LAPORAN HASIL PRAKTIKUM

ALGORITMA DAN PEMROGRAMAN II

NAMA : **SUGENG WAHYU NUGROHO**

NIM : **193010503005**

KELAS : **A**

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

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PALANGKA RAYA

2020

LAPORAN HASIL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN II

NAMA : **SUGENG WAHYU NUGROHO**

NIM : **193010503005**

KELAS : **A**

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

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	9
BAB II Pembahasan	60	50
BAB III Kesimpulan	20	10
Daftar Pustaka	5	3
Lampiran	5	3
Jumlah	100	75

Penilai

Asisten Praktikum

DIANA

BAB I
TUJUAN DAN LANDASAN TEORI

1.1 TUJUAN

Setelah menyelesaikan modul ini, mahasiswa diharapkan mampu:

1.1.1 Memahami dasar-dasar pemrograman berorientasi obyek

1.1.2 Memahami enkapsulasi

1.1.3 Membuat kelas dan objek

1.2 LANDASAN TEORI

1.2.1 Pengertian C++

C++ diciptakan oleh Bjarne Stroustrup di laboratorium Bell pada awal tahun 1980-an, sebagai pengembangan dari bahasa C dan Simula. Saat ini, C++ merupakan salah satu bahasa yang paling populer untuk pengembangan software berbasis OOP. Kompiler untuk C++ telah banyak beredar di pasaran. Software developer yang paling diminati adalah Borland Inc. dan Microsoft Corp. Produk dari Borland untuk kompiler C++ adalah Turbo C++, Borland C++, Borland C++ Builder. Sedangkan dari Microsoft adalah Ms. Visual C++. Walaupun banyak kompiler yang tersedia, namun pada intinya bahasa pemrograman yang dipakai adalah C++.

Sebelum mulai melakukan kode program, sebaiknya diingat bahwa C++ bersifat “case sensitive”, yang artinya huruf besar dan huruf kecil dibedakan.

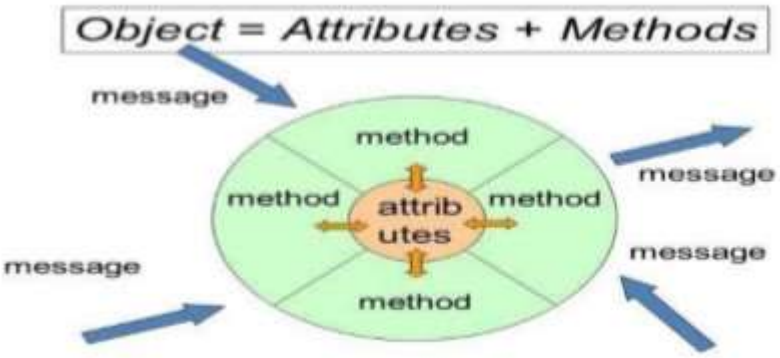
1.2.2 Pengertian PBO

Pemrograman berorientasi objek (Inggris: object-oriented programming disingkat OOP) merupakan paradigma pemrograman berdasarkan konsep "objek", yang dapat berisi data, dalam bentuk field atau dikenal juga sebagai atribut; serta kode, dalam bentuk fungsi/prosedur atau dikenal juga sebagai method. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman

terstruktur. 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 peranti 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.

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".



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 mengindentifikasi fungsi dari suatu kelas tertentu.

Privat merupakan fungsi yang hanya dapat diakses secara internal oleh objek. Sedangkan Public adalah fungsi yang dapat diakses oleh umum atau dapat diakses oleh siapa saja. Dalam badan Class program tersebut, Private yang dapat diaskes oleh program secara khusus adalah string model; string series; int Tipe; jadi maksudnya bahwa lokal ini hanya bleh diskses secara khusus saja. Kemudian dalam badan program terdapat publicyang artinya badan program ini yang akan diakses oleh program secara umum. Dalam badan program public disinilah badan private akan diakses, dimana dalam // Pembentuk atau outputan yang diberikan akan mengisikan data yang dimasukan dalam program ke tempatnya, seperti Modelponsel::model = model; yang mengartikan bahwa isi dari model akan disisi data dari model. Dan bentuk Modelponsel(string model, string series, int Tipe) berguna untuk menjadikan parameter dari yang diambil dari private.

Kemudian dalam // untuk menampilakan digunakan untuk memjadi konstruktor atau pembentuk dalam outputnya. Disana terdapat void perolehInfo yang menjadi konstruktornya.Dalam Badan Class tersebut akan menampilakan outputnya.

1.2.3 Pengertian enkapsulasi

Enkapsulasi adalah proses atau cara menyembunyikan informasi dari suatu class program, enkapsulasi akan melindungi program dari intervensi dari program lain yang dapat mempengaruhinya. hal ini sangat membantu untuk menjaga keutuhan program.

1.2.4 Jenis Enkapsulasi

Private : artinya semua yang berada didalam private mulai dari variabel dll tidak dapat diakses secara bebas, dapat diartikan semua yang berada dalam privat sudah tersembunyi

Public : artinya semua yang berada didalam public mulai dari variabel, class dll dapat diakses secara bebas, artinya siapa saja dapat mengaksesnya,

1.2.5 Manfaat Enkapsulasi

maanfaat dari enkapsulasi salah satunya adalah untuk information hidding yang artinya informasinya tersebunyi, dan juga modularitas memudahkan programmer dalam bekerja.

1.2.6 Pengertian Class dan Obyek

Kelas dan objek merupakan satuan yang berbeda, Kelas Merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu dimana didalamnya terdapat kumpulan atribut dan method,

Contoh kelas : binatang, kendaraan, benda dsb.

Sedangkan objek merupakan bentuk representasi dari sebuah kelas, membungkus data dan fungsi bersama menjadi suatu unit atau entitas dalam sebuah program komputer. pada dasarnya ada 2 karakteristik yang utama pada sebuah objek ;

a. Objek memiliki attribut sebagai status yang disebut Stat
b. Objek memiliki tingkahlaku yang kemudian disebut Method

Contoh sederhananya :

Objek motor memiliki attribute : Roda, Warna, Merk,. Kemudian objek motor tersebut memiliki tingkahlaku / method : pindah gerigi, kecepatan menaik, kecepatan menurun.

contoh objek :

a. Anjing, kucing, kuda : dari kelas binatang
b. Sepeda motor, mobil, pesawat, kapal : dari kelas kendaraan.
c. Batu, air, api, udara : dari kelas benda. dsb

Contoh : *Program SEPEDA*

```
#include<iostream.h>
#include<conio.h>
class Sepeda {
public: Sepeda(int, int, int);
void mengubahPutaran(int);
void mengubahGir(int);
void mengerem();
void tampilInfo();
private: int kecepatan; int putaran; int gir; };
 Sepeda::Sepeda(int k, int p, int g) {
kecepatan = k; putaran = p; gir = g;
 } void Sepeda::mengubahPutaran(int p) { putaran = p; }
void Sepeda::mengubahGir(int g){ gir = g; }
void Sepeda::mengerem() {
 cout<< "Kecepatan dan putaran berkurang ..." ;
 } void Sepeda::tampilInfo()
{ cout<< "Gir : " << gir << endl << "Kecepatan : " <<kecepatan<< endl <<
"Putaran : " << putaran << endl; }
int main() { Sepeda sepeda1(10,60,3), sepeda2(4,12,1), sepeda3(35,80,5);
sepeda1.tampilInfo(); sepeda2.tampilInfo(); sepeda3.tampilInfo();
getch(); return 0; }
```

Contoh : *Program hitung waktu*

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

class Time {
public:
Time();
void setTime( int, int, int );
void printUniversal();
void printStandard();
private:
int hour;
 int minute;
 int second; };
 Time::Time()
{
 hour = minute = second = 0; }
void Time::setTime( int h, int m, int s )
{
 hour = ( h >= 0 && h < 24 ) ? h : 0;
minute = ( m >= 0 && m < 60 ) ? m : 0;
second = ( s >= 0 && s < 60 ) ? s : 0; }
void Time::printUniversal()
{ cout<< setfill( '0' ) << setw( 2 ) << hour << ":" <<setw( 2 ) << minute << ":"
<<setw( 2 ) << second; }
void Time::printStandard()
```

```
cout << ( ( hour == 0 || hour == 12 ) ? 12 : hour % 12 ) << ":" <<setfill( '0' )
<< setw( 2 ) << minute << ":" <<setw( 2 ) << second <<( hour< 12 ? " AM" : "
PM" ); }


int main() { Time t;

cout<< "The initial universal time is "; t.printUniversal();

cout<< "\nThe initial standard time is "; t.printStandard();

t.setTime( 13, 27, 6 );

cout<< "\n\nUniversal time after setTime is "; t.printUniversal();

cout<< "\nStandard time after setTime is "; t.printStandard();

t.setTime( 99, 99, 99 );

cout<< "\n\nAfter attempting invalid settings:" << "\nUniversal time: ";
t.printUniversal();

cout<< "\nStandard time: "; t.printStandard(); cout<< endl;

getch(); return 0; }
```

BAB II
PEMBAHASAN

2.1 Pembahasan tugas praktikum program 1

Program yang dibuat adalah program yang berkaitan dengan buah-buahan dan hewan, namun disini saya mendapat bagian untuk buah-buahan. Yang output nya nanti harus ada unsur buah-buahanya dan bagian dari buah. Dan harus mengacu pada PBO.

```
#include <iostream>
```

Pada baris pertama, akan mendeklarasikan *library* agar program yang dibuat dapat dijalankan sebagaimana mestinya. Library *iostream* berfungsi untuk menampilkan sintaks input dan output program seperti *cin* dan *cout.*

```
using namespace std;

class Buah {
public:
        Buah(int);
        char nama[100];
        char buah[20];
        //consructor
        Buah()
        {
        cout<<"Masukan Nama :";cin.getline(nama,100);
        cout<<"Isi Buah Favorite : ";cin.getline(buah,20);
        cout<<"\n\n";
 }};
```

Sintaks using namespace std; berfungsi untuk mempersingkan penulisan kode yang semula “std::cin”, “std::cout” atau “std::endl” menjadi hanya “cin”, “cout”, dan “endl”.

Pemberian class dimaksudkan agar bisa meberi tempat member class agar mudah dipanggil nantinya dan namanya disini adalah class buah. Public diberikan agar memudahkan dalam pemanggilan karena ini sama saja bersifat global. Buah bertipe data integer untuk nama dalam kurung kotak 100 bermaksud agar nantinya dalam peninputan nama yang bertipe data char bisa sampai 100 karakter begitu juga untuk variabel buah.

Untuk construktor harus sama dengan nama classnya dan nanti outputnya ada nama buah dan akan dipanggil variabel nama. Penambahan getline pada cin bermaksud agar saat input karakter bisa dihapus dan diberi sepasi. Begitu juga selanjutnya.

```
int main()
{
{
Buah a;
a.nama;
a.buah;
cout<<"|################################|"<<endl;
cout<<"|Hai! "<<a.nama<<endl<<"|Buah Favorit kamu
adalah "<<a.buah<<endl;
}cout<<"|################################|";
cin.get();
}
```

int main() { , sintaks ini digunnakan sebagai pembuka bagian utama sebuah program. Di dalam sinilah nantinya program-program yang dimasukkan akan dijalankan.

Untuk Buah a; ini mendeklarasikan pemanggilan class buah dan diberi awalan a diikuti nama variabel.

cout << "=========="; dibuat untuk memperindah tampilan yang akan muncul pada output. Cout<<“Hai” disini berfungsi agar saat telah input nama, maka akan diawali dengan kata “Hai” didepannya , agar mudah saja. Diikuti dengan pernyataan a.nama agar dapat dipanggil. Begitu juga dengan pemanggilan input nama buah.

2.2 Pembahasan Program 2 modul 1

Untuk mengawalinya sama dengan program sebelum nya kita harus menginput library #include <iostream>.

```
using namespace std;
class Buah
{
        public:
                Buah(string, string, string);
                void nama(string);
                void rasa(string);
                void jlh(string);
                void tampil();
        private:
                string Nama_buah;
                string Rasa;
                string Jlh;
};
```

Hampir sama dengan program sebelumnya. Namun disini ditambahkan pada public, Buah dideklarasikan dengan tipe data integer ada 3 karena yang akan ditampilkan cuman 3. Lalu untuk privatenya hanya untuk nama buah, rasa dan jumlah buah. Kenapa menggunakan awalan 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.

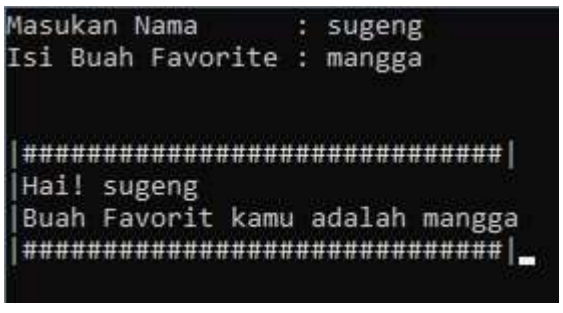
```
Buah::Buah(string n ,string r ,string j)
{
        Nama_buah=n;
        Rasa=r;
        Jlh=j;
}
void Buah::nama(string n)
{
        Nama_buah = n;
}
void Buah::rasa(string r)
{
        Rasa = r;
}
void Buah::jlh(string j)
{
        Jlh = j;
}
```

Nanti pendeklarasian masing masing variabel dilakukan dan pendefinisian pada class bahwa variabel didefinisikan dengan huruf yang diberikan. Contoh Nama_Buah didefinisikan dengan n; dan rasa didefinisikan dengan r, dan Jlh didefinisikan dengn j. Agar memudahkan pemanggilan.

```
void Buah::tampil()

{

        cout<<"|##########################################
##############|"<<endl;

        cout<<"|Nama Buah Ini adalah = " <<Nama_buah<<endl;

        cout<<"|Rasa Buah " <<Nama_buah <<" adalah = "
<<Rasa<<endl;

        cout<<"|Buah " <<Nama_buah << " Ini di Impor dari = "
<<Jlh<<endl<<endl;

        cout<<"|##########################################
##############|"<<endl;

}

int main()

{

         Buah

Buah1(
"NANAS","Asam","Thailand"),Buah2("MANGGA","Manis","Kamboja"),
Buah3( "PISANG","Manis","Zimbabwe");
```

Program diatas hampir sam dengan program sebelumnya hanya saja ditambahkan beberapa variabel dan tambahan lainnya seperto nama buah, rasa, dan jumlah bauh.

2.3 Output program pertama



Gambar 2.1

2.4 Output program kedua

Gambar 2.2

BAB III

KESIMPULAN

PBO berarti suatu program itu harus berorientasi sesuai objek program nya. Dan harus ada unsur pembangunnya seperti class, attribute, objek,

DAFTAR PUSTAKA

anonim. (2017, maret 8 april). Class and Object : Object Oriented Programming using C++. *https://socs.binus.ac.id/*, hal. https://socs.binus.ac.id/2016/12/13/class-and-object-object-oriented-programming-using-c/.

anonim. (2020, april 8). Pemrograman berorientasi objek. *id.wikipedia.org*, hal. https://id.wikipedia.org/wiki/Pemrograman_berorientasi_objek.

jumaidi. (Tuesday, September 17, 2013, 2020 8 april). pengertian objek dan kelas pada PBO. *http://rpldanpbo.blogspot.com/*, hal. http://rpldanpbo.blogspot.com/2013/09/pengertian-objek-dan-kelas-pada-pbo.html.

times new roman

LAMPIRAN

```
Masukan Nama      : sugeng
Isi Buah Favorite : mangga


|##############################|
|Hai! sugeng
|Buah Favorit kamu adalah mangga
|##############################|
```

gambar . . .

```
|############################################################|
|Nama Buah Ini adalah = NANAS
|Rasa Buah NANAS adalah = Asam
|Buah NANAS Ini di Impor dari = Thailand

|############################################################|
|############################################################|
|Nama Buah Ini adalah = MANGGA
|Rasa Buah MANGGA adalah = Manis
|Buah MANGGA Ini di Impor dari = Kamboja

|############################################################|
|############################################################|
|Nama Buah Ini adalah = PISANG
|Rasa Buah PISANG adalah = Manis
|Buah PISANG Ini di Impor dari = Zimbabwe

|############################################################|
```