

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



COVER  
tidak  
sesuai

**NAMA : OKTA ARYO**

**NIM : 193020503035**

**KELAS : A**

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

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PALANGKA RAYA**

**2020**

# BAB I

## TUJUAN DAN LANDASAN TEORI

### 1. TUJUAN

Setelah menyelesaikan modul ini, mahasiswa diharapkan mampu :

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

### 2. LANDASAN TEORI

**Pemrograman berorientasi objek** merupakan pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

**Class** adalah 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 pemrograman berorientasi object. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang nonprogrammer 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 OOP). 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.

**Enkapsulasi** digunakan untuk memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

**Attribute** adalah segala sesuatu (data) yang berhubungan dengan karakteristik objek dalam setiap kelas. Tiap atribut memiliki nilai berbeda untuk setiap objek. Method merupakan tingkah laku dari suatu objek, jika bersifat static berarti tingkah laku semua objek dalam kelas tersebut.

**Identifier** adalah suatu pengenal atau pengidentifikasi yang kita deklarasikan agar kompiler dapat mengenalinya. Identifier sendiri dapat berupa nama variabel, konstanta fungsi, kelas, template maupun namespace. Biasanya identifier ini terdiri dari 3 bentuk, yaitu privat public dan protected

1. public: anggota pribadi dari kelas dasar hanya dapat diakses oleh anggota yang kelas dasar.
2. private: anggota publik dari kelas dasar dapat diakses oleh anggota yang kelas dasar, anggota kelas turunan serta anggota yang berada di luar kelas dasar dan kelas turunan.
3. protected: anggota dilindungi dari kelas dasar dapat diakses oleh anggota kelas dasar serta anggota kelas turunan.

Constructor adalah anggota class yang bertugas melakukan inisialisasi obyek (instance) dari suatu class C. Constructor mempunyai nama yang sama dengan nama class, dan tidak mempunyai return value. Sebuah class dapat mempunyai lebih dari satu constructor. Constructor yang tidak mempunyai argumen, disebut default constructor, sebaliknya constructor yang mempunyai lebih dari satu argumen adalah non-default constructor.

Constructor dengan satu default argument tetap merupakan sebuah default constructor.

## BAB II PEMBAHASAN

315

### 1. Program Pertama

```
#include<iostream>
using namespace std;
class buahan
{
public:
    buahan(int);
        int jlhbiji();
        int wrnkulit();
private:
    int biji;
    int kulit;

};
    buahan::buahan(int b)
    {
        biji=b;
        kulit=b;
    }

    int buahan::jlhbiji(){
        return biji;
    }

    int buahan::wrnkulit(){
        return kulit;
    }
int main()
{
```

```
    buahan b(1);
    buahan k(1);
    cout<<"jumlah biji pada buah rambutan adalah :
    "<<b.jlhbiji()<<endl;
    cout<<"jumlah warna kulit pada buah rambutan adalah :
    "<<k.jlhbiji();

    return 0;

}
```

### **Penjelasan :**

```
#include<iostream>
using namespace std;
class buahan
{
public:
    buahan(int);
        int jlhbiji();
        int wrnkulit();
private:
    int biji;
    int kulit;

};
```

Code diatas merupakan kelas yang bernama kelas buahan dimana memiliki jenis public dan private. Didalam code diatas public memiliki methods dan constructor yang memiliki nama yang sama dengan kelas yaitu buahan. Dan private sendiri berisi infutan yang dimasukkan oleh sebuah user atau programmer yang berisi nilai atau objek.

```

buahan::buahan(int b)
{
    biji=b;
    kulit=b;
}

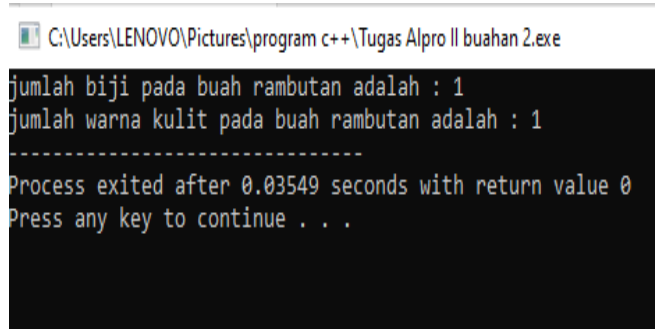
int buahan::jlhbiji(){
    return biji;
}

int buahan::wrnkulit(){
    return kulit;
}
int main()
{
    buahan b(1);
    buahan k(1);
    cout<<"jumlah biji pada buah rambutan adalah :
"<<b.jlhbiji()<<endl;
    cout<<"jumlah warna kulit pada buah rambutan adalah :
"<<k.jlhbiji();

    return 0;
}

```

Diatas ini merupakan code program utama dimana tipe data dari program diatas adalah integer. Didalam program ini juga terdapat cout yang berfungsi sebagai perintah untuk memasukkan infutan atau berfungsi menunjukkan hasil outputnya dimana dalam program ini hasil infutan tidak bisa dilakukan dioutputnya tetapi tergantung infutan user dibatang program utama.



```
C:\Users\LENOVO\Pictures\program c++\Tugas Alpro II buahan 2.exe
jumlah biji pada buah rambutan adalah : 1
jumlah warna kulit pada buah rambutan adalah : 1
-----
Process exited after 0.03549 seconds with return value 0
Press any key to continue . . .
```

Gambar 1.1 program pertama

Gambar diatas adalah hasil output dari program pertama dimana hasil penjumlahan biji dan kulit pada buah rambutan adalah (1). Dimana infutan jumlah nilai dari biji dan kulit buah rambutan adalah (1).

## 2. Program Kedua

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

class buahan
{
public:
    buahan (int);
    int jlhbiji();
    int wrnkulit();
private:
    int biji;
    int kulit;
};

buahan::buahan(int b)
{
    biji=b;
```



```

        kulit=b;
    }

    int buahan::jlhbiji(){
        return biji;
    }

    int buahan::wrnkulit(){
        return kulit;
    }
    int main(){
        buahan b(1);
        buahan k(2);
        cout<<"jumlah biji pada buah nanas adalah :
        "<<b.jlhbiji()<<endl;
        cout<<"jumlah warna kulit pada buah nanas adalah :
        "<<k.wrnkulit();
        return 0;
    }

```

### **Penjelasan :**

```

#include<iostream>
using namespace std;

class buahan
{
public:
    buahan (int);
    int jlhbiji();
    int wrnkulit();
private:

```

```
int biji;  
int kulit;  
};
```

Code diatas merupakan kelas yang bernama kelas buahan dimana memiliki jenis public dan private. Didalam code diatas public memiliki methods dan constructor yang memiliki nama yang sama dengan kelas yaitu buahan. Dan private sendiri berisi infutan yang dimasukkan oleh sebuah user atau programmer yang berisi nilai atau objek. Program kedua ini sama dengan program pertama karena perbedaan dari kedua program tersebut hanya pada coutnya saja.

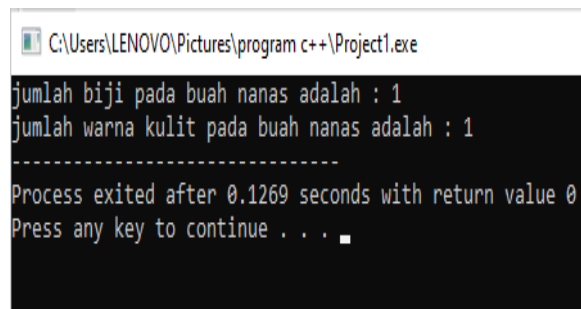
Justif

```
buahan::buahan(int b)  
{  
    biji=b;  
    kulit=b;  
}  
  
int buahan::jlhbiji(){  
    return biji;  
}  
  
int buahan::wrnkulit(){  
    return kulit;  
}  
  
int main(){  
    buahan b(1);  
    buahan k(2);  
    cout<<"jumlah biji pada buah nanas adalah :  
    "<<b.jlhbiji()<<endl;
```

```
cout<<"jumlah warna kulit pada buah nanas adalah :  
"<<k.wrnkulit();  
return 0;
```

Diatas ini merupakan code program utama dimana tipe data dari program diatas adalah integer. Didalam program ini juga terdapat cout yang berfungsi sebagai perintah untuk memasukkan infutan atau berfungsi menunjukkan hasil outputnya dimana dalam program ini hasil infutan tidak bisa dilakukan dioutputnya tetapi tergantung infutan user dibatang program utama.

1 enter



```
C:\Users\LENOVO\Pictures\program c++\Project1.exe  
jumlah biji pada buah nanas adalah : 1  
jumlah warna kulit pada buah nanas adalah : 1  
-----  
Process exited after 0.1269 seconds with return value 0  
Press any key to continue . . .
```

Gambar 1.2 program kedua

1 enter

Gambar diatas adalah hasil output dari program kedua dimana hasil penjumlahan biji dan kulit pada buah nanas adalah (1). Dimana infutan jumlah nilai dari biji dan kulit buah nanas adalah (1).

### BAB III

### KESIMPULAN

Pemrograman berorientasi objek pada dasarnya adalah sebuah program dimana objek lah yang menjadi fokus utamanya. Pemrograman berorientasi objek sendiri memiliki beberapa batang atau tumpuan dalam programnya seperti class, public, private dan lainnya. Didalam pemrograman berorientasi objek terdapat methods yang berfungsi sebagai penerima data atau masukkan nilai dari attribute. Pemrograman berorientasi objek sangat banyak dalam dunia nyata contohnya saja dalam kehidupan sehari-hari sangat sering kita jumpai.

Justify

•

## DAFTAR PUSTAKA

Dosen Teknik Informatika. Algoritma dan Pemrograman I. 2019.  
Modul Praktikum Algoritma dan Pemrograman II. Universitas Palangka  
Raya. Fakultas Teknik. Jurusan Teknik Informatika.  
<https://www.wikipedia.pemrogramanberorientasiobjek.com>

Akuran Penulisan Daftar  
Pustaka



## LAMPIRAN

```
C:\Users\LENOVO\Pictures\program c++\Tugas Alpro II buahan 2.exe  
jumlah biji pada buah rambutan adalah : 1  
jumlah warna kulit pada buah rambutan adalah : 1  
-----  
Process exited after 0.03549 seconds with return value 0  
Press any key to continue . . .
```

**GAMBAR 1.1 PROGRAM PERTAMA** ← non capslock

```
C:\Users\LENOVO\Pictures\program c++\Project1.exe  
jumlah biji pada buah nanas adalah : 1  
jumlah warna kulit pada buah nanas adalah : 1  
-----  
Process exited after 0.1269 seconds with return value 0  
Press any key to continue . . .
```

**GAMBAR 1.2 PROGRAM KEDUA**