

# Nested Class & Package

Saniati

[saniati@teknokrat.ac.id](mailto:saniati@teknokrat.ac.id)

STMIK Teknokrat, Bandar Lampung

# Inner Class

- Membuat class didalam class dapat dilakukan
- Hasil compile akan sesuai dengan jumlah class yang dibuat.

```
public class Luar {  
    int varLuar;  
  
    void cetakLuar(){  
        System.out.println("cetak luar...");  
    }  
  
    class Dalam{  
        int varDalam;  
  
        void cetakDalam(){  
            System.out.println("cetak dalam...");  
        }  
    }  
}
```

Outer class

Inner class / nested class

Luar\$Dalam.class	CLASS File	11/10/201
Luar.class	CLASS File	11/10/201
Luar.java	JAVA File	11/10/201

Hasil *compile* menunjukkan 2 class. Namun untuk *inner class* mengandung nama *outerclassnya* dibatasi dengan \$

# Inner Class / Nested Class

- Class yang berada di dalam class lain
- Inner class memiliki akses penuh kepada semua field/method class yang diluarnya, sekalipun berlabel “private”.
- Outer class tidak memiliki akses kepada semua semua field/method class yang didalamnya, sekalipun berlabel “public”.

# Object dari Inner Class

- Untuk membuat object dari inner class dapat dengan membuat object outer classnya secara terpisah atau langsung.

```
public class Main {  
    public static void main(String[] args) {  
        Luar l = new Luar();  
        Luar.Dalam d = l.new Dalam();  
  
        l.cetakLuar();  
        d.cetakDalam();  
    }  
}
```

ATAU

```
public class Main {  
    public static void main(String[] args) {  
        Luar.Dalam d = new Luar().new Dalam();  
  
        d.cetakDalam();  
    }  
}
```

# Akses antar Class

```
public class Luar {  
    int varLuar;  
  
    void cetakLuar(){  
        //varDalam=9;  
        //cetakDalam();  
        System.out.println("cetak luar...");  
    }  
  
    class Dalam{  
        int varDalam;  
  
        void cetakDalam(){  
            varLuar=10;  
            cetakLuar();  
            System.out.println("cetak dalam...");  
        }  
    }  
}
```

Outerclass tidak boleh mengakses variable dan method innerclassnya

Innerclass boleh mengakses variable dan method outerclassnya

# Nested Lebih dari satu Class

```
public class Luar {  
    int varLuar;  
    void cetakLuar(){  
        System.out.println("cetak luar...");  
    }  
    class Dalam{  
        int varDalam;  
        void cetakDalam(){  
            varLuar=10;  
            cetakLuar();  
            System.out.println("cetak dalam...");  
        }  
        class Terdalam {  
            int varTerdalam;  
            void cetakTerdalam(){  
                System.out.println("cetak terdalam...");  
            }  
        }  
    }  
}
```

Outer class

Inner class / nested class

Inner class / nested class

```
public class Main {  
    public static void main(String[] args) {  
        Luar l =new Luar();  
        Luar.Dalam d = l.new Dalam();  
        Luar.Dalam.Terdalam t = d.new Terdalam();  
  
        l.cetakLuar();  
        d.cetakDalam();  
        t.cetakTerdalam();  
    }  
}
```

Cara membuat object dari nested class

# Nested Class Dapat Extends/Implements

- Nested Class diijinkan meng-extends class lain maupun meng-implement interface.

```
public interface InterfaceDalam {  
    abstract void cetakTerdalam();  
}
```

```
public class Luar {  
    int varLuar;  
    void cetakLuar(){  
        System.out.println("cetak luar...");  
    }  
  
    class Dalam{  
        int varDalam;  
        void cetakDalam(){  
            varLuar=10;  
            cetakLuar();  
            System.out.println("cetak dalam...");  
        }  
  
        class Terdalam implements InterfaceDalam{  
            int varTerdalam;  
            public void cetakTerdalam(){  
                System.out.println("cetak terdalam...");  
            }  
        }  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Luar l =new Luar();  
        Luar.Dalam d = l.new Dalam();  
        InterfaceDalam t = d.new Terdalam();  
  
        l.cetakLuar();  
        d.cetakDalam();  
        t.cetakTerdalam();  
    }  
}
```

Membuat object nested dapat dibuat dengan nama interface yang diimplementnya

Nested class dapat mengimplement interface

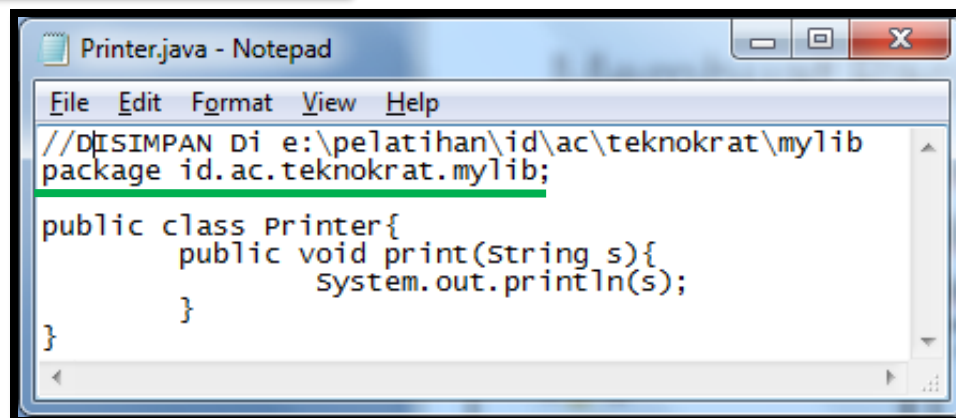
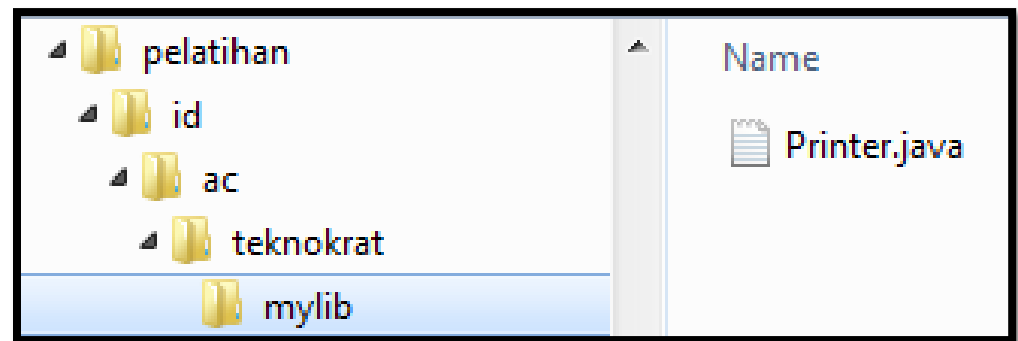
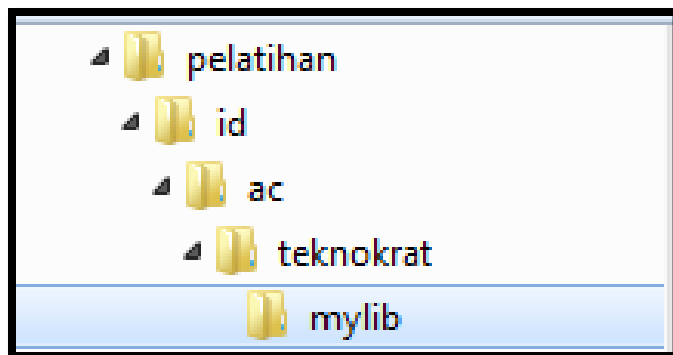
# Package

- Cara yang digunakan untuk mengorganisir java software
- Nama package yang disarankan : nama domain dibalik
  - Contoh : itb.ac.id
  - Package: id.ac.itb.db (untuk library database)  
id.ac.itb.gui (untuk library GUI)  
id.ac.itb.io (untuk library I/O)  
id.ac.itb.io.socket (untuk library I/O specific socket)



# Membuat Package

- Buat folder secara berurutan masuk sesuai nama package
- Untuk file java yang dibuat, tambahkan kode “package folder.folder.folder” dan disimpan ke folder yang sesuai.

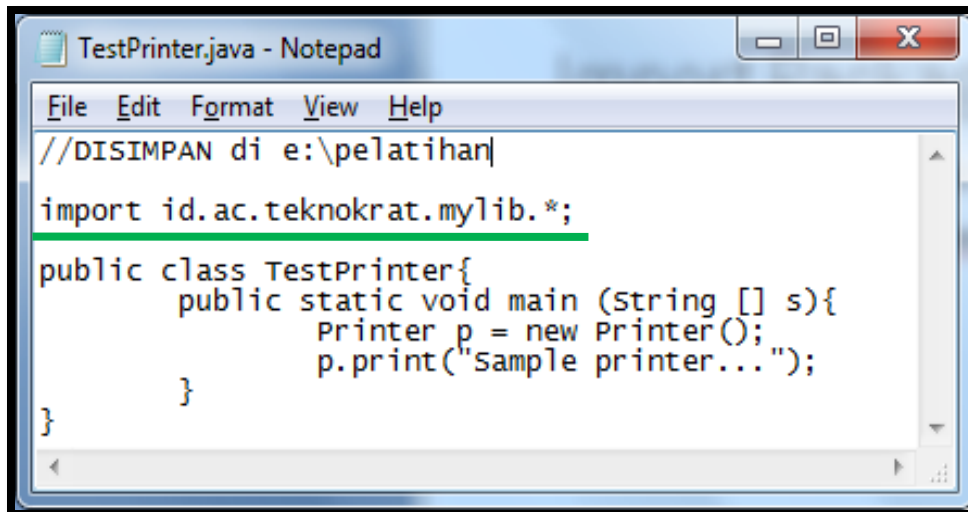


```
Printer.java - Notepad
File Edit Format View Help
//DISIMPAN Di e:\pelatihan\id\ac\teknokrat\mylib
package id.ac.teknokrat.mylib;

public class Printer{
    public void print(String s){
        system.out.println(s);
    }
}
```

# Import Package

- Untuk program java lain yang ingin menggunakan file java pada package yang berbeda lokasi, gunakan keyword “import”.
- Import sebuah package, hanya akan mengimport semua .java/ .class saja. Jika ada folder di dalamnya, tidak ikut ter-import.
- Statement “package” letaknya paling ATAS dan hanya dapat dideklarasikan SEKALI dalam satu source code.



```
TestPrinter.java - Notepad
File Edit Format View Help
//DISIMPAN di e:\pelatihan\
import id.ac.teknokrat.mylib.*;
public class TestPrinter{
    public static void main (String [] s){
        Printer p = new Printer();
        p.print("sample printer...");
    }
}
```

```
E:\pelatihan>javac TestPrinter.java
E:\pelatihan>java TestPrinter
Sample printer...
```

# “Import “ Mengimport Hanya Isi 1 Folder

```
Printer.java - Notepad
File Edit Format View Help
//DISIMPAN Di e:\pelatihan\id\ac\teknokrat\mylib
package id.ac.teknokrat.mylib;

public class Printer{
    public void print(String s){
        System.out.println(s);
    }
}
```

Scanner.java  
disimpan pada  
id.ac.teknokrat

Karena membutuhkan  
Printer maka import  
id.ac.teknokrat.mylib.\*

```
Scanner.java - Notepad
File Edit Format View Help
package id.ac.teknokrat;
import id.ac.teknokrat.mylib.*;
public class scanner{
    public void scan(){
        System.out.println("scanning...");
        Printer p = new Printer();
        p.print("scan complete...");
    }
}
```

Karena membutuhkan Printer  
dan Scanner maka import sesuai  
letak file class masing-masing

```
TestPrinter.java - Notepad
File Edit Format View Help
//DISIMPAN di e:\pelatihan
import id.ac.teknokrat.mylib.*;
import id.ac.teknokrat.*;

public class TestPrinter{
    public static void main (String [] s){
        Printer p = new Printer();
        p.print("sample printer...");
        Scanner s = new Scanner();
        s.scan();
    }
}
```

# Reference

- Pemaparan materi TOT Java Fundamental oleh bapak Tri Haryoko (7-11 Nopember 2016, Bandar Lampung)
- <https://docs.oracle.com/javase/tutorial/java>
- “Thinking in Java”, Bruce Eckel