Pertemuan 2 (2) :

Membuat Class dan mengakses Anggota Class

Tessy Badriyah, SKom. MT.

http://lecturer.eepis-its.edu/~tessy

Pendahuluan

- Pada bagian ini, kita akan membuat program berbasis obyek menggunakan bahasa java
- Sebelum itu, hal yang pertama dilakukan adalah mempelajari terlebih dahulu tentang pembuatan class.

Pengertian Class

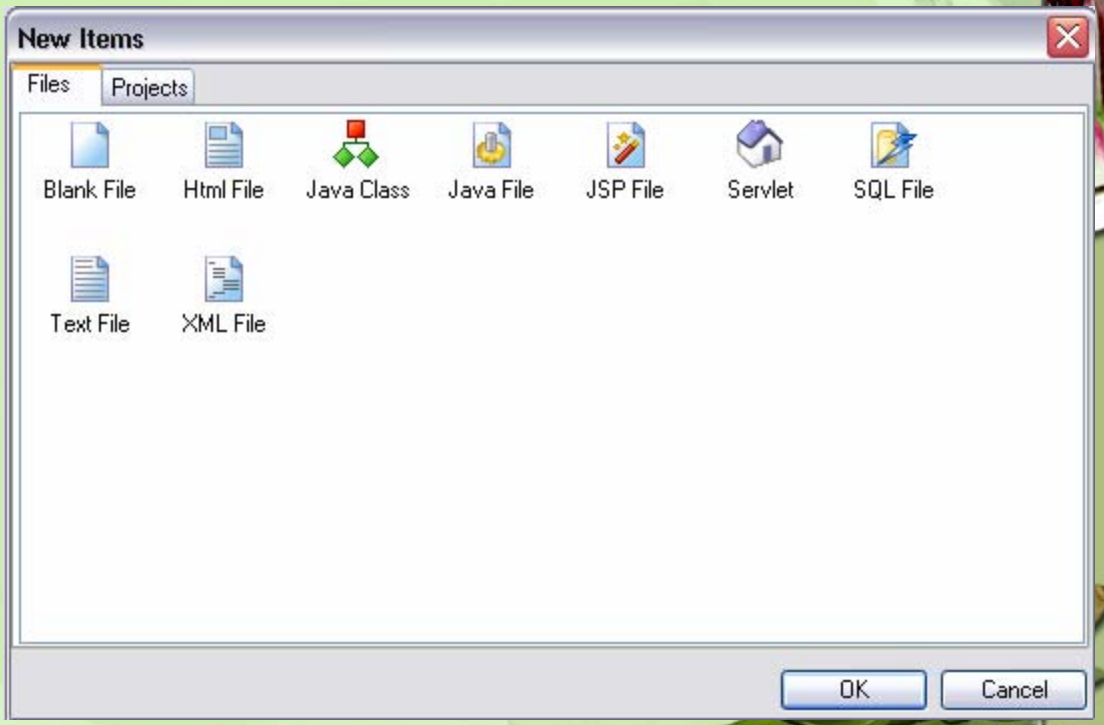
- Review : (konsep PBO)
- Class adalah template untuk pembuatan obyek
- Class memiliki anggota :
  - Atribut
  - Method

Pembuatan Class

- Kita akan membuat Class dengan nama *Lampu*
- Class *Lampu* memiliki atribut :
  - *status* => 0 atau 1
  - *merek* => ‘phillips’ atau ‘national’
- Class *Lampu* memiliki method :
  - *lampuDinyalakan();*
  - *lampuDimatikan();*

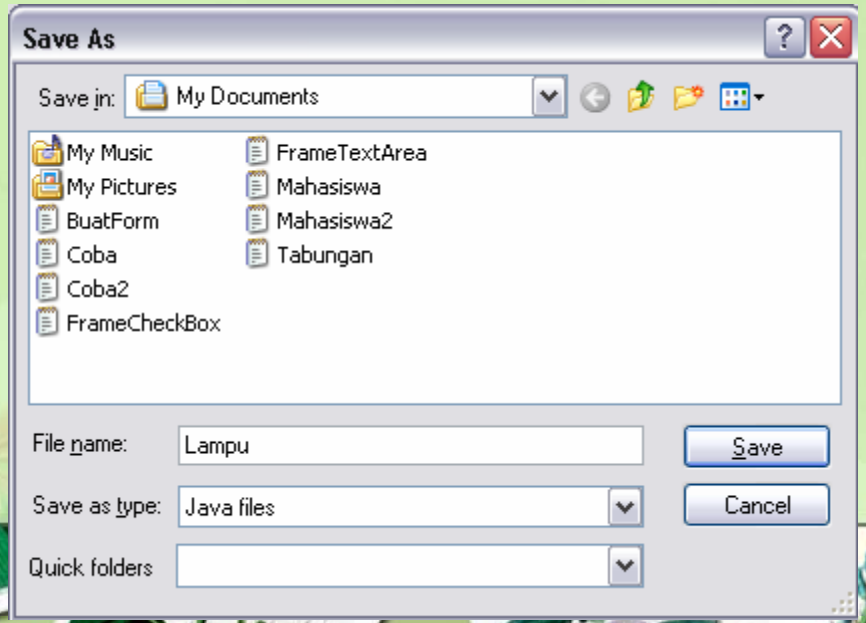
Membuat Class di Java

- Jalankan Gel.
- Kemudian pilih menu *File => New*
- Akan muncul kotak dialog berikut :
- Pilih *Java File*



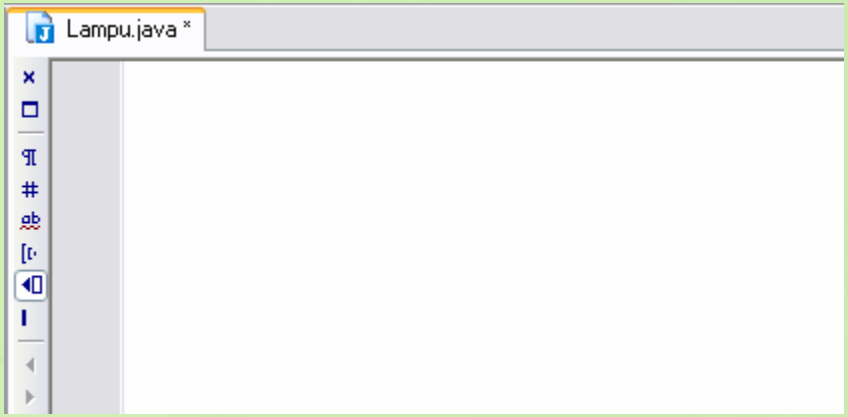
Membuat Class di Java

- Kemudian anda diminta untuk memasukkan nama file (otomatis akan diberi ekstensi .java) tempat menyimpan class yang akan dibuat.
- Beri nama yang sama dengan nama Class yang akan dibuat, lalu tekan tombol Save



Pembuatan Class pada Editor Gel

- Berikutnya, ketikkan listing program untuk pembuatan Class pada editor Gel berikut ini :



- Bagaimana cara penulisan class ?
- Baca pada slide berikut :

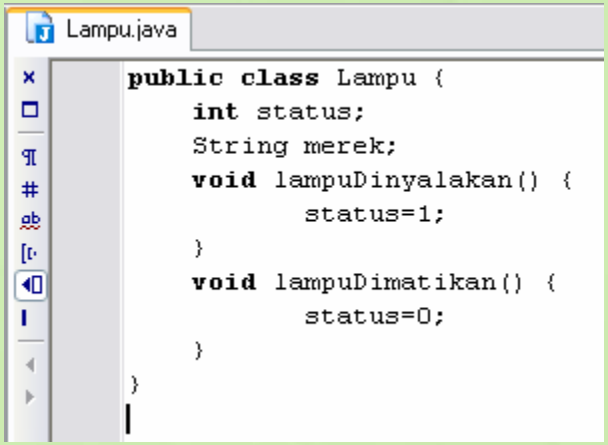
Cara Penulisan : CLASS

- Deklarasi class dapat dilakukan dengan sintak :
  <modifier> class <namaclass> {
  [deklarasi atribut]
  [deklarasi method]
- Deklarasi atribut sebagai berikut :
  <modifier> <tipe> <nama atribut> ;
- Deklarasi method dapat dilakukan dengan cara :
  <modifier> <return type> <nama_method> ([daftar argumen])
  {

  }

Pembuatan Class Lampu

```
public class Lampu {
    int status;
    String merek;
    void lampuDinyalakan() {
        status=1;
    }
    void lampuDimatikan() {
        status=0;
    }
}
```



Perhatikan : hanya ada satu class dengan modifier public. Dengan nama class yang sama dengan nama file tempat listing program disimpan.

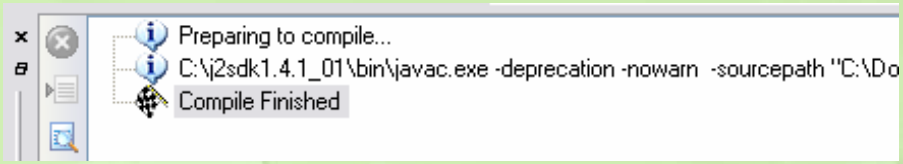
Compile Class
(namafile.JAVA => namafile.CLASS)

- Setelah class dibuat, agar dapat digunakan, class tersebut harus dicompile.
- Hasil dari proses compile :

  namafile.JAVA => namafile.CLASS

Compile Class pada Gel

- Untuk mengcompile Class pada Gel, pilih menu *Build => Compile File*
- Atau :
- Tekan icon 
- Jika berhasil dicompile dengan sukses pesannya :



- Periksa apakah telah terbentuk Lampu.CLASS

Meng-create Obyek dari suatu Class

- Sesuai definisi awal, class adalah template untuk membuat obyek
- Sekarang kita akan membuat sebuah obyek yang berasal dari class Lampu
- Nama obyeknya misal : *lampuku*
- Perhatikan cara penulisan untuk pembuatan obyek pada slide berikut :

Membuat Obyek dari Class

- Cara penulisan :

  namaClass namaObyek=new namaClass();

- Membuat obyek *lampuku* dari Class *Lampu*

  Lampu lampuku = new Lampu();

Dimana kita meletakkan obyek ?

- Obyek dapat diletakkan di class yang lain.
- Class yang lain ini bisa ditempatkan pada file yang sama dengan nama Class

Lampu.java *

```
public class Lampu {
    int status;
    String merek;
    void lampuDinyalakan() {
            status=1;
    }
    void lampuDimatikan() {
            status=0;
    }
}
class Lain {
      Lampu lampuku = new Lampu();
}
```

Dimana kita meletakkan obyek ?

- Atau obyek diletakkan di class lain dan dalam file yang berbeda
- (ingat) : nama file **sama dengan** nama class, dan hanya satu nama class yang memiliki modifier public

Lampu.java | Lain.java

```
public class Lampu {
    int status;
    String merek;
    void lampuDinyalakan() {
        status=1;
    }
    void lampuDimatikan() {
        status=0;
    }
}
```

Lampu.java | Lain.java

```
public class Lain {
    Lampu lampuku = new Lampu();
}
```

main method dalam JAVA

- Main method dalam JAVA, tempat kita menempatkan obyek, menampilkan suatu hasil ke layar, dsb adalah :

```
public static void main(String [] args) {
    ……..
}
```

- Seharusnya disinilah kita menempatkan obyek *lampuku* yang dibuat dari class Lampu
- Setelah obyek ditempatkan dalam main method, maka kita bisa memanggil method dari class Lampu (*lampuku.lampuDinyalakan();*)
- Untuk menampilkan pada layar menggunakan :

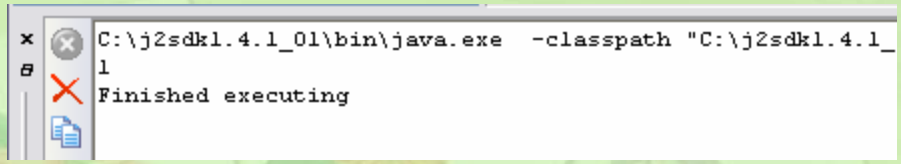
System.out.println(……..);

Lampu.java | Lain.java

```
public class Lain {
    public static void main(String [] args) {
        Lampu lampuku = new Lampu();
        lampuku.lampuDinyalakan();
        System.out.println(lampuku.status);
    }
}
```

Menjalankan program Java yang ada main method

- Dan Class yang berisi main method ini satu-satunya Class yang bisa dijalankan
- Pilih menu *Build* => *Run File* atau klik icon 
- Setelah menjalankan method *lampu*
- Hasilnya akan ditampilkan nilai dari variabel status



@Tessy Badriyah,SKom. MT.

Percobaan

Percobaan 1

- Dibuat program untuk menampilkan tulisan
- Nama program : CobaTulis.java
- Dalam program ada class Tulis yang berisi method Tulisan untuk menampilkan data di layar
- Dalam program utama diciptakan obyek baru bernama tulisanku yang berasal dari class Tulis
- Kemudian dengan menggunakan obyek tulisanku dipanggil method Tulisan untuk menampilkan data pada layar

Listing Program CobaTulis.java

```
public class CobaTulis {
    public static void main(String [] args) {
    Tulis tulisanku = new Tulis();
    tulisanku.Tulisan();
    }
}
class Tulis {
    void tulisan() {
        System.out.println("Saya belajar Java");
    }
}
```

Menjalankan program

- Listing program

CobaTulis.java

```
public class CobaTulis {
    public static void main(String [] args) {
    Tulis tulisanku = new Tulis();
    tulisanku.Tulisan();
    }
}
class Tulis {
    void Tulisan() {
        System.out.println("Saya belajar Java");
    }
}
```

- Hasil :

```
C:\j2sdk1.4.1_01\bin\java.exe  -classpath "C:\j2sdk1.4.1_01\jre\
Saya belajar Java
Finished executing
```

Percobaan 2

- Pada percobaan 2 ini dibuat program yang hasilnya sama dengan percobaan 1 sebelumnya
- Akan tetapi dibuat dari dua file terpisah, dengan nama **Tulis.java** dan **TesTulis.java**
- Pada **TesTulis.java** diciptakan obyek baru yang akan memanggil method yang ada pada **Tulis.java**

Listing program Tulis.java

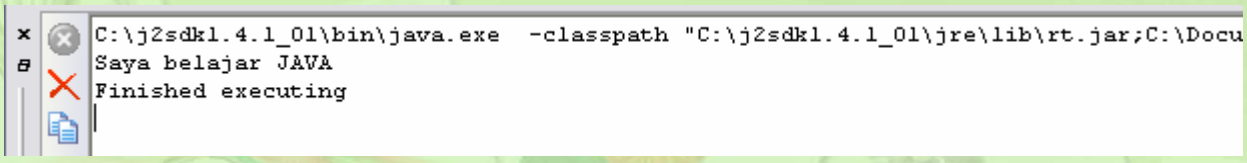
```
public class Tulis {
    void Tulisan() {
        System.out.println("Saya belajar JAVA");
    }
}
```

Listing program TesTulis.java

```
public class TesTulis {
    public static void main(String [] args) {
      Tulis tulisanku = new Tulis();
      tulisanku.Tulisan();
    }
}
```

Menjalankan program

- Untuk menjalankan program, pertama Tulis.java dicompile saja tanpa di-run, ini dikarenakan Tulis.java merupakan class yang tidak berisi program utama (public static void main)
- Berikutnya, TesTulis.java dicompile lalu dijalankan.
- Hasil running program :



Percobaan 3

- Program berikut mengimplementasikan class Mobil
- Class Mobil memiliki atribut : aktifitas (parkir atau jalan-jalan), warna, kecepatan
- Class Mobil memiliki method :
  - cekKecepatan
    jika kecepatan=0, maka aktifitas=parkir
  - cetakAtribut
    mencetak semua nilai atribut

Listing program

```
class Mobil {
  String aktifitas;
  String warna;
  int kecepatan;
  void cekKecepatan() {
    if (kecepatan==0)
      aktifitas="parkir";
  }
  void cetakAtribut() {
    System.out.println("Aktifitas    = "+aktifitas);
    System.out.println("warna       = "+warna);
    System.out.println("Kecepatan    = "+kecepatan);
  }
  public static void main(String [] args) {
    Mobil mobilku = new Mobil();
    mobilku.kecepatan=0;
    mobilku.warna="merah";
    mobilku.cekKecepatan();
    mobilku.cetakAtribut();
   }
}
```

Latihan 1

- Program berikut ini jika di-compile terdapat pesan error, betulkan kesalahannya !

a)

Salah1.java *

```
public Class Salah1 {
    System.out.println("Saya belajar JAVA");
}
```

b)

Salah1.java *

```
public Class Salah1 {

}
public Class Tampil {
   public static void main(String [] args) {
      System.out.println("Saya belajar JAVA");
   }
}
```

Latihan 2

- Buat Class Konversi yang anggotanya :
  - Atribut :
    - jarak *(dalam meter)*
  - Method :
    - meterKekilo();
    - kiloKemeter();
- Buat Class TesKonversi yang isinya pembuatan obyek dan pengaksesan anggota dari Class Konversi