MODUL II
Kelas Dan Objek

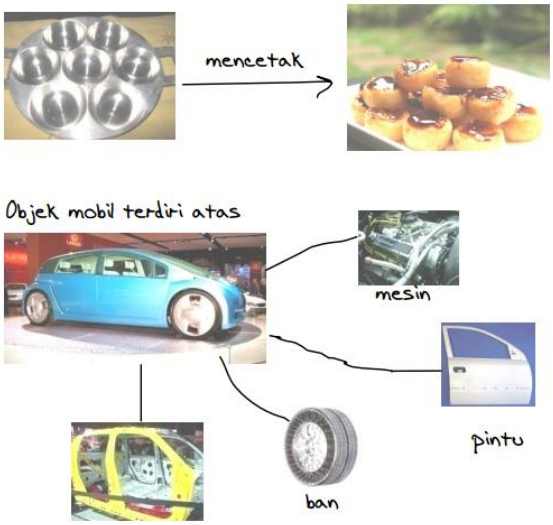
I. TUJUAN

a. Mahasiswa diharapkan mampu memahamai mengenai objek dan kelas
b. Mahasiswa diharapkan mampu menerjemahkan objek dan kelas dalam bahasa pemrograman

II. DASAR TEORI

a. Kelas Dan Objek

Objek adalah kesatuan entitas (benda), baik yang berwujud nyata ataupun hanya suatu sistem atau konsep yang memiliki sifat karakteristik dan fungsi. Dalam kehidupan sehari-hari kita sering menjumpai banyak objek dengan jenis yang sama, contohnya sepeda yang Anda miliki adalah salah satu jenis dari sepeda yang ada di dunia. Sepeda Anda adalah ***instance*** dari kelas yang disebut kelas sepeda.



Objek adalah suatu abstraksi dari suatu problem. Sedangkan **kelas** adalah kumpulan objek yang memiliki atribut dan perilaku yang sama (karakteristik sama). Setiap objek memiliki nilai atribut/state yang unik yang membedakannya dengan objek lain dari kelas yang sama, dan objek memliki perilaku/behaviour untuk mengakses atribut/state yang dimilikinya.

Contoh : kelas sepeda memiliki beberapa atribut/state (posisi gir, kecepatan, model, warna, dll) dan perilaku/behavior yang sama (mengubah gir, mengerem, dll). **Objek sepeda Anda** memliki nilai atribut yang berbeda bila dibandingkan dengan objek sepeda yang lain, tetapi sepeda Anda dan sepeda-sepeda yang lain memliki perilaku/behavior yang sama.

Kelas adalah pemodelan dari objek yang berisi informasi (aturan) tentang sifat karakteristik (data) dan tingkah laku (method) yang dimiliki oleh objek tersebut. Kelas dapat dianalogikan sebagai struktur data dari objek. Perbedaan kelas pada pemrograman berorientasi objek dengan struktur data pada pemrograman terstruktur adalah bahwa kelas pada pemrograman berorientasi objek tidak hanya berisi data saja tetapi juga fungsi-fungsi yang mengaksesnya, sehingga data dan fungsi harus dirancang secara bersamaan (**kelas = struktur data + fungsi**).

b. Mendefinisikan Kelas

Elemen-elemen dasar dalam mendefinisikan kelas :

1. Fields/variable

Field atau variable adalah implementasi dari atribut suatu objek. Field atau variable digunakan untuk menyimpan data dari objek.

Jenis Field atau variable :

- Instance variable

  Setiap objek memiliki salinan sendiri dan salinan tersebut memiliki nilai masing-masing.
- Class variable

  Suatu kelas hanya memiliki satu variable jenis ini dan digunakan bersama oleh semua objek dari kelas tersebut.

2. Constructor / Konstruktor

Konstruktor adalah method yang berfungsi untuk menginisialisasi variabel-variabel instance yang akan dimiliki oleh objek. Method konstruktor harus memiliki nama yang sama dengan nama kelas. Konstruktor ini dipanggil pada saat proses instansiasi kelas menjadi objek.

Kegunaan konstruktor :

- Mengalokasikan ruang bagi sebuah objek.
- Memberikan nilai awal terhadap anggota data suatu objek.
- Membentuk tugas-tugas umum lainnya.

Perlu diketahui :

- Konstruktor tidak mempunyai nilai balik(bahkan tanpa void).
- Konstruktor harus diletakkan pada bagian public.

Karakteristik constructor :

- Tidak pernah memiliki nilai balikan.
- Memiliki nama sama dengan nama kelas.

3. Methods

Method merupakan fungsi-fungsi implementasi perilaku objek untuk mengakses atribut-atributnya.

Sebagai ilustrasi, kita ingin membuat kelas **Buku**, yang memiliki objek diantaranya novel dan fiksi. Karena kelas merupakan abstraksi dari objek, maka pemilihan atribut haruslah yang dapat merepresentasikan objek secara umum. Beberapa atribut/property yang digunakan, yaitu judul dan pengarang yang bertipe char/string, kemudian jumlah (untuk mengetahui berapa banyaknya buku maka bertipe integer). Selain data yang telah didefinisikan sebelumnya,kita juga dapat menentukan method yang dimiliki oleh kelas tersebut diantaranya fungsi untuk mengisikan data dan menampilkan data.

III. GUIDED

a. Hubungan antara kelas, objek, atribut, dan method

Buku.cpp

```
#include<iostream.h>
#include<string.h>
#include <conio.h>
class Buku
{
     private :
           char judul[35];
           char pengarang[25];
           int  jumlah;
     public:
     void inisialisasiBuku(char *jdl,char *pngarang,int jmlh)
     {
           strcpy(judul, jdl);
           strcpy(pengarang, pngarang);
           jumlah = jmlh;
     }
     void infoBuku()
     {
           cout<<" Judul  :"<<judul<<endl;
           cout<<" Pengarang :"<<pengarang<<endl;
           cout<<" Jumlah buku :"<<jumlah<<endl;
     }
};

int main ()
{
     Buku novel,fiksi;
     novel. inisialisasiBuku
     ("Meriam Benteng  navarone","Alistair  Maclean",12);
     fiksi. inisialisasiBuku
     ("Jurassic park","Michael Crichton",3);
     novel. infoBuku();
     fiksi. infoBuku();
     getch();
     return 0;
}
```

ProjectBuku1

Buku.java

```
class Buku
{
    private String judul;
    private String pengarang;
    private int jumlah;

    public void setNilai(String judul,String pengarang,int
     jumlah)
    {
        this.judul  = judul;
        this.pengarang = pengarang;
        this.jumlah = jumlah;
    }
    void cetakKeLayar()
    {
     System.out.println("Judul : " + judul );
     System.out.println("Pengarang : " + pengarang );
     System.out.println("Jumlah : " + jumlah );
    }
}
```

Main.java

```
public class main
{
    public static void main(String[] args)
    {
       Buku a = new Buku();
       a.setNilai("Jurassic Park","Michael Crichton",21);
       a.cetakKeLayar();
    }
}
```

b. Konstruktor

Kompleks.cpp

```
#include <iostream.h>
#include <conio.h>
class Kompleks
{
     private :
        double re, im;
     public :
     Kompleks()
     {
         cout<<"Konstruktor Kompleks dijalankan...."<< endl;
         re = 5.2;
         im = 3.6;
     }
     Kompleks(double real,double imajiner)
     {
          cout<<"Konstruktor Kompleks dijalankan...."<< endl;
          re = real;
          im = imajiner;
      }
     void info()
     {
           cout<< "\nBilangan kompleks : "<< endl;
           cout<< "real = "<< re << endl;
           cout<< "imajiner = "<< im << endl;
           cout<< endl;
        }
};
int main()
{
      Kompleks a;
      a.info();
      Kompleks b(3.2, 4.1);
      b.info();
      getch();
      return 0;
}
```

ProjectBuku2

Buku.java

```
class Buku
{
    private String judul;
    private String pengarang;
    public Buku()
    {
          judul = "Tidak diketahui";
          pengarang = "Tidak diketahui";
    }
    public Buku(String judul,String pengarang)
    {
         System.out.println
         ("konstruktor buku sedang dijalankan...");
         this.judul  = judul;
         this.pengarang = pengarang;
    }
    void cetakKeLayar()
    {
         if(judul==null && pengarang==null)
         return;
         System.out.println("Judul : " + judul );
         System.out.println("Pengarang : " + pengarang );
    }
}
```

Main.java

```
public class main
{
     public static void main(String[] args)
     {
        Buku a,b = new Buku();
        a = new Buku(" Siaga Merah "," Alistair Maclean ");
        b = new Buku();
        a.cetakKeLayar();
        b.cetakKeLayar();
     }
}
```

IV. UNGUIDED

1. Buatlah program dalam bahasa **Java** dimana anda diminta untuk mengimplementasikan kasus operasi terhadap suatu Bank dalam konsep PBO.
   a. Buat class `Bank`
      - Buat konstruktor class Bank dengan parameter: `saldo` **(bobot 5 %)**
      - Buat method: `simpanUang`, `ambilUang`, dan `getSaldo` **(bobot 30 %)**
   c. Buat class `BankBeraksi`, tetapkan saldo awal lewat konstruktur Rp. 100000, jalankan 3 method di atas, dan tampilkan proses sebagai berikut : **(bobot 15 %)**

      Selamat Datang di Bank ABC
      Saldo saat ini: Rp. 100000

      Simpan uang: Rp. 500000
      Saldo saat ini: Rp. 600000

      Ambil uang: Rp. 150000
      Saldo saat ini: Rp. 450000

2. Buatlah program dalam bahasa **Java** dimana anda diminta untuk mengimplementasikan kasus operasi terhadap suatu Titik dalam konsep PBO.

   Buatlah kelas `Titik` dengan ketentuan sebagai berikut :
   a. Memiliki atribut `x` dan `y` **(bobot 5 %)**
   b. Memiliki *default* konstruktor **(bobot 10 %)**
   c. Memiliki *method* `inisialisasiTitik (int x, int y)`, untuk mengeset nilai `Titik` sesuai masukan dari user **(bobot 10 %)**
   d. Memilki *method* `tampilTitik()`, untuk menampilkan nilai `Titik` **(bobot 5 %)**
   e. Memilki *method* `perkalihanSkalar(int skalar)`, untuk mengalikan nilai `Titik` dengan suatu nilai skalar sesuai masukan dari user (x dan y dikalikan dengan nilai skalar) **(bobot 10%)**
   f. Memiliki *metod* `pencerminanSumbuX()`, untuk melalukan pencerminan `Titik` terhadap sumbu X (nilai y menjadi negatif) **(bobot 10%)**
   g. Memiliki *metod* `pencerminanSumbuY()`, untuk melalukan pencerminan `Titik` terhadap sumbu Y (nilai x menjadi negatif) **(bobot 10%)**
   h. Memiliki *method* `int jarak(Titik t2)`, untuk mencari jarak antara 2 buah titik, dengan rumus : `jarak` $=\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ **(bobot 10 %)**

i. Pada Menu Utama, buatlah menu untuk mengimplementasikan fungsi-fungsi dari kelas Titik diatas. **(bobot 30 %)**

   MENU OPERASI TITIK

   1. INISIALISASI TITIK

   2. TAMPIL TITIK

   3. PERKALIHAN SKALAR TITIK

   4. PENCERMINAN TERHADAP SUMBU X

   5. PENCERMINAN TERHADAP SUMBU Y

   6. JARAK ANTARA DUA TITIK

   0. KELUAR

   MASUKAN PILIHAN :

   Pada Menu JARAK ANTARA DUA TITIK, masukan dahulu koordinat `Titik` kedua

==========[Selamat Berlatih]==========