kelas-kelas turunan atau lingkungan luar didalam program tidak diizinkan untuk mengakses data tersebut.

int kaki;

Int (integer) adalah tipe data untuk angka numerik dan tidak menggunakan koma, tipe data int membutuhkan 32 bit. Karena pada saat menampilkan jumlah kaki menggunakan angka, maka tipe data yang digunakan dalam jumlahkaki adalah integer.

};

Berfungsi sebagai blok penutup.

```
kucing::kucing(int t) {
```

Berfungsi untuk memanggil constructor yang bertipe data integer.

```
kaki=t;
```

Berfungsi untuk memudahkan perhitungan jumlah kaki.

```
int kucing::jumlahkaki()
```

Berfungsi untuk pemanggilan function.

```
{
```

Berfungsi untuk memulai program.

```
return kaki;
```

Berfungsi untuk mengakhiri program jumlah kaki.

```
}
```

Berfungsi untuk mengakhiri program.

```
void kucing::data()
```

Perintah void, void sering disebut juga prosedur, disebut void karena fungsi tersebut tidak mengembalikan suatu nilai keluaran yang didapat dari hasil proses fungsi tersebut, void digunakan untuk sebuah function atau procedure yang tidak membutuhkan nilai balik.

```
{
```

Berfungsi untuk memulai program.

```
cout<<"Kucing adalah hewan yang menggemaskan"<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. Endl merupakan anggota dari iostream dan masuk sebagai anggota

namespace std yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
}
```

Berfungsi untuk mengakhiri program.

```
int main ()
```

Int main() merupakan fungsi (function) utama, fungsi yang akan dibaca pertama kali oleh kompilator secara otomatis, tidak akan function yang dibaca otomatis oleh kompilator kecuali function utama int main(), fungsi utama adalah program yang mengatur arah kompilator. Int dalam fungsi utama adalah sebuah return type integer. Tanda () merupakan tempat mengisi parameter untuk function, tetapi dalam fungsi utama, hal itu tidak dibutuhkan dan dibiarkan dalam keadaan kosong.

```
{
```

Berfungsi untuk mengawali program.

```
kucing k (4);
```

berfungsi sebagai inisialisasi untuk memanggil sebuah fungsi.

```
cout<<"Kucing ";
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput. Tanda " untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor.

```
cout<<"jumlah kakinya ada "<<k.jumlahkaki()<<endl;
```

Cout adalah sebuah objek yang digunakan untuk menampilkan teks di layar monitor dan untuk mengarahkan data ke dalam standar output. Tanda << adalah sebagai operator pemasukan (insertion operators) dan berfungsi untuk menunjukkan pada compiler menampilkan output sesuai dengan yang diinput.

Tanda “ untuk menampilkan kata atau kalimat yang terdapat didalamnya kepada layar monitor. Endl merupakan anggota dari iostream dan masuk sebagai anggota namespace std yang digunakan untuk mengakhiri baris atau pindah baris (enter) pada hasil layar program.

```
k.data();
```

berfungsi sebagai pemanggil function, di mana function yang digunakan adalah data, data tersebut berisi data hewan kucing yaitu seperti yang terdapat di dalam cout.

```
return 0;
```

Return 0; merupakan pernyataan pengembalian, untuk memberi tahu kepada system operasi bahwa program telah berakhir secara normal dengan bernilai 0 yang berarti tanpa kesalahan.

```
}
```

Berfungsi untuk mengakhiri program.

Gambar hasil output

BAB III

KESIMPULAN

Dengan menggunakan PBO maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi oleh objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Semua system terdiri dari class-class dan objek. Suatu system dicapai melalui kerjasama antar object, interaksi antar object disebut object relationship.

Class untuk membuat sebuah objek terlebih dahulu harus dideklarasikan class yang akan dibuat. Method merepresentasikan operasi-operasi yang dapat dilakukan oleh objek. Yang membedakan antara variabel dan method adalah method selalu diakhiri dengan () atau {}. Constructor merupakan suatu method yang akan memberikan nilai awal dari pada saat suatu objek dibuat. Modifier ini digunakan untuk memberi tanda hak akses pada resource tertentu. Modifier tersebut diantaranya adalah public, private, protected, static. Object adalah suatu entry yang memiliki keadaan, behavior, dan identitas yang tugasnya dirumuskan dalam suatu lingkup masalah dengan baik. Object dapat juga disebut suatu instans (instance).

DAFTAR PUSTAKA

<https://www.petanikode.com/cpp-fungsi/>

<https://www.sinauarduino.com/artikel/fungsi-pada-pemrograman-cpp/>

<http://www.adityarizki.net/2012/06/konsep-dasar-pemrograman-berorientasi-objek/>

<http://dioirwansyah.wordpress.com/2009/03/18/kelas-dan-objek-dalam-c-/>

Aluram Penulisan daftar Pustaka

.

LAMPIRAN

```
C:\Users\Acer\Documents\tugas AP1 xx.exe

=====
KLARIFIKASI HEWAN AYAM
=====
jumlah kaki           : 2
-----
berdasarkan jenis makanan : omnivora
-----
berdasarkan cara berkembang biak: ovipar
-----

Process exited after 0.4938 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\Acer\Documents\tugas AP 1.exe

Kucing jumlah kakinya ada 4
Kucing adalah hewan yang menggemaskan

-----

Process exited after 0.2411 seconds with return value 0
Press any key to continue . . .
```

Nomor gambar

