

Konsep Pemrograman Berorientasi Objek (OOP)

Kata Program dan Pemrograman mungkin sudah tidak asing didengar, tetapi apakah sebenarnya Program itu dan apa bedanya dengan Pemrograman ?

Program adalah suatu bentuk aplikasi atau bisa juga disebut *software* yang dibuat dengan menggunakan Bahasa Pemrograman dan dapat menyelesaikan suatu Tugas, Fungsi, Perhitungan dengan menggunakan komputer. Bahasa Pemrograman merupakan kumpulan perintah-perintah yang dimengerti komputer untuk melakukan tugas-tugas tertentu. Sedangkan Pemrograman atau *Programming* merupakan teknik untuk membuat suatu Program dengan menggunakan Bahasa Pemrograman.

Pemrograman Berorientasi Objek (*Object Oriented Programming*) merupakan teknik membuat suatu program berdasarkan objek. Untuk lebih memudahkan pemahaman tentang objek, saya beri sebuah contoh : Meja, Kursi, Jam Dinding dan yang lainnya merupakan contoh objek. Objek yang disebutkan tadi umumnya berada didalam sebuah ruangan, ruangan apabila kita melihatnya dari sebuah rumah bisa juga dikategorikan sebuah objek dan rumah jika dilihat dari cara pandang yang lebih luas lagi juga merupakan objek. Jadi sebuah Objek merupakan suatu bentuk atau model yang tergantung dari cara pandang objek dimaksud.

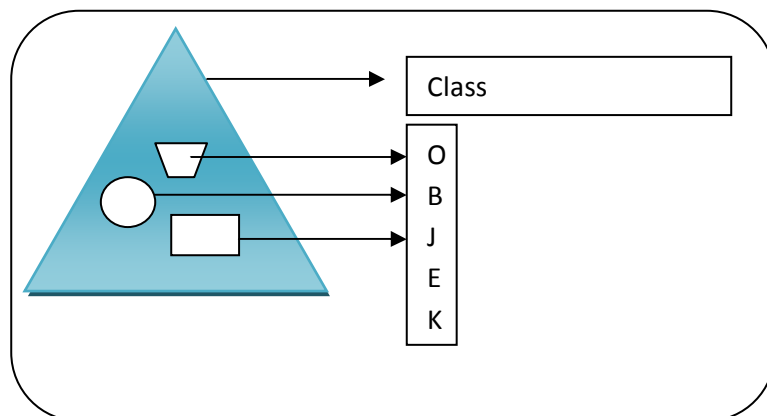
Meja, Kursi, Jam Dinding ada dalam sebuah Ruangan.

Ruang A, Ruang B, dan Ruang yg lain berada dalam sebuah Rumah.

Rumah 1, Rumah 2 dan Rumah yang lain juga berada didalam suatu tempat yang lebih luas yaitu Bumi, Dst.

Dan bukan tidak mungkin didalam Objek Jam Dinding terdapat objek-objek yang lain yang lebih kecil misalnya Batterai, Gear, dll.

Cara pandang kita dari sebuah Ruangan, Rumah atau Bumi merupakan **Class** apabila kita melihatnya dalam Pemrograman.



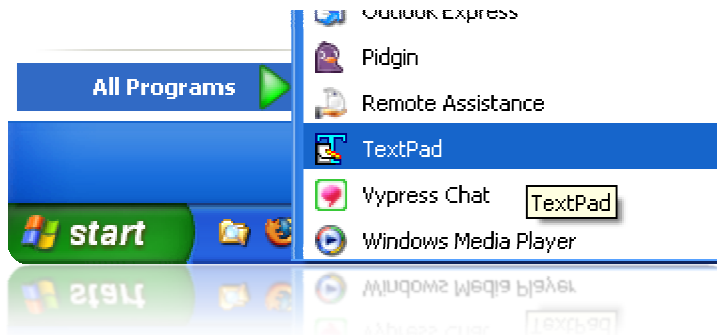
Gambar 1. Class dan Objek

Setiap objek memiliki **Atribut** dan **Method**. Atribut adalah segala sesuatu yang berhubungan dengan karakteristik objek. Seperti Tipe, Model, Pintunya, Kusennya menggunakan kayu apa?, Warna Catnya, merupakan salah satu atribut dari Rumah. Sedangkan Method dari Rumah adalah dapat melindungi dari panas, hujan, angin. Jadi Method merupakan Fungsi atau segala sesuatu yang dapat dilakukan Objek.

Tanpa harus berteori lagi, kita coba langsung pemahaman tentang Objek dalam Pemrograman dan untuk Bahasa Pemrogramannya saya menggunakan **Java**. Mengapa Java? Karena pengelolaan Objek didalam Java lebih mudah dipahami.

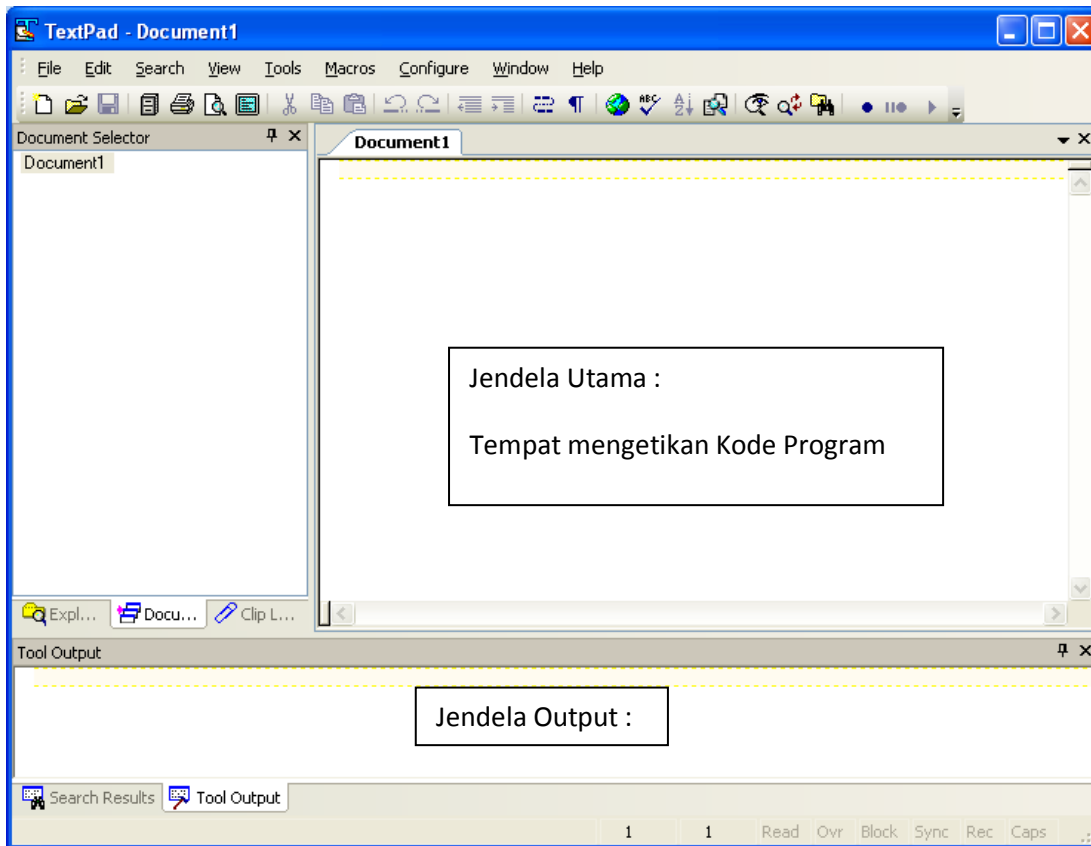
Asumsi saya, alat pendukung (*Tools*) pemrograman java sudah terinstall dalam komputer anda. Apabila belum anda dapat men-*download*-nya, yang diperlukan adalah *Compiler* java (<http://java.sun.com/j2se/1.4.2/download.html>) dan sebuah teks editor (<http://www.textpad.com>) atau dapat menggunakan Notepad pada Windows.

1. Jalankan TextPad pada komputer anda



Gambar 2. Menjalankan TextPad pada menu Windows

Tampilan TextPad seperti gambar berikut,

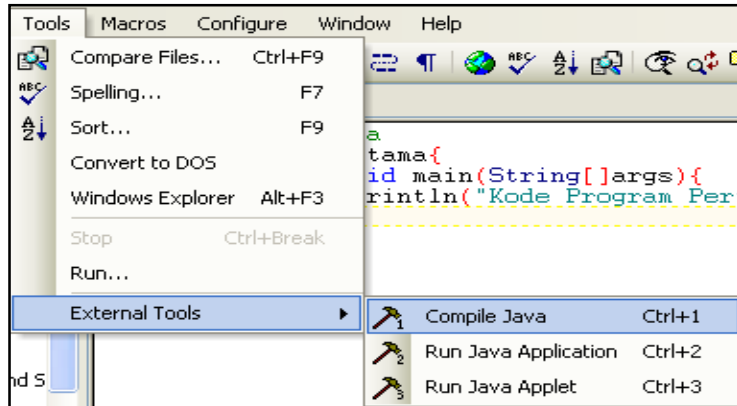


Gambar 3. Tampilan TextPad

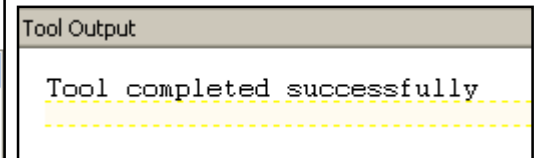
2. Ketiklah kode Program berikut ini, pada jendela utama TextPad. (“Perhatikan penulisannya !”)

```
//Latihan pertama
public class Pertama
{
    public static void main(String[]args)
    {
        System.out.println("Kode Program Pertamaku");
    }
}
```

3. Buatlah sebuah Folder (mis: ContohJava) di C: (“ boleh juga selain C: “)
4. Simpanlah Kode Program diatas pada Folder yang telah dibuat, beri nama file : “Pertama.java” tanpa tanda petik (“Perhatikan penulisan nama filenya!”).
5. Lakukan Kompilasi (*Compile*) kode program yang telah dibuat dengan mengklik Menu **Tools, External Tools, Compile Java**. Atau dengan menekan Tombol **CTRL + 1**. (Gambar 4)
Jika tidak ada yang salah, pada Jendela Output akan ditampilkan pesan “**Tools Completed Successfully**” (Gambar 5)

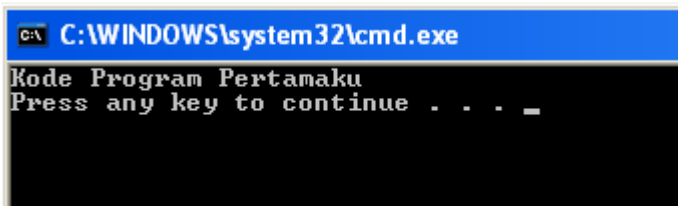


Gambar 4. Menu untuk mengkompilasi Kode Program



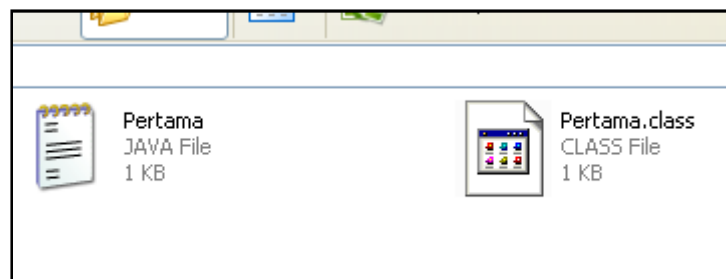
Gambar 5. Keterangan Sukses

6. Setelah dikompilasi, jalankan program dengan mengklik Menu **Tools, External Tools, Run Java Applications** atau dengan menekan tombol **CTRL + 2**.



Gambar 6. Hasil program setelah dijalankan

7. Sekarang, lihatlah pada folder "ContohJava" terdapat 2 buah file :
- Pertama.java
 - Pertama.class



Gambar 7. Tampilan pada Folder "ContohJava"

Penjelasan :

Kita mulai dari hasil kompilasi kode program yang membuat 1 (satu) file dengan nama "Pertama.class" file ini adalah hasil dari baris perintah

```
public class Pertama{
```

Baris perintah ini adalah untuk membuat sebuah *class* dengan nama "Pertama", kata kunci **public** menandakan bahwa *class* ini memiliki jangkauan publik atau global artinya *class* "Pertama" ini dapat digunakan oleh *class* lain.

Kalau ada *class* seharusnya ada objeknya *dong*, lalu yang mana objeknya ?

Mungkin pertanyaan anda adalah seperti itu, baik kita lanjutkan penjelasannya.

Coba perhatikan baris perintah

```
public static void main(String[] args)
{
    System.out.println("Kode Program Pertamaku");
}
```

Masih ingat penjelasan diatas tentang Objek yang memiliki Atribut dan Method, pada baris

```
public static void main(String[] args)
```

inilah objeknya.

Mungkin pertanyaan anda adalah, kalau baris ini adalah objeknya lalu nama objeknya apa, terus atribut dan methodnya mana?

Pada contoh program ini saya belum mendefinisikan sebuah nama objek dengan jelas karena programnya masih sangat sederhana, untuk contoh ini “main” adalah nama objeknya berikutnya yang ada didalam

```
(String[] args)
```

Ini adalah Atribut dari objek “main” dan terakhir Methodnya ada pada baris

```
System.out.println("Kode Program Pertamaku");
```

Pada baris ini objek “main” dengan atributnya “String[] args” melakukan sebuah *method* yaitu menampilkan teks/tulisan “Kode Program Pertamaku” pada layar.

Baiklah, sekarang coba ketikan lagi kode program berikut : (“perhatikan penulisannya!”)

```
//Latihan kedua
public class Rumah //Buat class Rumah
{
    //buat method static Meja
    public static void Meja()
    {
        System.out.println("Ada Meja");
    }

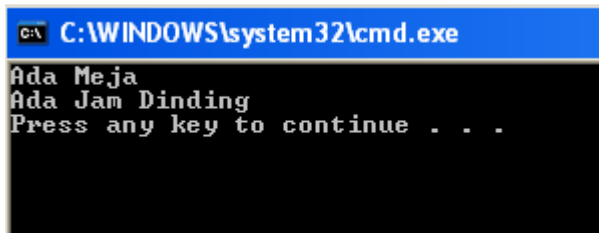
    //method jam dinding
    void JamDinding()
    {
        System.out.println("Ada Jam Dinding");
    }
}
```

- Simpan kode program diatas dengan nama file “Rumah.java”
- Lakukan kompilasi (*compile*) pada kode program, jika tidak ada pesan *error*, berikutnya
- Buat lagi satu file baru, ketikkan kode program dibawah ini kemudian simpan file dengan nama “IsiRumah.java”
- Lakukan *compile* pada kode program dan lihatlah hasilnya!

```
//Latihan kedua
public class IsiRumah
{
    public static void main(String[]args)
    {
        Rumah.Meja();

        Rumah Panggil = new Rumah();

        Panggil.JamDinding();
    }
}
```



```
C:\WINDOWS\system32\cmd.exe
Ada Meja
Ada Jam Dinding
Press any key to continue . . .
```

Penjelasan :

Pada latihan kedua ini kita telah membuat 2 (dua) buah file, yaitu :

1. Rumah.java → membuat sebuah *class* dengan nama “Rumah.class”
2. IsiRumah.java → *class* yang lain “IsiRumah.class” yang menggunakan *class* Rumah

Pada **file pertama** “Rumah.java” diawali dengan membuat sebuah *class public* “Rumah” agar dapat digunakan pada *class* yang lain, didalam *class* “Rumah” terdapat 2 (dua) buah *method* yaitu “Meja” dan “JamDinding”, perhatikan dari cara penulisannya!

Method “Meja”

```
public static void Meja()
{
    System.out.println("Ada Meja");
}
```

Kata kunci “public static” digunakan agar method ini mempunyai jangkauan global dan isinya tetap, sedangkan

Method “JamDinding” hanya digunakan untuk jangkauan lokal saja

```
void JamDinding()  
{  
    System.out.println("Ada Jam Dinding");  
}
```

Pada file **yang kedua** “IsiRumah.java” diawali juga dengan membuat sebuah *class public* “IsiRumah” didalamnya terdapat method utama “main” dengan Atribut yang digunakan “String[]args” fungsi kode program ini adalah memanggil *class* “Rumah” untuk digunakan pada *class* “IsiRumah”, perhatikan cara pemanggilan yang berbeda dari Method yang ada pada *class* “Rumah”!

```
Rumah.Meja();
```

Pada baris ini, karena Method “Meja” menggunakan kata kunci “public static” maka pemanggilannya dapat secara langsung tanpa harus membuat satu objek baru untuk dapat menggunakan/memanggil *method* pada *class* lain, seperti method “JamDinding” pada *class* “Rumah”

```
Rumah Panggil = new Rumah();
```

Pada baris ini dibuat satu objek baru dengan nama “Panggil” yang merupakan nama lain dari *class* “Rumah”, cara pemanggilnya adalah seperti pada baris berikut

```
Panggil.JamDinding();
```

Kesimpulannya adalah pengelolaan objek dalam Java lebih mudah untuk dipelajari, tentunya harapan saya juga tulisan ini dapat membantu bagi siapa saja yang ingin memulai untuk belajar Pemrograman dengan Java, lebih lanjut tentang konsep pemrograman berorientasi objek ini akan dibahas pada sesi selanjutnya, jadi... tetap link di blog <http://asepramdhani.wordpress.com>

Terima Kasih.

Referensi :

1. <http://id.wikipedia.org/>
2. Jeni-dasarpemrogramanjava.pdf
3. Codenhead-javabasic.pdf