PERTEMUAN 2

PEMOGRAMAN BERORIENTASI OBJEK

L/O/G/O

PENGERTIAN

- Secara logika kelas dalam dunia pemrograman dapat kita bayangkan seperti halnya kelas-kelas yang ada pada sekolah dasar. Kelas digunakan untuk mengelompokan komponen-komponen dengan kriteria tertentu yang hampir sama dimiliki semua anggota kelas. Misalnya kelas 1 pada sekolah dasar adalah anak-anak yang memiliki usia 6 sampai 7 tahun
- Dalam dunia pemrograman biasanya digunakan untuk mengelompokan benda dan proses yang terkait dengan aplikasi komputer yang akan dibuat (benda dan proses). Oleh karena itu sebelum merancang kelas-kelas apa saja yang dibutuhkan perlu dipelajari, bagaimana merancang data mengenai benda-benda yang terkait dan juga proses –proses yang terkait pada aplikasi komputer yang akan dibuat

KELAS DAN OBJEK

Kelas

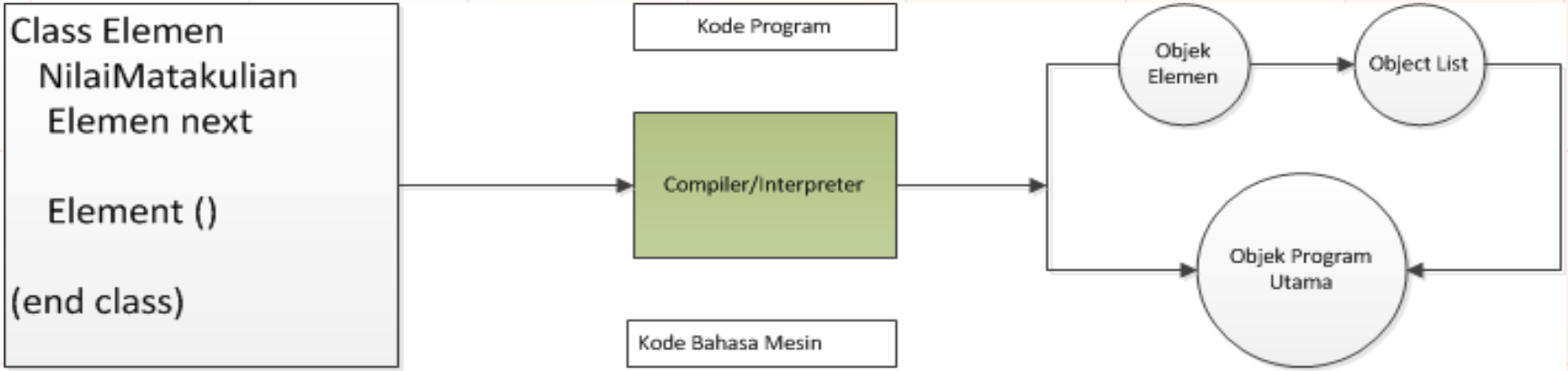
- Kode program kelas yang ditulis oleh programmer
- Sebuah file yang berisi kode program

Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat runtime (saat sebuah program dieksekusi), yang kita punya adalah objek dan apabila berada didalam teks program yang kita lihat hanyalah kelas

Objek

- Kelas yang telah di eksekusi dan dijalankan menjadi program komputer yang sudah siap digunakan
- Tidak dapat dilihat bentuknya karena onjek dijalankan didalam proses komputer mengeksekusi program berorientasikan objek
- Onjek adalah elemen pada aat runtime yang akan diciptakan, dimanipulasi dan dihancurkan saat eksekusi sehingga sebuah objek hanya ada saat sebuah program dieksekusi

KELAS DAN OBJEK



Ilustrasi jalannya kelas menjadi objek

PHP, C++ dan JAVA

- C++ merupakan bahasa pengembangan dari bahasa C yang berbasiskan prosedural untuk mendukung pemrograman berorientasi objek.
- PHP pada awalnya berbasiskan pemrograman prosedural atau menggunakan fungsi-fungsi dalam membuat membuat kode programnya. Saat ini PHP5 telah mendukung pemrograman berorientasi objek seperti halnya C++
- PHP bukanlah skrip pemrograman berorientasi objek murmi seperti Java, tetapi lebih bersifat semi berorientasi objek seperti halnya C++.
- PHP masih menggunakan cara prosedural pada program utamannya (main program)

PHP, C++ dan JAVA

- Sebuah file dapat terdiri dari beberapa kelas, namun biasanya pada Java, sebuah file hanya terdiri dari satu kela yang disimpan dengan nama kelas, misalnya file List.java berisi kelas List, namun jika kelas yang dibuat misalkan (dalam bahasa pemrograman Java):

  Public class nama_kelas{

  ……..

  }

maka kelas itu harus disimpan dalamsatu file hanya untuk satu kelas.

- Setelah dikompilasi maka pada Java akan ada sebuah file .class yang berisi bytecode dari setiap kelas
- Jika sebuah file terdiri dari dua kelas maka setelah dikompilasi akan dihasilkan dua buah file .class yang nantinya akan dibaca oleh interpreter Java pada saat program dieksekusi.
- Setiap kelas sebaiknya memang disimpan dalam satu file agar mudah diambil jika akan digunakan pada program lain yang membuuhkan kelas itu.

PHP, C++ dan JAVA

- Kelas didefinisikan sebagai file yang berbeda dengan program utama (program utama memiliki file terpisah dengan file kelas).
- Aplikasi penjadwalan penerbangan pesawat, maka jika diperlukan sebuah aplikasi pencarian daftar nama pesawat yang telah dibuat sebuah pabrik maka kelas pesawat dapat digunakan untuk membuat aplikasi tanpa harus membuat kode program dari awal (karena data yang dimiliki pesawat pada aplikasi pertama dan kedua sama)
- Java merupakan bahasa pemrograman berorientasi objek karena semua kode program di dalam bahasa Java dibungkus di dalam kelas

Ilustrasi Pemrograman Berorientasi Objek

PHP, C++ dan JAVA

class Halo	Mendeklarasikan kelas bernama Halo
private kata : string	Mendeklarasikan sebuah atribut bernama kata bertipe string dan bersifat provat (tidak dapat diakses langsung oleh kelas lain)
procedure main() kata <- "Halo Dunia" output (kata) {end procedure} {end class}	Mendeklarasikan sebuah prosedur utama untuk menampilkan kata "Halo Dunia"

Ilustrasi Pembuatan Kelas

PHP, C++ dan JAVA

- Pada C++, prosedur utama tidak dapat dimasukan ke dalam kelas, tetapi harus diprogram secara prosedural (khusus prosedur utama) dan setiap kelas memiliki konstruktor. Konstruktur adalah sebuah metode dan dalam pemrograman berorientasi objek, metode yaitu prosedur dan fungsi
- Pada pemrograman PHP, fungsi dan prosedur sama-sama dianggap sebagai fungsi saja (function) maka metode yang ada didalam kelas semuanya berupa fungsi.
- Pada Java, program utama dapat juga dibungkus dalam sebuah kelas. Java merupakan bahasa pemrograman beorientasi objek murni sehingga program utamanya pun dibungkus didalam kelas.

ATRIBUT

- Atribut dari sebuah kelas adalah variable global yang dimiliki sebuah kelas, misalkan :

```
class Buku
  private judul : string
  private pengarang : string
  Buku ()
   {end constructor}
 {end class}
```

- Maka judul adalah atribut dari kelas Buku yang bertipe string dan pengarang adalah atribut dari kelas Buku yang bertipe string juga.
- Atribut pada sebuah kelas memiliki izin akses jika kelas digunakan oleh kelas lain, izin akses itu seperti provate, public dan protected.

ATRIBUT

Atribut Private

- Izin akses private pada sebuah atribut biasanya digunakan oleh sebuah kelas untuk melindungi atribut-atributnya agar tidak dapat diakses oleh kelas lain
- Sebuah atribut yang dinyatakan private hanya dapat diakses secara langsung oleh kelas yang membungkusnya sedangkan kelas lainnya tidak dapat mengakses atribut ini secara langsung
- Sebuah atribut umumnya mempunyai ijin akses private agar tidak dapat diakses secara langsung oleh kelas lain
- Atribut merupakan bagian kelas yang harus dilindungi dalam enkapsulasi kelas. Oleh karena itu, akses kelas lain terhadap atribut suatu kelas harus dibatasi. Maka untuk mengubah atau mengambil nilai atribut kelas lain sebaiknya menggunakan metode set and get untuk setiap atribut yang dimiliki sebuah kelas.

ATRIBUT

```
class Buku
    private judul : string
    private pengarang : string
    Buku ()
     {end constructor}
{end class}
```

- Yang dapat mengakses judul dan pengarang hanyalah kelas buku sehingga jika sebuah kelas lain di dalamnya mempunyai kode

  Buku b <- new Buku(),

  maka pengaksesan b.judul tidak diizinkan pada kelas lain yang menggunakan kelas buku.
- Agar isi dari sebuah atribut private dapat diakses oleh kelas lain dapat dibuat sebuah metode yang mengembalikan nilai atribut tersebut. Misalkan

  ```
  public getJudul() → string
     -> judul
  {end getJudul}
  ```

  Sehingga kelas lain akan mengakses atribut judul pada kelas Buku dengan kode string j <- b.judul

ATRIBUT

Atribut Public

- Izin akses public sebuah atribut biasanya digunakan oleh sebuah kelas jika sebuah atribut diperbolehkan diakses secara langsung oleh kelas lain.
- Sebuah atribut yang dinyatakan sebagai public dapat diakses secara langsung oleh kelas lain diluar kelas yang membungkusnya, misalkan pada kelas Buku

```
class Buku
    public judul : string
    public pengarang : string
    Buku ()
     {end constructor}
{end class}
```

maka atribut judul dan pengarang dapat diakses langsung oleh kelas lain misalkan dengan kode

```
Buku b <- new Buku ()
Output (b.judul)
```

ATRIBUT

Atribut Protected

- Izin akses protected sebuah atribut biasanya digunakan oleh sebuah kelas jika sebuah atribut diperbolehkan diakses secara langsung oleh kelas lain yang merupakan kelas turunannya (inheritance)
- Izin akses protected dimaksudkan untuk melindungi atribut agar tidak diakses secara langsung oleh sembarang kelas lain namun diizinkan diakses secara langsung oleh kelas turunannya
- Sebuah atribut yang dinyatakan sebagai protected tidak dapat diakses secara langsung oleh kelas lain diluar kelas yang membungkusnya kecuali kelas yang mengaksesnya adalah kelas turunan dari kelas yang membungkusnya

```
class Buku
    protected judul : string
    protected pengarang : string
    Buku ()
     {end constructor}
{end class}
```

KONSTRUKTOR

- Sebuah metode (method) yang dimiliki oleh sebuah kelas
- Nama sebuah konstruktor harus sama dengan nama dari sebuah kelas misalkan kelas Buku maka konstruktornya adalah Buku()
- Sebuah konstruktor juga bisa menerima masukan seperti halnya prosedur pada pemrograman prosedural.
- Konstruktor harus bersifat public karena sebuah konstruktor akan diakses oleh kelas lain untuk membuat objek suatu kelas
- Sebuah kelas dapat memiliki konstruktor lebih dari satu.
- Pada saat eksekusi program, compiler atau intrepeter akan mencari konstruktor mana yang sesuai dengan konstruktor yang dipanggil. Hal ini disebut sebagai overloading

KONSTRUKTOR

Fungsi dari Konstruktor

- Mengalokasikan sebuah objek saat program dieksekusi (memerintahkan dibuatnya alokasi objek di memori saat program dijalankan)
- Memberikan nilai awal sebagai inisialisasi dari semua atribut yang perlu diinisialisasi
- Mengerjakan proses- proses yang diperlukan saat sebuah objek dibuat

KONSTRUKTOR

```
class Titik
   private x  : integer
   private y  : integer
   Titik ()
      x <- 0
      y <- 0
   {end Titik}
      Titik(xp : integer, yp : integer)
          x <- xp
          y <- yp
       {end Titik}
{end class}
```

Maka ketika objek Titik dibuat dengan kode :

```
Titik t <- new Titik()
```

Konstruktor yang akan dipanggil adalah Titik(), namun jika objek Titik dibuat dengan kode :

```
Titik t <- new Titik(5,7)
```

Maka konstruktor yang dipangil adalah Titik (xp : integer, yp : integer)

KONSTRUKTOR

- Pada C++, konstruktor diimplementasikan sebagai metode yang bernama sama dengan kelasnya.

```
class Buku{
        private :
 ……….
        public :
        Buku () {
       //konstruktor
 }
        Buku (char j[], char p[] {
        //konstruktor
 }
 ……….
}
```

Pada Java, konstruktor diimplementasikan sama dengan C++ yaitu sebagai metode yang bernama sama dengan kelasnya.

```
Class Buku{
 ……….
 Buku() {
   //konstruktor
 }
 Buku (string j, string p) {
   //konstruktor
 }
 ……..
}
```

KONSTRUKTOR

- Pada PHP4, konstruktor diimplementasikan sebagai fungsi yang bernama sama dengan kelasnya.

```
class Buku{
   ……….
   function Buku () {
         //konstruktor
    }
    function constructor ($j=“” , $p=“”) {
         //konstruktor
    }
     ……..
}
```

Pada PHP5, konstruktor diimplementasikan dengan menggunakan metode _construct().

```
class Buku{
     ……..
     function _construct () {
        //konstruktor
     }
     function_construct  ($j=“” , $p=“”)) {
        //konstruktor
     }
     ……..
}
```

DESTRUKTOR

- Metode yang dipanggil secara otomatis ketika objek dihancurkan
- Destruktor tidak perlu harus ada pada kode program sebuah kelas jika compiler atau interpreter tidak memiliki garbage collection (mekanisme membersihkan alokasi objek-objek yang sudah tidak dipakai dalam memori)
- Sebuah destruktor biasanya hanya terdapat satu metode pada sebuah kelas dan tidak memiliki parameter masukan
- Destruktor biasanya berisi proses=proses finalisasi dari suatu kelas, misalnya menghapus alokasi meori yang terlah dipakan oleh objek
- Penamaan Destruktor sama dengan konstruktor, hanya saja destruktor diberi tanda ~ pada bagian penamaan metode

```
class Titik
   private x : integer
   private y : integer
       Titik() {
          x <- 0
          y <- 0
        }
       ~Titik()
       {end ~Titik}
  }
```

DESTRUKTOR

- Pada Pemograman C++, destruktor dipanggil secara otomatis saat sebuah objek dihancurkan
- Pada PHP4, tidak mengenal destruktor atau belum mendukung desktruktor
- PHP5 mengenal fungsi _destruct() sebagai destruktor. Desktruktor pada PHP5 dipanggil saat onjek dihancurkan seperti C++
- Pemograman Java terdapat metode finalize() yang dapat disungsikan seperti detruktor namun metode ini tidak harus diimplementasikan pada sebuah kelas pada bahasa pemrograman Java. Java objek dibebaskan secara otomatis oleh bagian runtime Java yang disebut dengan garbage collector yang berjalan secara background dimana java akan melakukan pengecekan secara bertahap dan menghapus onjek-objek yang sudah tidak diacu lagi sehingga destuktot tidak wajib ada pada kelas yang menggunakan Java

DESTRUKTOR

C++

```
class Buku {
  private :
  ……
  public :
  Buku(){
   ……
   ~Buku(){
   //desktruktor
   ……
   }
}
```

PHP5

```
class Buku {
   ……
function_construct(){
//konstruktor
 ……
}
function_destruct(){
   //desktruktor
   ……
   }
}
```

Java

```
class Buku {
   ……
Bukut(){
//konstruktor
 ……
}
Protected void finalize(){
   //desktruktor
   ……
   }
}
```

METODE

- Metode atau method pada sebuah kelas hampir sama dengan fungsi atau prosedur pada pemrogrman prosedural
- Metode didalam sebuah kelas juga memiliki akses seperti halnya atribut pada kelas, izin akses itu antara lain private, public dan protected yang memiliki arti sama pada izin akses atribut
- Sebuah kelas boleh memiliki lebih dari satu metode dengan nama yang sama (overloading) asalkan memiliki parameter masikan yang berbeda sehingga compiler atau interpreter dapat mengenali metode mana yang dipanggil

METODE

- Pada bahasa pemrograman Java, dalam sebuah kelas, terdapat juga yang disebut sebagai metode atau atribut statis yang memiliki kata kunci “static”
- Yang dimaksud statis disini yaitu metode yang dapat diakses secara berbagi (share) dengan semua objek lain tanpa harus membuat objek yang memilki metode statis tadi (tanpa proses new)
- Metode statis mempunyai keterbatasan yaitu hanya dapat mengakses atribut atau metode lain didalam kelas yang membungkusnya yang juga bersifat statis
- Metode statis biasanya diimplementasikan pada metode main
- Metode atau atribut yang bersifat statis lebih dahulu dialokasikan di memori untuk dieksekusi atau dipergunakan walaupun kelas yang membungkusnya belum mengalami proses new (tanpa dikenai perintah new)