

Pemrograman Berbasis Obyek

Pengenalan Konsep OOP

Oleh Politeknik Elektronika Negeri Surabaya



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Konten

- Starting OOP Programming
 - Deklarasi class, attribute, method
 - Pembuatan Object
 - Pengaksesan Anggota Object
 - Constructor
 - Reference type

Start OOP Programming

Deklarasi Class

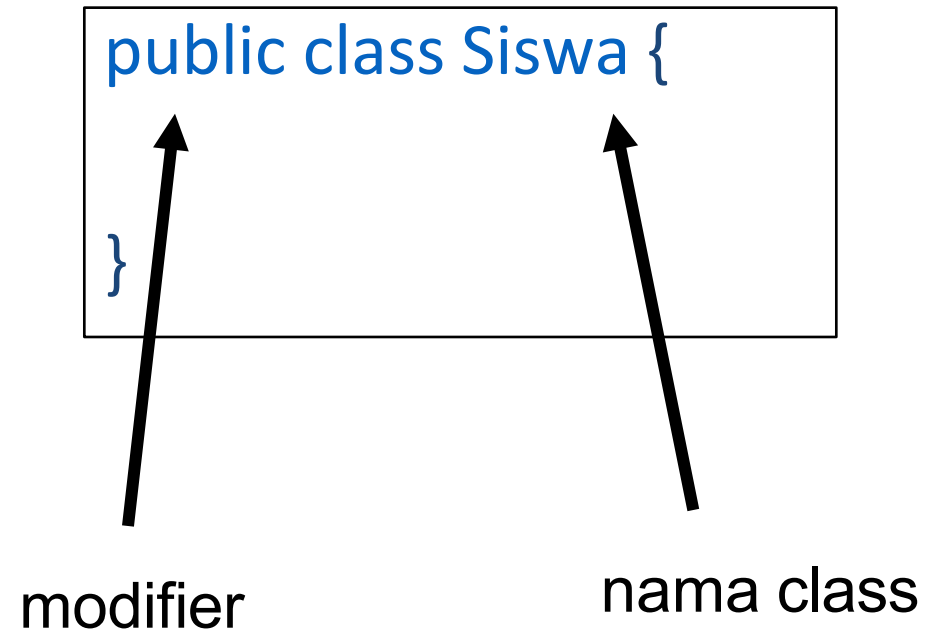
- Class merupakan salah satu Source File dalam Java
- Source File Java harus diakhiri dengan ekstensi `.java`.
- Tiga top-level elemen dalam file java:
 - Package Declaration
 - Import Statements
 - Class Definitions

Tiga element tersebut
dituliskan pada
bagian paling awal
sebuah File Java

Deklarasi Class

```
<modifier> class <classname> {  
    [deklarasi_atribut]  
    [deklarasi_konstruktor]  
    [deklarasi_method]  
}
```

Contoh Deklarasi Class



Deklarasi Atribut

```
<modifier> <tipe> <nama_atribut>;
```

Contoh Deklarasi Atribut

```
public class Siswa {  
    public int nrp;  
    public String nama;  
}
```

Deklarasi method

```
<modifier> <return_type> <nama_method> ([daftar_argumen]){  
    [<statement>]  
}
```

Contoh Deklarasi Method

```
public class Siswa {  
    public int nrp;  
    public String nama;  
    public void Info() {  
        System.out.println("Saya siswa PENS");  
    }  
}
```

Pembuatan Object

- Object adalah instance dari sebuah Class
- Untuk membuat object kita harus menginstansiasi sebuah class

```
public class Siswa{  
    public int nrp;  
    public String nama;  
    public void info(){}  
}
```

```
public class IsiData{  
    Public static void main(String args[]){  
        Siswa budi = New Siswa();  
    }  
}
```

Pengaksesan Anggota Obyek

- Gunakan notasi titik (.) sebagai berikut:
 <object>.<member>
- Member bisa berupa:
 - atribut
 - method

Pengaksesan Anggota Object

```
public class IsiData {  
    public static void main(String args[ ]) {  
        Siswa budi= new Siswa();  
        budi.nrp    = 5;  
        budi.nama   = "Andi";  
        budi.info();  
    }  
}
```

budi adalah sebuah Object
budi adalah instance dari Class Siswa

Constructor (Konstruktor)

- Constructor (Konstruktor) adalah kode yang pertama kali dijalankan pada saat pembuatan suatu obyek.
- Ciri-ciri konstruktor:
 - Mempunyai nama yang sama dengan nama kelas
 - Tidak mempunyai return type
 - Memiliki argumen sebanyak 0..n

Contoh Konstruktor

```
public class Siswa{  
    private int nrp;  
    private String nama;  
  
    public Siswa(int n, String m){  
        nrp = n;  
        nama = m;  
    }  
}
```

Default Constructor

- Jika tidak menuliskan kode konstruktor, maka secara otomatis kompiler akan menambahkan default constructor
- Default constructor → no argument and no body.
- The default constructor has the same access modifier as the class itself, either: public, protected, private or package (no modifier)

Contoh Default Constructor

```
modifiers ClassName() {  
    super();  
}
```

```
public class Siswa{  
    public Siswa(){  
        super(); // menjalankan konstruktor parent  
    }  
}
```

Ingat!!!

- Sekali saja konstruktor dibuat / ditulis secara eksplisit, maka default konstruktor akan hilang

- Contoh:

```
public class Siswa{  
    String nama;  
    public Siswa(int n){  
        this.mana = n;  
    }  
}
```



Reference Type

- Tipe selain tipe primitif dinamakan reference type
- Reference type adalah tipe berbentuk suatu class.
- Pembuatan suatu reference type untuk mengalokasikan memori dilakukan dengan menggunakan kata kunci `new XXX()`. Dimana XXX adalah **konstruktor** dari reference type

Kejadian bila new xxx() dipanggil

- Alokasi memori: ruang untuk obyek baru dibuat di memori dan variabel-variabel diset ke masing-masing nilai default-nya (false, 0, null, dll)
- Inisialisasi nilai atribut yang diberikan secara eksplisit
- Menjalankan konstruktor
- Assignment antara atribut-atribut dengan obyek

Contoh

```
public class MyDate {  
    private int day=1;  
    private int month=1;  
    private int year=2000;  
    public MyDate(int day, int month, int year) {...}  
}
```

```
public class TestMyDate {  
    public static void main(String args[]) {  
        MyDate today=new MyDate(10,10,2005);  
    }  
}
```

Alokasi memori

```
MyDate today = new MyDate(10, 10, 2005);
```

Sudah dibuat object
today, namun belum
mereference alamat
memorio manapun

today

????

Inisialisasi default value

```
MyDate today = new MyDate(10, 10, 2005);
```

today

????

day

0

month

0

year

0

Menjalankan konstruktor

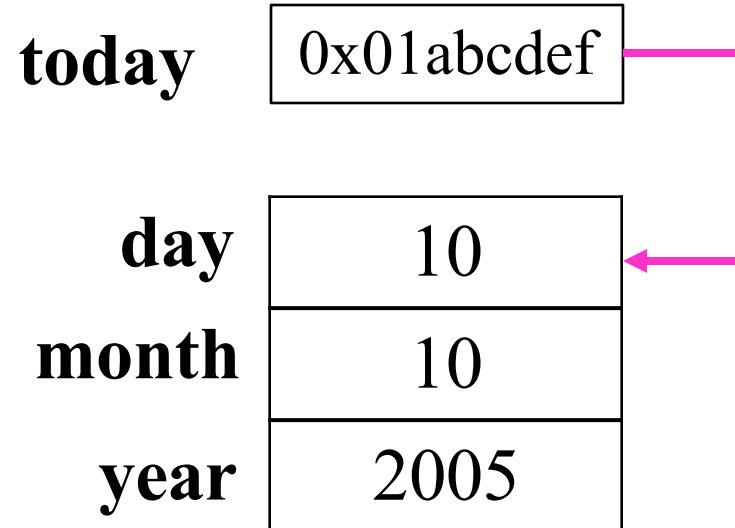
```
MyDate today = new MyDate(10, 10, 2005);
```

today	????
day	10
month	10
year	2005

Membuat referensi

```
MyDate today = new MyDate(10, 10, 2005);
```

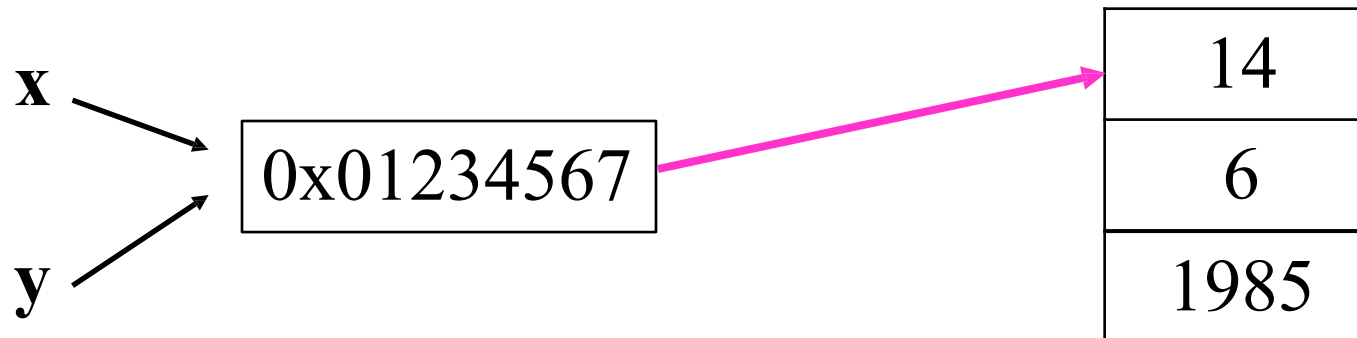
Today merference
object yang baru
dibuat dalam memori



Men-assign reference variable

```
MyDate x = new MyDate(14, 6, 1985);  
MyDate y = x;
```

Alamat yang direferensi oleh x di assign pada y. Sehingga y dan x adalah dua object yang mereference alamat memori yang sama



Class Fundamental : main method

- The *main()* Method

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

- **Public** : method main() dapat diakses oleh apa saja, termasuk java technology interpreter.
- **Static** : keyword ini berfungsi untuk memberi tahu kompiler bahwa method main bisa langsung digunakan dalam contex class yang bersangkutan. Untuk mengeksekusi/menjalankan method yang bertipe static, tidak diperlukan instance nya.
- **Void** : menunjukkan bahwa method main() tidak mengembalikan nilai
- **Main** : merupakan nama method utama dari program java
- **String [] args** : Menyatakan bahwa method main() menerima single parameter yaitu args yang bertipe array. Digunakan pada saat memasukkan parameter pada saat menjalankan program.

Contoh: `java TestGreeting args[0] args[1] ...`



Tugas

1. Buatlah contoh suatu kelas dan definisikan atribut dan methodnya!
2. Buatlah kode program soal no. 2 diatas!
3. Buatlah kelas yang berisi main method yang membuat obyek dari kelas yang telah dibuat di soal no. 3. Selanjutnya obyek tersebut mengakses atribut dan methodnya.

1. Oracle Java Documentation, The Java™ Tutorials, <https://docs.oracle.com/javase/tutorial/>, Copyright © 1995, Oracle 2015.
2. Tita Karlita, Yuliana Setrowati, Rizky Yuniar Hakkun, Pemrograman Berorientasi Obyek, PENS-2012
3. Sun Java Programming, Sun Educational Services, Student Guide, Sun Microsystems, 2001.
bridge to the future
4. John R. Hubbard, Programming With Java, McGraw-Hill, ISBN: 0-07-142040-1, 2004.
<http://www.eepis-its.edu>
5. Patrick Niemeyer, Jonathan Knudsen, Learning Java, O'reilly, CA, ISBN: 1565927184, 2000.
6. Philip Heller, Simon Roberts, Complete Java 2 Certification Study Guide, Third Edition, Sybex, San Francisco, London, ISBN: 0-7821-4419-5, 2002.
7. Herbert Schildt, The Complete Reference, Java™ Seventh Edition, Mc Graw Hill, Osborne, ISBN: 978-0-07-163177-8, 2007