

Bengkel J2EE/Linux
DASAR PEMROGRAMAN JAVA

By ekobs@developerforce.net

Version **0.1.2**

Last update 02 November 2003

Untuk release terbaru, question, troubleshoot dan feedback
kami undang Anda bergabung dengan
Masyarakat J2EE/Linux Indonesia
Bisa dilakukan dengan mengirim email ke

jlinux-subscribe@yahoogroups.com

PENGANTAR

Bagaimana membuat program dengan bahasa Java ?

Table of Contents

| | | |
|----|--------------------------------------|----|
| 1 | Java Development Kit | 5 |
| | CS-051-001..... | 6 |
| | CS-051-011..... | 8 |
| | CS-051-012..... | 10 |
| | CS-051-013..... | 11 |
| 2 | Platform | 13 |
| | CS-051-021..... | 14 |
| | CS-051-022..... | 16 |
| | CS-051-023..... | 18 |
| 3 | Package, Jar dan CLASSPATH | 20 |
| | CS-051-031..... | 21 |
| | CS-051-032..... | 23 |
| | CS-051-033..... | 25 |
| | CS-051-034..... | 27 |
| 4 | Source code | 29 |
| | CS-051-041..... | 30 |
| | CS-051-042..... | 32 |
| 5 | Tipe Data | 35 |
| | CS-051-051..... | 36 |
| 6 | Operator | 39 |
| | CS-051-061..... | 40 |
| | CS-051-062..... | 41 |
| 7 | Kendali Aliran dan Perulangan | 43 |
| | CS-051-071..... | 44 |
| | CS-051-072..... | 46 |
| | CS-051-073..... | 47 |
| | CS-051-074..... | 48 |
| | CS-051-075..... | 50 |
| 8 | Method | 51 |
| 9 | Penanganan Exception | 52 |
| 10 | Pemrograman Berorientasi Obyek | 53 |
| | CS-051-101..... | 54 |
| 11 | Interface dan Abstract Class | 56 |
| 12 | Java Bean | 57 |
| 13 | Composition, Aggregation | 58 |
| 14 | Format | 59 |
| 15 | File | 60 |
| | CS-051-151..... | 61 |
| | CS-051-152..... | 62 |
| | CS-051-153..... | 64 |
| | CS-051-154..... | 68 |
| | CS-051-155..... | 70 |
| | CS-051-156..... | 72 |
| 16 | mySQL | 77 |

| | | |
|----|------------------------------------|-----|
| | CS-051-161..... | 78 |
| | CS-051-162..... | 79 |
| | CS-051-163..... | 80 |
| 17 | JDBC | 82 |
| | CS-051-171..... | 83 |
| | CS-051-172..... | 85 |
| | CS-051-173..... | 88 |
| | CS-051-174..... | 91 |
| | CS-051-175..... | 93 |
| | CS-051-176..... | 95 |
| | CS-051-177..... | 98 |
| | CS-051-178..... | 101 |
| | CS-051-179..... | 104 |
| 18 | Data Access Object | 107 |
| 19 | Desktop | 108 |
| | CS-051-191..... | 109 |
| | CS-051-192..... | 112 |
| | CS-051-193..... | 115 |
| | CS-051-194..... | 119 |
| | CS-051-195..... | 123 |
| | CS-051-196..... | 126 |
| | CS-051-197..... | 129 |
| | CS-051-198..... | 133 |
| 20 | Aplikasi Database di Desktop | 135 |
| 21 | Security | 136 |
| | CS-051-211..... | 137 |
| | CS-051-212..... | 140 |

1 Java Development Kit

Java Development Kit (JDK) memuat program-program dan library yang Anda butuhkan untuk meng-compile dan me-launch program Java. Dua program utama yang ada di JDK adalah javac dan java.

Untuk meng-install JDK, Anda terlebih dahulu men-download installation file dari <http://java.sun.com>. Anda perlu men-download JDK yang sesuai dengan platform di mana Anda bekerja. Tersedia JDK untuk Linux, Solaris dan Windows.

Installation file untuk Linux dari JDK versi 1.4 adalah j2sdk-1_4_1_02-linux-i586.bin. Yang perlu Anda lakukan hanyalah meng-execute file ini. File ini akan mengalami de-compression. Anda akan mendapatkan sebuah directory j2sdk1.4.1_02 yang memuat JDK Anda. Directory ini dikenal sebagai JAVA_HOME. Banyak aplikasi Java, seperti Tomcat dan JBoss membutuhkan variable JAVA_HOME sebagai environment variable.

Program java dan javac berada di sub-directory bin/ di bawah JAVA_HOME. Agar Anda bisa meng-compile dan me-launch program Java dari directory mana saja, Anda bisa menambahkan JAVA_HOME/bin ke dalam environment variable bernama PATH.

Untuk mengetahui cara penggunaan dari javac, yaitu Java compiler, Anda bisa menjalankan perintah **javac -help** melalui terminal.

Untuk mengetahui cara penggunaan dari java, yaitu Java application launcher, Anda bisa menjalankan perintah **java -help** melalui terminal.

CS-051-001

Panduan instalasi Java Development Kit di Linux ...

Langkah ke-1

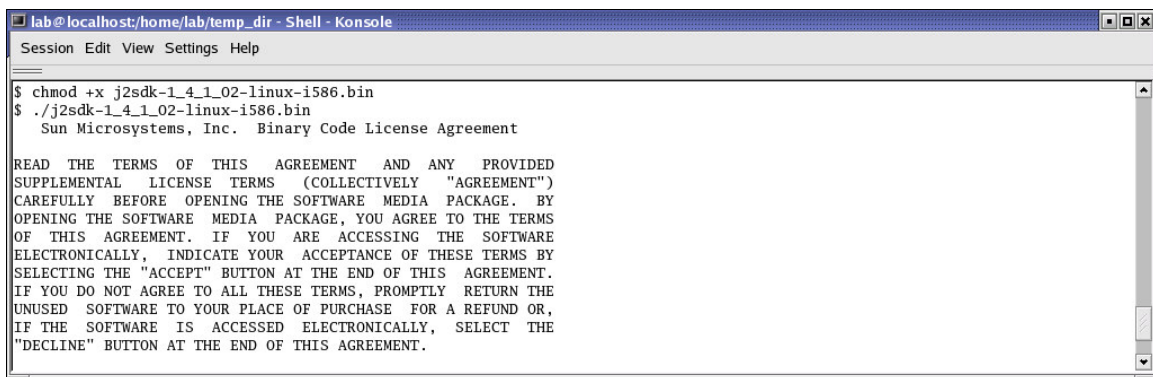
Installation file bisa di-download dari java.sun.com. Anda dapat memilih installation file sesuai Operating System pilihan Anda, yaitu apakah Linux, Solaris atau Windows. Installation file juga tersedia dalam beberapa pilihan format, yaitu compressed file, executable binary, ataupun RPM.

Langkah ke-2

Untuk meng-install Java Development Kit dari installation file ber-format executable compressed file, Anda harus menjalankannya. Perlu Anda perhatikan bahwa Anda perlu mengubah mode dari installation file agar dapat di-execute.

Misalkan di atas Linux, installation file yang Anda download bernama `j2sdk-1_4_1_02-linux-i586.bin`. Untuk menjalankannya, melalui console :

```
$ ./j2sdk-1_4_1_02-linux-i586.bin
```



```
lab@localhost/home/lab/temp_dir - Shell - Konsole
Session Edit View Settings Help
$ chmod +x j2sdk-1_4_1_02-linux-i586.bin
$ ./j2sdk-1_4_1_02-linux-i586.bin
Sun Microsystems, Inc. Binary Code License Agreement

READ THE TERMS OF THIS AGREEMENT AND ANY PROVIDED
SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT")
CAREFULLY BEFORE OPENING THE SOFTWARE MEDIA PACKAGE. BY
OPENING THE SOFTWARE MEDIA PACKAGE, YOU AGREE TO THE TERMS
OF THIS AGREEMENT. IF YOU ARE ACCESSING THE SOFTWARE
ELECTRONICALLY, INDICATE YOUR ACCEPTANCE OF THESE TERMS BY
SELECTING THE "ACCEPT" BUTTON AT THE END OF THIS AGREEMENT.
IF YOU DO NOT AGREE TO ALL THESE TERMS, PROMPTLY RETURN THE
UNUSED SOFTWARE TO YOUR PLACE OF PURCHASE FOR A REFUND OR,
IF THE SOFTWARE IS ACCESSED ELECTRONICALLY, SELECT THE
"DECLINE" BUTTON AT THE END OF THIS AGREEMENT.
```

Awalnya, Anda akan mendapatkan license dari Sun Microsystems, Inc. Untuk kemudian Anda ditanyakan apakah Anda setuju atau tidak. Jika Anda setuju, maka proses installation akan dilanjutkan.

Selanjutnya installation file akan meng-extract file-file JDK ke dalam sebuah sub directory baru. Misalkan Anda menjalankan `j2sdk-1_4_1_02-linux-i586.bin` dari directory `/home/lab` maka sebuah directory `j2sdk1.4.1_02` akan dibuat dan di directory inilah file-file JDK berada.

```
lab@localhost/home/lab/temp_dir - Shell - Konsole
Session Edit View Settings Help
Do you agree to the above license terms? [yes or no]
yes
Unpacking...
Checksumming...
0
0
Extracting...
UnZipSFX 5.40 of 28 November 1998, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  creating: j2sdk1.4.1_02/
  creating: j2sdk1.4.1_02/jre/
  creating: j2sdk1.4.1_02/jre/bin/
  inflating: j2sdk1.4.1_02/jre/bin/java
```

```
Do you agree to the above license terms? [yes or no]
yes
Unpacking...
Checksumming...
0
0
Extracting...
UnZipSFX 5.40 of 28 November 1998, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  creating: j2sdk1.4.1_02/
  creating: j2sdk1.4.1_02/jre/
  creating: j2sdk1.4.1_02/jre/bin/
  inflating: j2sdk1.4.1_02/jre/bin/java
  inflating: j2sdk1.4.1_02/jre/bin/keytool
  inflating: j2sdk1.4.1_02/jre/bin/policytool
  inflating: j2sdk1.4.1_02/jre/bin/kinit
  inflating: j2sdk1.4.1_02/jre/bin/klist
  inflating: j2sdk1.4.1_02/jre/bin/ktab
  inflating: j2sdk1.4.1_02/jre/bin/rmiregistry
  inflating: j2sdk1.4.1_02/jre/bin/rmid
  inflating: j2sdk1.4.1_02/jre/bin/orbd
  inflating: j2sdk1.4.1_02/jre/bin/servertool
  inflating: j2sdk1.4.1_02/jre/bin/tnameserv
  inflating: j2sdk1.4.1_02/jre/bin/java_vm
```

sehingga akhirnya :

```
  inflating: j2sdk1.4.1_02/man/ja_JP.eucJP/man1/native2ascii.1
  inflating: j2sdk1.4.1_02/man/ja_JP.eucJP/man1/rmic.1
  inflating: j2sdk1.4.1_02/man/ja_JP.eucJP/man1/serialver.1
  inflating: j2sdk1.4.1_02/man/ja_JP.eucJP/man1/idlj.1
Done.
```

Proses instalasi adalah sederhana. Dan kini Anda mempunyai JDK yang siap Anda gunakan untuk mengembangkan aplikasi dengan Java.

CS-051-011

Sebuah praktikum untuk menulis sebuah aplikasi Java yang menampilkan pesan ke console.

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava

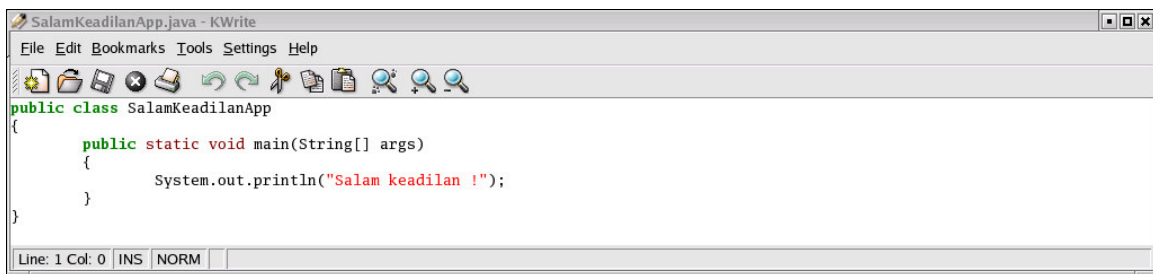
Langkah

Langkah ke-1

Tulis SalamKeadilanApp.java, simpan di-directory yang sudah dipersiapkan.

SalamKeadilanApp.java

```
public class SalamKeadilanApp
{
    public static void main(String[] args)
    {
        System.out.println("Salam keadilan !");
    }
}
```



Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

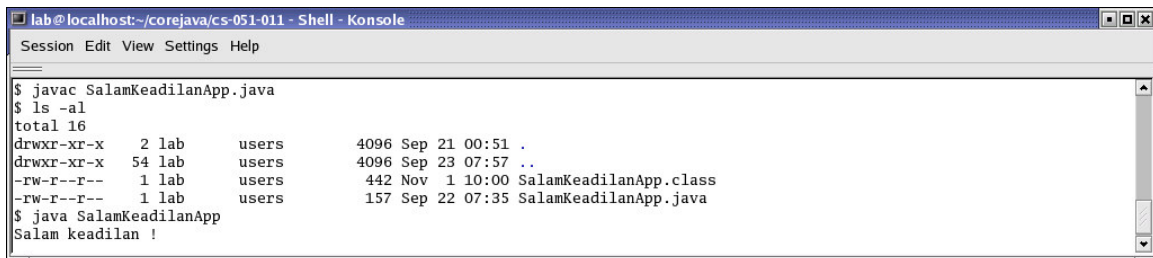
Compile ...

```
$ javac SalamKeadilanApp.java
```

akan dihasilkan SalamKeadilan.class

```
$ ls
```

```
SalamKeadilanApp.class SalamKeadilanApp.java
```



Langkah ke-4

Launch ...

```
$ java SalamKeadilanApp  
Salam keadilan !
```

CS-051-012

Sebuah praktikum dasar lain ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava

Langkah

Langkah ke-1

Tulis DatePrintlnApp.java, simpan di-directory yang sudah dipersiapkan.

DatePrintlnApp.java

```
import java.text.SimpleDateFormat;
import java.util.Date;

public class DatePrintlnApp
{
    public static void main(String[] args)
    {
        Date date = new Date();
        SimpleDateFormat format
            = new SimpleDateFormat("dd MMMM yyyy hh:mm:ss");
        String s = format.format(date);
        System.out.println(s);
    }
}
```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac DatePrintlnApp.java
```

akan dihasilkan DatePrintlnApp.class

Langkah ke-4

Launch ...

```
$ java DatePrintlnApp
01 November 2003 03:46:58
```

CS-051-013

Sebuah praktikum dasar lain ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava

Langkah

Langkah ke-1

Tulis FilePrintlnApp.java, simpan di-directory yang sudah dipersiapkan.

FilePrintlnApp.java

```
import java.io.*;

public class FilePrintlnApp
{
    public static void main(String[] args)
        throws Exception
    {
        String fileName = args[0];

        RandomAccessFile raf
            = new RandomAccessFile(fileName, "r");
        String line;
        while((line = raf.readLine()) != null)
        {
            System.out.println(line);
        }
        raf.close();
    }
}
```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac FilePrintlnApp.java
```

akan dihasilkan FilePrintlnApp.class

Langkah ke-4

Launch ... misal untuk membaca file /home/lab/.bashrc

```
$ java FilePrintlnApp /home/lab/.bashrc
# .bashrc

# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
```

2Platform

Untuk membuat program dengan bahasa Java, Anda menulis source code yang disimpan dalam file berekstension **.java**. Selanjutnya Anda meng-compile source code tersebut dengan perintah **javac**. Yang Anda dapatkan adalah sebuah file berekstension **.class**. Untuk meluncurkan program Anda, Anda memanggil perintah **java**.

File berekstension **.class** yang diperoleh setelah Anda meng-compile source code, bukanlah executable binary. Java meng-compile source code Anda menjadi **byte-code** yang independen terhadap platform. Dengan strategi ini program yang Anda kembangkan bisa **write once, run anywhere**. Anda bisa mengembangkan program di atas Linux, ia akan bekerja di atas Operating System lain seperti Solaris, dan Windows.

Untuk menjalankan byte-code ini dibutuhkan Java Virtual Machine (JVM). Anda menghidupkan JVM dengan memanggil perintah **java**. JVM bertanggung jawab untuk menjalankan byte-code tersebut di atas platform tertentu. Awalnya JVM bekerja sebagai interpreter dari byte-code.

Dalam perkembangannya, JVM mempunyai kemampuan berupa **Just-In-Time** compiler, yang meng-compile byte code ke dalam native code saat dijalankan di dalam JVM.

CS-051-021

Merupakan pengembangan dari SalamKeadilanApp.java ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava

Langkah

Langkah ke-1

Tulis MyAgent.java, simpan di-directory yang sudah dipersiapkan.

MyAgent . java

```
public class MyAgent
{
    private String name;

    public MyAgent(String name)
    {
        this.name = name;
    }

    public String getMessage()
    {
        return
            "Salam keadilan !. dari " + name;
    }
}
```

Langkah ke-2

Tulis SalamKeadilanApp.java, simpan di-directory yang sudah dipersiapkan.

SalamKeadilanApp . java

```
public class SalamKeadilanApp
{
    public static void main(String[] args)
    {
        MyAgent agent = new MyAgent("ibrahim");
        String s = agent.getMessage();
        System.out.println(s);
    }
}
```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac SalamKeadilanApp.java
```

akan dihasilkan SalamKeadilanApp.class dan MyAgent.class

```
$ ls
MyAgent.class  MyAgent.java
SalamKeadilanApp.class  SalamKeadilanApp.java
```

Langkah ke-5

Launch ...

```
$ java SalamKeadilanApp  
Salam keadilan !. dari ibrahim
```

CS-051-022

Merupakan pengembangan dari DatePrintlnApp.java ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava

Langkah

Langkah ke-1

Tulis MyDate.java, simpan di-directory yang sudah dipersiapkan.

MyDate.java

```
import java.text.SimpleDateFormat;
import java.util.Date;

public class MyDate
{
    private Date date;

    public MyDate()
    {
        date = new Date();
    }

    public String toString()
    {
        return toString("dd MMMM yyyy hh:mm:ss");
    }

    public String toString(String formatString)
    {
        SimpleDateFormat format
            = new SimpleDateFormat(formatString);
        String s = format.format(date);
        return s;
    }
}
```

Langkah ke-2

Tulis DatePrintlnApp.java, simpan di-directory yang sudah dipersiapkan.

DatePrintlnApp.java

```
public class DatePrintlnApp
{
    public static void main(String[] args)
    {
        MyDate date = new MyDate();
        String s = date.toString();
        System.out.println(s);

        s = date.toString("dd/MM/yyyy hh:mm:ss");
        System.out.println(s);
    }
}
```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac DatePrintlnApp.java
```

akan dihasilkan MyDate.class dan DatePrintlnApp.class

Langkah ke-5

Launch ...

```
$ java DatePrintlnApp
01 November 2003 03:56:11
01/11/2003 03:56:11
```


CS-051-023

Merupakan pengembangan dari FilePrintlnApp.java ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava

Langkah

Langkah ke-1

Tulis MyFile.java, simpan di-directory yang sudah dipersiapkan.

MyFile.java

```
import java.io.*;

public class MyFile
    extends RandomAccessFile
{
    public MyFile(String fileName)
        throws Exception
    {
        super(fileName, "r");
    }

    public void println()
        throws Exception
    {
        seek(0);
        String line;
        while((line = readLine()) != null)
        {
            System.out.println(line);
        }
    }
}
```

Langkah ke-2

Tulis FilePrintlnApp.java, simpan di-directory yang sudah dipersiapkan.

FilePrintlnApp.java

```
public class FilePrintlnApp
{
    public static void main(String[] args)
        throws Exception
    {
        String fileName = args[0];
        MyFile file = new MyFile(fileName);
        file.println();
    }
}
```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac FilePrintlnApp.java
```

akan dihasilkan MyFile.class dan FilePrintlnApp.class

Langkah ke-5

Launch ...

```
$ java FilePrintlnApp /home/lab/.bashrc  
# .bashrc
```

```
# User specific aliases and functions
```

```
# Source global definitions  
if [ -f /etc/bashrc ]; then  
    . /etc/bashrc  
fi
```

3Package, Jar dan CLASSPATH

Package, Jar dan CLASSPATH merupakan tiga pendekatan teknis dari teknologi Java yang mendasar.

Package digunakan untuk mengelompokkan class-class. Secara teknis pengelompokkan ini diwujudkan dengan membuat struktur direktori untuk menyimpan class-class tersebut.

Jar digunakan untuk meng-compress class-class ke dalam satu archive, dalam ekstension .jar. Class-class di dalam .jar mempunyai struktur direktori sesuai package-package yang ada.

CLASSPATH adalah environment variable yang menunjukkan kepada javac dan java, di mana class-class yang dibutuhkan berada.

CS-051-031

Merupakan pengembangan dari SalamKeadilanApp.java, di mana source code dipisahkan ke dalam package.

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava. Di bawah directory tersebut buat dua buah sub directory : agent dan app.

Langkah

Langkah ke-1

Tulis MyAgent.java, simpan di sub directory agent yang sudah dipersiapkan.

agent/MyAgent . java

```
package agent;

public class MyAgent
{
    private String name;

    public MyAgent(String name)
    {
        this.name = name;
    }

    public String getMessage()
    {
        return "Salam keadilan !. dari " + name;
    }
}
```

Langkah ke-2

Tulis SalamKeadilanApp.java, simpan di sub directory app yang sudah dipersiapkan.

app/SalamKeadilanApp . java

```
package app;

import agent.MyAgent;

public class SalamKeadilanApp
{
    public static void main(String[] args)
    {
        MyAgent agent = new MyAgent("ibrahim");
        String s = agent.getMessage();
        System.out.println(s);
    }
}
```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac app/SalamKeadilanApp.java
```

akan dihasilkan app/SalamKeadilanApp.class dan agent/MyAgent.class

Langkah ke-5

Launch ...

```
$ java app.SalamKeadilanApp  
Salam keadilan !. dari ibrahim
```

CS-051-032

Merupakan pengembangan dari DatePrintlnApp.java, di mana source code dipisahkan ke dalam package.

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava. Di bawah directory tersebut buat dua buah sub directory : date dan app.

Langkah

Langkah ke-1

Tulis MyDate.java, simpan di sub directory date yang sudah dipersiapkan.

date/MyDate.java

```
package date;

import java.text.SimpleDateFormat;
import java.util.Date;

public class MyDate
{
    private Date date;

    public MyDate()
    {
        date = new Date();
    }

    public String toString()
    {
        return toString("dd MMMM yyyy hh:mm:ss");
    }

    public String toString(String formatString)
    {
        SimpleDateFormat format
            = new SimpleDateFormat(formatString);
        String s = format.format(date);
        return s;
    }
}
```

Langkah ke-2

Tulis DatePrintlnApp.java, simpan di sub directory app yang sudah dipersiapkan.

app/DatePrintlnApp.java

```
package app;

import date.MyDate;

public class DatePrintlnApp
{
    public static void main(String[] args)
    {
        MyDate date = new MyDate();
        String s = date.toString();
    }
}
```

```
        System.out.println(s);  
        s = date.toString("dd/MM/yyyy hh:mm:ss");  
        System.out.println(s);  
    }  
}
```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac app/DatePrintlnApp.java
```

akan dihasilkan app/DatePrintlnApp.class dan date/MyDate.class

Langkah ke-5

Launch ...

```
$ java app.DatePrintlnApp  
02 November 2003 05:55:22  
02/11/2003 05:55:22
```

CS-051-033

Merupakan contoh menggunakan library Log4J

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava. Di bawah directory tersebut buat dua buah sub directory : date dan app.

Langkah

Langkah ke-1

Tulis MyDate.java, simpan di sub directory date yang sudah dipersiapkan.

date/MyDate.java

```
package date;

import java.text.SimpleDateFormat;
import java.util.Date;

import org.apache.log4j.*;

public class MyDate
{
    private Logger logger
        = Logger.getLogger(MyDate.class);

    private Date date;

    public MyDate()
    {
        logger.debug("inside MyDate()");
        date = new Date();
    }

    public String toString()
    {
        logger.debug("inside toString()");
        return toString("dd MMMM yyyy hh:mm:ss");
    }

    public String toString(String formatString)
    {
        logger.debug("inside toString(String)");
        SimpleDateFormat format
            = new SimpleDateFormat(formatString);
        String s = format.format(date);
        return s;
    }
}
```

Langkah ke-2

Tulis DatePrintlnApp.java, simpan di sub directory app yang sudah dipersiapkan.

app/DatePrintlnApp.java

```
package app;
```



```
import date.MyDate;

import org.apache.log4j.*;

public class DatePrintlnApp
{
    public static void main(String[] args)
    {
        BasicConfigurator.configure();

        MyDate date = new MyDate();
        String s = date.toString();
        System.out.println(s);

        s = date.toString("dd/MM/yyyy hh:mm:ss");
        System.out.println(s);
    }
}
```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac app/DatePrintlnApp.java
```

akan dihasilkan app/DatePrintlnApp.class dan date/MyDate.class

Langkah ke-5

Launch ...

```
$ java app.DatePrintlnApp
1 [main] DEBUG date.MyDate - inside MyDate()
9 [main] DEBUG date.MyDate - inside toString()
9 [main] DEBUG date.MyDate - inside toString(String)
02 November 2003 06:04:26
107 [main] DEBUG date.MyDate - inside toString(String)
02/11/2003 06:04:26
```

CS-051-034

Merupakan pengembangan dari FilePrintlnApp.java ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava. Di bawah directory ini buat dua sub directory, yaitu file dan app

Langkah

Langkah ke-1

Tulis MyFile.java, simpan di-sub directory file ...

file/MyFile.java

```
package file;

import java.io.*;

public class MyFile
    extends RandomAccessFile
{
    public MyFile(String fileName)
        throws Exception
    {
        super(fileName, "r");
    }

    public void println()
        throws Exception
    {
        seek(0);
        String line;
        while((line = readLine()) != null)
        {
            System.out.println(line);
        }
    }
}
```

Langkah ke-2

Tulis FilePrintlnApp.java, simpan di sub-directory app

app/FilePrintlnApp.java

```
package app;

import file.MyFile;

public class FilePrintlnApp
{
    public static void main(String[] args)
        throws Exception
    {
        String fileName = args[0];
        MyFile file = new MyFile(fileName);
        file.println();
    }
}
```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac app/FilePrintlnApp.java
```

akan dihasilkan file/MyFile.class dan app/FilePrintlnApp.class

Langkah ke-5

Launch ...

```
$ java app.FilePrintlnApp /home/lab/.bashrc  
# .bashrc
```

```
# User specific aliases and functions
```

```
# Source global definitions
```

```
if [ -f /etc/bashrc ]; then
```

```
    . /etc/bashrc
```

```
fi
```

4Source code

Dalam mengembangkan program Java, Anda menulis class-class. Source code dari class Java di simpan dalam file berekstension .java. Di dalam source code dideklarasikan package, import, class, variable dan method.

Deklarasi package digunakan untuk menunjukkan package dimana class akan dikumpulkan. Deklarasi package adalah optional, jika class akan dikumpulkan ke dalam default package.

Deklarasi import digunakan untuk menunjukkan package atau class yang dibutuhkan dalam class tersebut. Deklarasi import adalah optional.

Deklarasi class merupakan bagian inti dari sebuah source code Java. Di sini terutama menuliskan nama class. Nama class ini adalah case-sensitive, dan harus sama dengan nama file di mana source code disimpan.

Anda juga bisa menuliskan apakah class ini sebuah interface, abstract class atau class biasa. Terdapat perbedaan antara interface, abstract class dan class biasa. Sebuah interface hanya memuat deklarasi method, tetapi method-method tsb tidak mempunyai implementasi. Sebuah abstract class memuat sejumlah method yang sudah diimplementasikan, tetapi satu atau lebih belum. Sedangkan sebuah class yang biasa mempunyai method-method yang semuanya diimplementasikan.

Kemudian Anda bisa menuliskan bahwa class ini meng-extends class tertentu. Ini adalah perwujudan dari inheritance di dalam bahasa Java. Dengan meng-extends dari class tertentu, maka class yang Anda tulis adalah sub-class dari class tersebut.

Anda juga bisa menuliskan bahwa class tsb meng-implements interface-interface tertentu. Dengan meng-implements suatu interface, Anda terikat dengan contract untuk mengimplementasikan semua method yang dideklarasikan dalam interface tsb.

Selanjutnya di dalam tubuh dari class yang Anda tulis, Anda bisa mendeklarasikan variable-variable dan method-method.

CS-051-041

Merupakan contoh source code Java yang lebih lengkap ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava. Di bawah directory tersebut buat dua buah sub directory : bean dan app.

Langkah

Langkah ke-1

Tulis MyEmployee.java, simpan di sub directory bean yang sudah dipersiapkan.

bean/MyEmployee.java

```
package bean;

import java.util.Date;

public class MyEmployee
{
    private String id;
    private String name;
    private String department;

    public MyEmployee(String id)
    {
        this(id, null, null);
    }

    public MyEmployee
        (String id, String name, String department)
    {
        this.id = id;
        this.name = name;
        this.department = department;
    }

    public String getId()
    {
        return this.id;
    }

    public String getName()
    {
        return this.name;
    }

    public String getDepartment()
    {
        return this.department;
    }
}
```

Langkah ke-2

Tulis EmployeePrintlnApp.java, simpan di sub directory app yang sudah dipersiapkan.

app/EmployeePrintlnApp.java

```

package app;

import bean.MyEmployee;

public class EmployeePrintlnApp
{
    public static void main(String[] args)
    {
        MyEmployee employee
            = new MyEmployee(
                "101", "Lutung Kasarung", "Finance");
        System.out.println("EMPLOYEE DATA");
        System.out.println("ID           : "
            + employee.getId());
        System.out.println("Name           : "
            + employee.getName());
        System.out.println("Dept.         : "
            + employee.getDepartment());
    }
}

```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac app/EmployeePrintlnApp.java
```

akan dihasilkan app/ EmployeePrintlnApp.class dan bean/MyEmployee.class

Langkah ke-5

Launch ...

```

$ java app.EmployeePrintlnApp
EMPLOYEE DATA
ID           : 101
Name        : Lutung Kasarung
Dept.      : Finance

```

CS-051-042

Merupakan contoh source code Java yang lebih lengkap ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava. Di bawah directory tersebut buat dua buah sub directory : bean dan app.

Langkah

Langkah ke-1

Tulis MyEmployee.java, simpan di sub directory bean yang sudah dipersiapkan.

bean/MyEmployee.java

```
package bean;

import java.util.Date;

public class MyEmployee
{
    private String id;
    private String name;
    private String department;
    private String jobTitle;

    public MyEmployee(String id)
    {
        this(id, null, null, null);
    }

    public MyEmployee(String id, String name,
        String department, String jobTitle)
    {
        this.id = id;
        this.name = name;
        this.department = department;
        this.jobTitle = jobTitle;
    }

    public String getId()
    {
        return this.id;
    }

    public String getName()
    {
        return this.name;
    }

    public String getDepartment()
    {
        return this.department;
    }

    public String getJobTitle()
    {
        return this.jobTitle;
    }
}
```

```

    public void setJobTitle(String jobTitle)
    {
        this.jobTitle = jobTitle;
    }
}

```

Langkah ke-2

Tulis EmployeePrintlnApp.java, simpan di sub directory app yang sudah dipersiapkan.

app/EmployeePrintlnApp.java

```

package app;

import bean.MyEmployee;

public class EmployeePrintlnApp
{
    public static void main(String[] args)
    {
        MyEmployee employee
            = new MyEmployee
                ("101", "Lutung Kasarung", "Finance", "Manager");
        System.out.println("EMPLOYEE DATA --- before modified ... ");
        System.out.println("ID           : " + employee.getId());
        System.out.println("Name      : " + employee.getName());
        System.out.println("Dept.     : " + employee.getDepartment());
        System.out.println("Job Title. : " + employee.getJobTitle());
        System.out.println();

        employee.setJobTitle("Director");

        System.out.println("EMPLOYEE DATA --- after modified ... ");
        System.out.println("ID           : " + employee.getId());
        System.out.println("Name      : " + employee.getName());
        System.out.println("Dept.     : " + employee.getDepartment());
        System.out.println("Job Title. : " + employee.getJobTitle());
    }
}

```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac app/EmployeePrintlnApp.java
```

akan dihasilkan app/ EmployeePrintlnApp.class dan bean/MyEmployee.class

Langkah ke-5

Launch ...

```

$ java app.EmployeePrintlnApp
EMPLOYEE DATA --- before modified ...
ID           : 101
Name      : Lutung Kasarung

```


Dept. : Finance
Position. : Manager

EMPLOYEE DATA --- after modified ...

ID : 101
Name : Lutung Kasarung
Dept. : Finance
Position. : Director

5 Tipe Data

Di dalam program Java, Anda bisa mempunyai variable dengan tipe data primitive ataupun variable yang merupakan obyek. Variable yang mempunyai tipe data primitive hanya bisa menggemang data. Sedangkan variable yang merupakan obyek, adalah merupakan instance dari Java class, sehingga mempunyai method-method yang bisa di-invoke.

Tipe data primitive yang disediakan terdiri dari boolean, char, byte, short, int, long, double, float.

Tersedia juga class-class yang merupakan wrapper class untuk tipe data primitive. Wrapper class tersebut adalah Boolean, Character, Byte, Short, Integer, Long, Double dan Float.

Class Integer mempunyai dua method yang lumrah digunakan yaitu `parseInt()` dan `intValue()`.

Terdapat beberapa class yang lumrah digunakan dalam bahasa pemrograman Java, yaitu String, StringBuffer dan Date.

Class String digunakan untuk menggemang serangkaian karakter, seperti kata, dan kalimat. String adalah non mutable.

Class StringBuffer juga mempunyai digunakan untuk menggemang serangkaian karakter. StringBuffer adalah mutable.

Untuk variable yang menggemang sekumpulan data, bisa digunakan array, Vector dan Hashtable.

Array bisa digunakan untuk menggemang sekumpulan data bertipe primitive ataupun obyek. Data-data di dalam array bertipe sama, dan disusun dalam urutan. Jumlah data yang bisa digemang array adalah terbatas, ditentukan dalam deklarasi variable.

Vector hanya bisa menggemang data bertipe obyek. Data-data di dalam Vector bisa berbeda-beda. Sebagaimana array, data juga disusun dalam urutan. Jumlah data yang bisa digemang oleh Vector bersifat dinamik, bisa ditambah jika dibutuhkan.

Hashtable juga hanya bisa menggemang data bertipe obyek. Sebagaimana Vector, data yang digemang bisa berbeda-beda. Tetapi di dalam Hashtable, data tidak diurutkan. Data disimpan sebagai pasangan key-value.

CS-051-051

Merupakan ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava. Di bawah directory tersebut buat dua buah sub directory : bean dan app.

Langkah

Langkah ke-1

Tulis MyEmployee.java, simpan di sub directory bean yang sudah dipersiapkan.

bean/MyEmployee.java

```
package bean;

import java.util.Date;

public class MyEmployee
{
    private String id;
    private boolean permanentEmployee;
    private String name;
    private String department;
    private String jobTitle;
    private double salary;

    public MyEmployee(
        String id, String name,
        String department, String jobTitle,
        boolean permanentEmployee, double salary)
    {
        this.id = id;
        this.permanentEmployee = permanentEmployee;
        this.name = name;
        this.department = department;
        this.jobTitle = jobTitle;
        this.salary = salary;
    }

    public String getId()
    {
        return this.id;
    }

    public String getName()
    {
        return this.name;
    }

    public String getDepartment()
    {
        return this.department;
    }

    public String getJobTitle()
    {
        return this.jobTitle;
    }
}
```

```

public void setJobTitle(String jobTitle)
{
    this.jobTitle = jobTitle;
}

public String toString()
{
    return
        "ID          : " + this.id + "\n" +
        "Permanent Employee : "
        + this.permanentEmployee + "\n" +
        "Name           : " + this.name + "\n" +
        "Dept.          : " + this.department + "\n" +
        "Position      : " + this.jobTitle + "\n" +
        "Salary        : " + this.salary;
}
}

```

Langkah ke-2

Tulis EmployeePrintlnApp.java, simpan di sub directory app yang sudah dipersiapkan.

app/EmployeePrintlnApp.java

```

package app;

import bean.MyEmployee;

public class EmployeePrintlnApp
{
    public static void main(String[] args)
    {
        MyEmployee employee
        = new MyEmployee
            ("101", "Lutung Kasarung",
            "Finance", "Manager", true, 10000000);
        System.out.println("EMPLOYEE DATA --- after creation");
        System.out.println(employee.toString());
    }
}

```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac app/EmployeePrintlnApp.java
```

akan dihasilkan app/ EmployeePrintlnApp.class dan bean/MyEmployee.class

Langkah ke-5

Launch ...

```

$ java app.EmployeePrintlnApp
EMPLOYEE DATA --- after creation
ID          : 101
Permanent Employee      : true
Name           : Lutung Kasarung

```

Dept. : Finance
Position : Manager
Salary : 1.0E7

6Operator

Unary operator

Latihan + - * /

Latihan = untuk object ...

Latihan equals

CS-051-061

Merupakan contoh penggunaan operator +, -, * dan /

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis CalculatorApp.java, simpan di directory yang sudah dipersiapkan.

CalculatorApp

```
public class CalculatorApp
{
    public static void main(String[] args)
    {
        double x = 7.5;
        double y = 3.3;
        System.out.println("x      = " + x);
        System.out.println("y      = " + y);
        System.out.println("x + y = " + (x+y));
        System.out.println("x - y = " + (x-y));
        System.out.println("x * y = " + (x*y));
        System.out.println("x / y = " + (x/y));
    }
}
```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac CalculatorApp.java
```

akan dihasilkan CalculatorApp.class

Langkah ke-4

Launch ...

```
$ java CalculatorApp
x      = 7.5
y      = 3.3
x + y = 10.8
x - y = 4.2
x * y = 24.75
x / y = 2.272727272727273
```

CS-051-062

Merupakan contoh penggunaan operator +, -, * dan /

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava. Persiapkan dua buah sub directory calculator dan app.

Langkah

Langkah ke-1

Tulis CalculatorApp.java, simpan di sub-directory calculator yang sudah dipersiapkan.

calculator/MyCalculator.java

```
package calculator;

public class MyCalculator
{
    public double add(double x, double y)
    {
        return x + y;
    }
    public double subscribe(double x, double y)
    {
        return x - y;
    }
    public double multiply(double x, double y)
    {
        return x * y;
    }
    public double divide(double x, double y)
    {
        return x / y;
    }
}
```

Langkah ke-2

Tulis CalculatorApp.java, simpan di sub-directory app

app/CalculatorApp.java

```
package app;

import calculator.MyCalculator;

public class CalculatorApp
{
    public static void main(String[] args)
    {
        double x = 7.5;
        double y = 3.3;

        MyCalculator calc = new MyCalculator();

        System.out.println("x      = " + x);
        System.out.println("y      = " + y);
        System.out.println("x + y = " + calc.add(x, y));
        System.out.println("x - y = " + calc.subscribe(x, y));
        System.out.println("x * y = " + calc.multiply(x, y));
    }
}
```



```
        System.out.println("x / y = " + calc.divide(x,y));
    }
}
```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac app/CalculatorApp.java
```

akan dihasilkan app/CalculatorApp.class dan calculator/MyCalculator.class

Langkah ke-4

Launch ...

```
$ java app.CalculatorApp
x      = 7.5
y      = 3.3
x + y  = 10.8
x - y  = 4.2
x * y  = 24.75
x / y  = 2.272727272727273
```

7Kendali Aliran dan Perulangan

Untuk mengendalikan aliran program di dalam Java dapat digunakan kata kunci **if** dan **switch**. Sedangkan untuk perulangan dapat digunakan **for**, **do-while** dan **while**.

Dengan **if**, Anda bisa menentukan bahwa sebuah blok program akan dijalankan jika kondisi tertentu terpenuhi.

Dengan **switch**, Anda bisa menentukan blok program mana yang akan dijalankan berdasarkan nilai dari sebuah variable.

Dengan **for**, Anda bisa menentukan agar sebuah blok program dijalankan beberapa kali.

Dengan **do-while**, Anda bisa menentukan agar sebuah program dijalankan, lalu dijalankan kembali berulang ulang selama kondisi tertentu masih terpenuhi.

Dengan **while**, Anda bisa menentukan agar sebuah blok program dijalankan jika sebuah kondisi tertentu terpenuhi, dan dijalankan kembali berulang ulang selama kondisi tersebut masih terpenuhi.

CS-051-071

Merupakan contoh bagaimana menyusun kendali aliran dengan if

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis IfElseApp.java, simpan di directory yang sudah dipersiapkan.

IfElseApp.java

```
public class IfElseApp
{
    public static void main(String[] args)
    {

        double a = 4;
        double b = -13;
        double c = -12;

        double D = b*b - 4*a*c;
        double x1, x2;

        System.out.println
            ("Persamaan kuadrat a*x^2+b*x+c=0, di mana");
        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("c = " + c);
        System.out.println
            ("dengan menerapkan rumus ABC diketahui bahwa : ");

        if(D<0)
        {
            System.out.println
                ("Akar-akar persamaan imajiner.");
        }
        else
        if(D==0)
        {
            System.out.println
                ("Akar-akar persamaan nyata tetapi kembar.");
            x1 = -b/(2*a);
            System.out.println("x1 = x2 = " + x1);
        }
        else
        {
            System.out.println
                ("Akar-akar persamaan nyata dan berbeda.");
            x1 = (-b+Math.sqrt(D))/(2*a);
            x2 = (-b-Math.sqrt(D))/(2*a);
            System.out.println("x1 = " + x1 + " x2 = " + x2);
        }
    }
}
```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-
Compile ...

```
$ javac IfElseApp.java
```

akan dihasilkan IfElseApp.class

Langkah ke-4
Launch ...

```
$ java IfElseApp  
Persamaan kuadrat  $a*x^2+b*x+c=0$ , di mana  
a = 4.0  
b = -13.0  
c = -12.0  
dengan menerapkan rumus ABC diketahui bahwa :  
Akar-akar persamaan nyata dan berbeda.  
x1 = 4.0 x2 = -0.75
```

CS-051-072

Merupakan contoh bagaimana menyusun kendali aliran dengan switch ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis SwitchCaseApp.java, simpan di directory yang sudah dipersiapkan.

SwitchCaseApp.java

```
public class SwitchCaseApp
{
    public static void main(String[] args)
    {
        char inisial = 'M';
        String nama = null;
        switch (inisial)
        {
            case 'A' : nama = "Agung";break;
            case 'E' : nama = "Eko";break;
            case 'F' : nama = "Fikri";break;
            case 'M' : nama = "Millati";break;
            case 'S' : nama = "Syarif";break;
            case 'T' : nama = "Toosa";break;
            case 'Y' : nama = "Yudo";break;
            default  : nama = "Tak dikenal";
        }

        System.out.println("Nama Anda adalah " + nama);
    }
}
```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-

Compile ...

```
$ javac SwitchCaseApp.java
```

akan dihasilkan SwitchCaseApp.class

Langkah ke-4

Launch ...

```
$ java SwitchCaseApp
Nama Anda adalah Millati
```

CS-051-073

Merupakan contoh bagaimana menyusun perulangan dengan for

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis ForApp.java, simpan di directory yang sudah dipersiapkan.

ForApp.java

```
public class ForApp
{
    public static void main(String args[])
    {
        int bilangan = 5;

        int hasil = 1;
        for(int iterator=2;iterator<=bilangan;iterator++)
        {
            System.out.print
                (hasil + " * " + iterator + " = ");
            hasil *= iterator;
            System.out.println(hasil);
        }
        System.out.println
            ( bilangan + " ! = " + hasil);
    }
}
```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-

Compile ...

```
$ javac ForApp.java
```

akan dihasilkan ForApp.class

Langkah ke-4

Launch ...

```
$ java ForApp
1 * 2 = 2
2 * 3 = 6
6 * 4 = 24
24 * 5 = 120
5 ! = 120
```

CS-051-074

Merupakan contoh bagaimana menyusun perulangan dengan do-while ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis DoWhileApp.java, simpan di directory yang sudah dipersiapkan.

DoWhileApp.java

```
public class DoWhileApp
{
    public static void main(String args[])
    {
        int bilangan = 5;

        int x = 2;
        int hasil = 1;

        do
        {
            System.out.print
                (hasil + " * " + x + " = ");
            hasil = hasil*x;
            System.out.println(hasil);
            x++;
        }
        while(x<=bilangan);

        System.out.println
            ( bilangan + " ! = " + hasil);
    }
}
```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-

Compile ...

```
$ javac DoWhileApp.java
```

akan dihasilkan DoWhileApp.class

Langkah ke-4

Launch ...

```
$ java DoWhileApp
1 * 2 = 2
2 * 3 = 6
6 * 4 = 24
24 * 5 = 120
5 ! = 120
```

CS-051-075

Merupakan contoh bagaimana menyusun perulangan dengan while ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis WhileApp.java, simpan di directory yang sudah dipersiapkan.

WhileApp.java

```
public class WhileApp
{
    public static void main(String args[])
    {
        int bilangan = 5;

        int x = 2;
        int hasil = 1;

        while(x<=bilangan)
        {
            System.out.print
                (hasil + " * " + x + " = ");
            hasil = hasil*x;
            System.out.println(hasil);
            x++;
        }
        System.out.println
            ( bilangan + " ! = " + hasil);
    }
}
```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-

Compile ...

```
$ javac WhileApp.java
```

akan dihasilkan WhileApp.class

Langkah ke-4

Launch ...

```
$ java WhileApp
1 * 2 = 2
2 * 3 = 6
6 * 4 = 24
24 * 5 = 120
5 ! = 120
```


8Method

Method memuat prosedur ...

Passing parameter adalah by reference ...

9Penanganan Exception

Exception dimunculkan dalam program saat menemukan keadaan yang bisa menjadikan program mengalami error. Sebuah exception akan menjadikan aliran program keluar dari aliran yang normal.

Sebagai contoh, exception dimunculkan jika sebuah program mencoba meng-invoke method dari sebuah obyek yang belum di-instantiate. Exception juga dimunculkan jika sebuah program mencoba membaca file yang tidak ada. Exception juga dimunculkan jika Anda mencoba meng-execute SQL statement yang salah.

Sebuah program Java perlu dirancang untuk bisa menghadapi exception yang bisa terjadi. Sebuah program yang mengakses database melalui JDBC, misalnya perlu dirancang untuk menghadapi exception yang bisa terjadi saat mengakses database tsb.

Untuk menangani Anda bisa menggunakan blok try-catch.

Cara lainnya Anda juga bisa men-throw exception tersebut dengan menuliskannya dalam deklarasi method dimana exception tsb bisa terjadi. Dalam rancangan ini, maka method yang meng-invoke method ini mesti menangani exception tsb.

Lebih lanjut, Anda juga bisa membuat sendiri class Exception baru, dengan meng-extends Exception. Anda bisa menggunakannya untuk men-throw Exception yang sesuai dengan business process dari aplikasi Anda.

10 Pemrograman Berorientasi Obyek

Di dalam Java, modul program dimodelkan, dirancang dan ditulis sebagai obyek. Kerangka berpikirnya berbeda dengan pemrograman tradisional. Pemrograman tradisional memisahkan antara data, dan prosedur yang mengolah data tersebut. Sedangkan di dalam Java, data dan prosedur ini dipadukan sebagai sebuah obyek.

Ini lebih alami. Di alam, berbagai benda, baik yang real maupun abstrak, mempunyai dua hal yang terpadu yaitu ukuran dan tingkah laku. Sebuah perusahaan mempunyai ukuran seperti jumlah karyawan, dan mempunyai tingkah laku seperti mengangkat karyawan. Seekor burung yang sedang terbang mempunyai ukuran seperti ketinggian, dan bisa menukik sebagai tingkah laku.

Sebagaimana obyek di alam yang mempunyai ukuran dan tingkah laku, sebuah obyek di dalam program juga bisa mempunyai data dan prosedur. Di dalam Java, rancangan sebuah obyek ditulis dalam sebuah class. Di dalam class data diimplementasikan sebagai member variable, sedangkan prosedur yang mengolah data tersebut diimplementasikan dalam method.

Class ini bisa di-instantiate, dengan memanggil constructor-nya. Instance dari sebuah class disebut sebagai obyek.

Dalam perkembangannya, pemrograman berorientasi obyek, terdapat 3 ciri utama, yaitu encapsulation, inheritance dan polymorphism.

Encapsulation adalah sebuah teknik untuk menyembunyikan detail dari client.

Inheritance adalah sebuah teknik untuk membuat sebuah obyek yang menurunkan data dan prosedur yang dimiliki oleh obyek lain. Obyek yang ada terlebih dahulu disebut super class. Sedangkan obyek yang diturunkan darinya disebut sub class.

Polymorphism adalah sebuah teknik yang membolehkan sebuah obyek mempunyai beragam implementasi.

CS-051-101

Merupakan contoh menuliskan obyek ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis MyDouble.java, simpan di directory yang sudah dipersiapkan.

MyDouble.java

```
public class MyDouble
{
    private double d;
    public MyDouble(double d)
    {
        this.d = d;
    }

    public MyDouble add(double x)
    {
        this.d += x;
        return this;
    }

    public MyDouble subscribe(double x)
    {
        this.d -= x;
        return this;
    }

    public MyDouble multiply(double x)
    {
        this.d *= x;
        return this;
    }

    public MyDouble divide(double x)
    {
        this.d /= x;
        return this;
    }

    public String toString()
    {
        return String.valueOf(d);
    }
}
```

Langkah ke-2

Tulis MyDoubleApp.java, simpan di directory yang sudah dipersiapkan.

MyDoubleApp.java

```
public class MyDoubleApp
```

```
{
    public static void main(String[] args)
    {
        MyDouble myDouble = new MyDouble(7.5);
        myDouble
            .add(12.5).divide(5)
            .subscribe(2.5).multiply(100);
        System.out.println(myDouble);
    }
}
```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac MyDoubleApp.java
```

akan dihasilkan MyDoubleApp.class dan MyDouble.clas

Langkah ke-5

Launch ...

```
$ java MyDoubleApp
150.0
```

11Interface dan Abstract Class

...

12Java Bean

Java Bean adalah teknik untuk merepresentasikan satu struktur data dalam Java. Secara sederhana sebuah Java Bean mempunyai property, dan setter serta getter method untuk masing-masing property tersebut.

13Composition, Aggregation

14Format

Library untuk melakukan mem-format data tersedia dalam package `java.util`. Dengan library ini Anda bisa mem-format angka, tanggal dll. Class yang digunakan di antaranya adalah `DecimalFormat` dan `SimpleDateFormat`.

15File

Dua class yang lumrah digunakan untuk bekerja dengan file di dalam Java adalah File dan RandomAccessFile.

CS-051-151

Merupakan contoh menggunakan File, untuk me-list sebuah directory.

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis ListApp.java, simpan di directory yang sudah dipersiapkan.

ListApp.java

```
import java.io.File;

public class ListApp
{
    public static void main(String[] args)
    {
        String directoryName = args[0];
        File directory = new File(directoryName);

        String[] list = directory.list();

        if(list != null)
            for(int i = 0;i<list.length;i++)
            {
                System.out.println(list[i]);
            }
    }
}
```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac ListApp.java
```

akan dihasilkan ListApp.class

Langkah ke-4

Launch ...

```
$ java ListApp /home/lab/jboss-3.0.4/
bin
client
docs
lib
server
```

CS-051-152

Merupakan contoh menggunakan File, untuk mendapatkan informasi tentang sebuah file

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis FileInfoApp.java, simpan di directory yang sudah dipersiapkan.

FileInfoApp.java

```
import java.io.File;
import java.util.Date;

public class FileInfoApp
{
    public static void main(String[] args)
    {
        String fileName = args[0];
        File file = new File(fileName);

        if(file.isFile())
        {
            System.out.println("is a file");
        }
        else
        {
            System.out.println("is a directory");
        }

        System.out.println("can read : " + file.canRead());
        System.out.println("can write : " + file.canWrite());
        System.out.println("length : " + file.length());
        System.out.println("last modified : "
            + new Date(file.lastModified()));
    }
}
```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac FileInfoApp.java
```

akan dihasilkan FileInfoApp.java

Langkah ke-4

Launch ...

```
$ java FileInfoApp /home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-bin.jar
is a file
```

can read : true
can write : true
length : 112241
last modified : Mon Feb 10 21:26:01 SGT 2003

CS-051-153

Merupakan contoh

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis MyFileCommand.java, simpan di directory yang sudah dipersiapkan.

MyFileCommand.java

```
import java.io.*;

public class MyFileCommand
{
    private MyFileCommand(){};

    public static void ls(String directoryName)
    {
        File directory = new File(directoryName);
        if(directory.exists())
        {
            String[] list = directory.list();
            for(int i = 0;i<list.length;i++)
            {
                System.out.println(list[i]);
            }
        }
        else
        {
            System.out.println
                ("can not list directory '"
                + directoryName
                + "' : No such file or directory");
        }
    }

    public static void view(String fileName)
        throws IOException
    {
        RandomAccessFile raf = new RandomAccessFile(fileName, "r");
        String line;
        while((line = raf.readLine()) != null)
        {
            System.out.println(line);
        }
        raf.close();
    }

    public static void mkdir(String directoryName)
    {
        File directory = new File(directoryName);
        if(directory.exists())
        {
            System.out.println
                ("can not create directory '"
```

```

        + directoryName + "' : File exists");
    }
    else
        directory.mkdirs();
}

public static void mv(String oldPath, String newPath)
{
    File oldFile = new File(oldPath);

    if(oldFile.exists())
    {
        File newFile = new File(newPath);
        oldFile.renameTo(newFile);
    }
    else
    {
        System.out.println
            ("can not move directory '"
            + oldPath + "' : No such file or directory");
    }
}

public static void rm(String path)
{
    File file = new File(path);

    if(file.exists())
    {
        file.delete();
    }
    else
    {
        System.out.println
            ("can not remove directory '"
            + path + "' : No such file or directory");
    }
}
}
}

```

Langkah ke-2

Tulis FileCommandApp.java, simpan di directory yang sudah dipersiapkan.

FileCommandApp.java

```

import java.io.File;

public class FileCommandApp
{
    public static void main(String[] args)
        throws Exception
    {
        if(args.length==0)
        {
            usage();
        }
        else
            if(args[0].equals("ls") && args.length>=2)
            {
                MyFileCommand.ls(args[1]);
            }
        else
    }
}

```

```

        if(args[0].equals("view") && args.length>=2)
        {
            MyFileCommand.view(args[1]);
        }
        else
        if(args[0].equals("mkdir") && args.length>=2)
        {
            MyFileCommand.mkdir(args[1]);
        }
        else
        if(args[0].equals("mv") && args.length>=3)
        {
            MyFileCommand.mv(args[1], args[2]);
        }
        else
        if(args[0].equals("rm") && args.length>=2)
        {
            MyFileCommand.rm(args[1]);
        }
        else
        {
            usage();
        }
    }

    private static void usage()
    {
        System.out.println
        (
            "List directory contents : \n" +
            "    java FileCommandApp ls directoryName\n" +
            "View text file : \n" +
            "    java FileCommandApp view fileName\n" +
            "Make directory : \n" +
            "    java FileCommandApp mkdir directoryName\n" +
            "Rename files : \n" +
            "    java FileCommandApp mv oldPath newPath\n" +
            "Remove files or directory : \n" +
            "    java FileCommandApp rm path\n"
        );
    }
}

```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac FileCommandApp.java
```

akan dihasilkan MyFileCommand.class dan FileCommandApp.class

Langkah ke-5

Launch ...

```
$ java FileCommandApp
List directory contents :
    java FileCommandApp ls directoryName
View text file :
```



```
        java FileCommandApp view fileName
Make directory :
        java FileCommandApp mkdir directoryName
Rename files :
        java FileCommandApp mv oldPath newPath
Remove files or directory :
        java FileCommandApp rm path
```

menjalankan ls atas sebuah directory ...

```
$ java FileCommandApp ls .
FileCommandApp.java
MyFileCommand.java
MyFileCommand.class
FileCommandApp.class
```

CS-051-154

Merupakan contoh

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis EncryptionService.java, simpan di directory yang sudah dipersiapkan.

EncryptionService.java

```
import java.io.*;

public class EncryptionService
{
    public static void service
        (String srcFileName,
         String destinationFileName,
         String password)
        throws IOException
    {
        FileInputStream fis
            = new FileInputStream(srcFileName);
        FileOutputStream fos
            = new FileOutputStream(destinationFileName);

        byte[] passwordBytes = password.getBytes();

        int oneByte;
        while ((oneByte = fis.read()) != -1)
        {
            for(int i=0;i<passwordBytes.length;i++)
            {
                oneByte ^= passwordBytes[i];
            }
            fos.write(oneByte);
        }

        fis.close();
        fos.flush();
        fos.close();
    }
}
```

Langkah ke-2

Tulis EncryptionApp.java, simpan di directory yang sudah dipersiapkan.

EncryptionApp.java

```
public class EncryptionApp
{
    public static void main(String[] args)
        throws Exception
```

```

    {
        if (args.length >= 3)
        {
            EncryptionService.service(args[0], args[1], args[2]);
        }
        else
        {
            System.out.println(
                "Cara penggunaan : "
                + "java EncryptionApp source destination password"
                );
        }
    }
}

```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac EncryptionApp.java
```

akan dihasilkan EncryptionService.class dan EncryptionApp.class

Langkah ke-5

... bisa digunakan untuk meng-encrypt file ascii maupun binary ...

Tulis sebuah doc, misal dengan Open Office, simpan sebagai MyFile.doc

jalankan :

```
$ java EncryptionApp MyFile.doc MyFile2.doc j2ee
```

Coba buka MyFile2.doc dengan Open Office ... file MyFile2.doc adalah file yang ter-encrypt

jalankan :

```
$ java EncryptionApp MyFile2.doc MyFile3.doc j2ee
```

Dan Anda bisa membuka MyFile3.doc yang isinya sama dengan MyFile.doc

CS-051-155

Merupakan contoh

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis LineInverseService.java, simpan di directory yang sudah dipersiapkan.

LineInverseService.java

```
import java.io.*;

public class LineInverseService
{
    public static void service
        (String sourceFileName, String destinationFileName)
        throws IOException
    {
        RandomAccessFile sourceRaf
            = new RandomAccessFile(sourceFileName, "r");
        RandomAccessFile destinationRaf
            = new RandomAccessFile(destinationFileName, "rw");

        String sourceLine;
        while((sourceLine = sourceRaf.readLine()) != null)
        {
            String destinationLine = "";

            for(int end=sourceLine.length()-1;end>=0;end--)
            {
                destinationLine += sourceLine.charAt(end);
            }
            destinationRaf.writeBytes(destinationLine);
            destinationRaf.writeBytes("\n");
        }
        destinationRaf.close();
        sourceRaf.close();
    }
}
```

Langkah ke-2

Tulis LineInverseApp.java, simpan di directory yang sudah dipersiapkan.

LineInverseApp.java

```
public class LineInverseApp
{
    public static void main(String[] args)
        throws Exception
    {
        if(args.length>=2)
        {
            LineInverseService.service(args[0], args[1]);
        }
    }
}
```

```

        else
        {
            System.out.println(
                "Cara penggunaan : "
                + "java LineInverseApp source destination"
            );
        }
    }
}

```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac LineInverseApp.java
```

akan dihasilkan LineInverseService.class dan LineInverseApp.class

Langkah ke-5

Launch ... meng-invert file LineInverseApp.java dan disimpan sebagai X.txt

```

$ java LineInverseApp LineInverseApp.java X.txt
$ more X.txt
ppAesrevnIeniL ssalc cilbup
{
)sgra ][gnirtS(niam dioV citats cilbup
noitpecxE sworht
{
)2=>htgnel.sgra(fi
{
;)][sgra ,]0[sgra(ecivres.ecivreSesrevnIeniL
}
}
esle
{
(nltnirp.tuo.metsyS
" : naanuggnep araC"
"noitanitsed ecruos ppAesrevnIeniL avaj" +
;)
}
}
}
}

```

CS-051-156

Merupakan contoh

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis TelephoneService.java, simpan di directory yang sudah dipersiapkan.

TelephoneService.java

```
import java.util.*;
import java.io.*;

public class TelephoneService
{
    private String dbFileName;
    private Hashtable table;

    public TelephoneService(String dbFileName)
        throws IOException, ClassNotFoundException
    {
        this.dbFileName = dbFileName;
        load();
    }

    public void load()
        throws IOException, ClassNotFoundException
    {
        File file = new File(dbFileName);
        if(file.exists())
        {
            FileInputStream fis
                = new FileInputStream(dbFileName);
            ObjectInputStream ois
                = new ObjectInputStream(fis);
            table = (Hashtable) ois.readObject();
        }
        else
        {
            table = new Hashtable();
        }
    }

    public void store()
        throws IOException
    {
        FileOutputStream fis
            = new FileOutputStream(dbFileName);
        ObjectOutputStream ois
            = new ObjectOutputStream(fis);
        ois.writeObject(table);
    }

    public void create(String name, String phoneNo)
        throws IOException, ClassNotFoundException
    {
```

```

        table.put(name, phoneNo);
        store();
    }

    public boolean view(String name)
        throws IOException, ClassNotFoundException
    {
        String phoneNo = (String)table.get(name);
        if(phoneNo != null)
        {
            System.out.println("Phone No : " + phoneNo);
            return true;
        }
        else
        {
            return false;
        }
    }

    public boolean remove(String name)
        throws IOException, ClassNotFoundException
    {
        if(table.containsKey(name))
        {
            table.remove(name);
            store();
            return true;
        }
        else
        {
            return false;
        }
    }

    public void viewAll()
        throws IOException, ClassNotFoundException
    {
        Enumeration names = table.keys();
        while(names.hasMoreElements())
        {
            String name = (String) names.nextElement();
            String phoneNo = (String) table.get(name);
            System.out.println(name + " " + phoneNo);
        }
    }
}

```

Langkah ke-2

Tulis TelephoneApp.java, simpan di directory yang sudah dipersiapkan.

TelephoneApp.java

```

import java.io.*;

public class TelephoneApp
{
    public static void main(String[] args)
        throws Exception
    {
        if(args.length>0)

```

```

    {
        TelephoneApp app
            = new TelephoneApp(args[0]);
        char command;
        do
        {
            System.out.println();
            System.out.println("Press Enter to continue ...");
            app.readLine();
            System.out.println("Available command : ");
            System.out.println("C Create a new phone number");
            System.out.println("A View all phone number");
            System.out.println("V View a phone number");
            System.out.println("R Remove a phone number");
            System.out.println("X Exit");
            System.out.println();

            command
                = app.readLine("Press C, A, V, R, or X")
                    .toUpperCase().charAt(0);

            System.out.println();

            switch(command)
            {
                case 'C' : app.create();break;
                case 'V' : app.view();break;
                case 'A' : app.viewAll();break;
                case 'R' : app.remove();break;
                case 'X' : break;
                default :
                    System.out.println("You press invalid key");
            }
        }
        while(command != 'X');
    }
    else
    {
        System.out.println(
            "Cara penggunaan : java TelephoneService dbFileName"
        );
    }
}

private BufferedReader in;
private TelephoneService service;

public TelephoneApp(String dbFileName)
    throws Exception
{
    in = new BufferedReader(new InputStreamReader(System.in));
    service = new TelephoneService(dbFileName);
}

private char readChar()
    throws IOException
{
    char ch = (char) in.read();
    return ch;
}

private void readLine()
    throws IOException
{

```



```

        in.readLine();
    }

    private String readLine(String parameterName)
        throws IOException
    {
        System.out.print(parameterName + " : ");
        return in.readLine();
    }

    public void create()
        throws Exception
    {
        System.out.println("Please enter Name and Phone No to create");
        String name = readLine("Name");
        String phoneNo = readLine("Phone No");
        service.create(name, phoneNo);
        System.out.println("Created successfully");
    }

    public void view()
        throws Exception
    {
        System.out.println("Please enter Name to view");
        String name = readLine("Name");
        if(!service.view(name))
        {
            System.out.println("No such Name");
        }
    }

    public void remove()
        throws Exception
    {
        System.out.println("Please enter Name to remove");
        String name = readLine("Name");
        if(service.remove(name))
            System.out.println("Removed successfully");
        else
            System.out.println("No such Name");
    }

    public void viewAll()
        throws Exception
    {
        System.out.println("All ...");
        service.viewAll();
    }
}

```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

```
$ javac TelephoneApp.java
```

akan dihasilkan TelephoneService.class dan TelephoneApp.class

Langkah ke-5

Launch ...

```
$ java TelephoneApp myphone.data
```

```
Press Enter to continue ...
```

```
Available command :
```

```
C Create a new phone number
```

```
A View all phone number
```

```
V View a phone number
```

```
R Remove a phone number
```

```
X Exit
```

```
Press C, A, V, R, or X :
```

16mySQL

Java Database Connectivity (JDBC) merupakan salah satu teknologi inti di dalam Java

CS-051-161

Merupakan latihan untuk meng-create database ...

Persiapan

Anda perlu mempunyai akses ke MySQL server yang berjalan ...

Langkah

Langkah ke-1

Buka terminal ...

Langkah ke-2

Login ke MySQL sebagai root ...

```
$ mysql -uroot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.23.54

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

Langkah ke-3

Create database ...

```
mysql> create database MYAPP_DB;
Query OK, 1 row affected (0.49 sec)
```

Langkah ke-3

meng-create user dengan hak akses ke MYAPP_DB

```
mysql> GRANT ALL PRIVILEGES ON MYAPP_DB.* TO ekobs@localhost IDENTIFIED
BY 'j2ee';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON MYAPP_DB.* TO ekobs@localhost.localdomain
IDENTIFIED BY 'j2ee';
Query OK, 0 rows affected (0.13 sec)
```

CS-051-162

Merupakan latihan untuk meng-create table ...

Persiapan

Anda perlu mempunyai akses ke MySQL server yang berjalan ...

Langkah

Langkah ke-1

Buka terminal ...

Langkah ke-2

Login ke MySQL sebagai user ...

```
$ mysql -uekobs -pj2ee MYAPP_DB
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 3.23.54
```

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

Langkah ke-3

Create table ...

```
mysql> create table EMPLOYEE_TBL(ID varchar(10) primary key, NAME
varchar(40), DEPARTMENT varchar(20),JOB_TITLE varchar(30), HIRE_DATE
date, PERMANENT_EMPLOYEE bool, SALARY float);
Query OK, 0 rows affected (0.00 sec)
```

CS-051-163

Merupakan latihan untuk menjalankan perintah SQL : INSERT, SELECT, UPDATE, DELETE

Persiapan

Anda perlu mempunyai akses ke mySQL server yang berjalan ...

Langkah

Langkah ke-1

Buka terminal ...

Langkah ke-2

Login ke mySQL sebagai user ...

```
$ mysql -uekobs -pj2ee MYAPP_DB
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 3.23.54

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

Langkah ke-3

Insert ...

```
mysql> INSERT INTO EMPLOYEE_TBL (ID, NAME, DEPARTMENT, JOB_TITLE,
HIRE_DATE,PERMANENT_EMPLOYEE,SALARY ) VALUES ('033','Wawan Setiawan',
'Technical Support', 'System Admin', '2003-01-31',0, 5000000);
Query OK, 1 row affected (0.00 sec)
```

Langkah ke-4

Select ...

```
mysql> select NAME, DEPARTMENT, JOB_TITLE from EMPLOYEE_TBL where
ID='033';
+-----+-----+-----+
| NAME          | DEPARTMENT          | JOB_TITLE          |
+-----+-----+-----+
| Wawan Setiawan | Technical Support    | System Admin       |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Langkah ke-5

Update ...

```
mysql> update EMPLOYEE_TBL set DEPARTMENT='E-Commerce',
JOB_TITLE='Software Engineer' where ID='033';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select NAME, DEPARTMENT, JOB_TITLE from EMPLOYEE_TBL where
ID='033';
+-----+-----+-----+
| NAME          | DEPARTMENT          | JOB_TITLE          |
+-----+-----+-----+
| Wawan Setiawan | E-Commerce          | Software Engineer  |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Langkah ke-5

Delete ...

```
mysql> delete from EMPLOYEE_TBL where ID='033';  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select NAME, DEPARTMENT, JOB_TITLE from EMPLOYEE_TBL where  
ID='033';  
Empty set (0.00 sec)
```

17JDBC

Java Database Connectivity (JDBC) merupakan teknologi inti di dalam Java untuk mengakses dengan database. JDBC digunakan baik dalam aplikasi desktop, aplikasi Web maupun aplikasi enterprise.

JDBC menyediakan satu cara untuk mengakses database secara independen terhadap database vendor. Melalui JDBC Anda bisa mengakses mySQL dengan program yang sama untuk mengakses Oracle, UDB2 dan PostgreSQL. Ini mendukung semangat Java untuk write once, run anywhere.

Kunci utama dari JDBC adalah JDBC Driver yang bertanggung jawab menjadi jembatan antara program Java dan database tertentu. JDBC Driver untuk mySQL, misalnya, bertanggung jawab untuk menjembatani antara program yang ditulis dengan Java ke mySQL RDBMS.

Untuk mengakses database melalui JDBC, pertama Anda perlu mendapatkan obyek Connection. Selanjutnya Anda bisa mendapatkan obyek-obyek seperti Statement dan PreparedStatement. Melalui Statement, dan PreparedStatement, Anda bisa mengirimkan perintah SQL ke database, seperti INSERT, SELECT, UPDATE dan DELETE.

CS-051-171

Merupakan latihan untuk membuat JDBC connection ke MySQL database server.

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava. Anda membutuhkan JDBC Driver ke MySQL.

Dalam contoh ini file JDBC Driver yang digunakan adalah :
/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-bin.jar

Langkah

Langkah ke-1

Tulis ConnectionApp.java, simpan di directory yang sudah dipersiapkan.

ConnectionApp.java

```
import java.io.*;
import java.sql.*;

public class ConnectionApp
{
    public static void main(String[] args)
    {
        Connection conn = null;
        try
        {
            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

            String url = "jdbc:mysql://localhost:3306/MYAPP_DB";
            String user = "ekobs";
            String pwd = "j2ee";

            System.out.println
                ("Mencoba membangun connection ke '" + url
                 + "' dengan user '" + user
                 + "' dan password '" + pwd + "' ...");
            conn = DriverManager.getConnection(
                url, user, pwd);
            System.out.println("Success.");
        }
        catch(ClassNotFoundException cnfe)
        {
            cnfe.printStackTrace();
        }
        catch(SQLException sqle)
        {
            sqle.printStackTrace();
        }
        finally
        {
            try
            {
                if(conn != null)
                {
                    System.out.println("Menutup connection.");
                }
            }
        }
    }
}
```

```

        conn.close();
    }
}
catch(SQLException sqle)
{
    sqle.printStackTrace();
}
}
}
}

```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac ConnectionApp.java
```

akan dihasilkan ConnectionApp.java

Langkah ke-4

Set CLASSPATH agar memuat JDBC driver yang dibutuhkan untuk mengakses mySQL database server ...

```
$ export CLASSPATH=.
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-
bin.jar
```

Langkah ke-5

Launch ...

```
$ java ConnectionApp
```

Mencoba membangun connection ke 'jdbc:mysql://localhost:3306/MYAPP_DB' dengan user 'ekobs' dan password 'j2ee' ...

Success.

Menutup connection.

```

lab@localhost:~ - Shell - Konsole
Session Edit View Settings Help
$ java ConnectionApp
Mencoba membangun connection ke 'jdbc:mysql://localhost:3306/MYAPP_DB' dengan user 'ekobs' dan password 'j2ee' ...
Success.
Menutup connection.
$

```

CS-051-172

Merupakan latihan untuk meng-insert data ke sebuah table di sebuah database ...

Persiapan

Anda perlu melakukan praktikum CS-051-171 terlebih dahulu. Directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis InsertApp.java, simpan di directory yang sudah dipersiapkan.

InsertApp.java

```
import java.io.*;
import java.sql.*;

public class InsertApp
{
    public static void main(String[] args)
    {
        Connection conn = null;
        try
        {
            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

            String url = "jdbc:mysql://localhost:3306/MYAPP_DB";
            String user = "ekobs";
            String pwd = "j2ee";
            conn = DriverManager.getConnection(
                url, user, pwd);

            String id = "032";
            String name = "Muhammad Sigit Arifianto";
            String department = "Business Development";
            String jobTitle = "Account Manager";
            String hireDate = "2000-12-30";
            byte permanentEmployee = 1;
            double salary = 8500000;

            String sqlInsert =
                "INSERT INTO EMPLOYEE_TBL "
                + "("
                + "ID, NAME, DEPARTMENT, JOB_TITLE, "
                + "HIRE_DATE, PERMANENT_EMPLOYEE, SALARY "
                + ")"
                + " VALUES "
                + "("
                + "'" + id + "',"
                + "'" + name + "',"
                + "'" + department + "',"
                + "'" + jobTitle + "',"
                + "'" + hireDate + "',"
                + permanentEmployee + ","
                + salary
                + ")";
            System.out.println("mysql >");
```

```

        System.out.println(sqlInsert);

        Statement stmt = conn.createStatement();
        int count = stmt.executeUpdate(sqlInsert);
        System.out.println();
        System.out.println("Query OK, " + count + " row affected");
    }
    catch(ClassNotFoundException cnfe)
    {
        cnfe.printStackTrace();
    }
    catch(SQLException sqle)
    {
        sqle.printStackTrace();
    }
    finally
    {
        try
        {
            if(conn != null)
            {
                conn.close();
            }
        }
        catch(SQLException sqle)
        {
            sqle.printStackTrace();
        }
    }
}
}
}

```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac InsertApp.java
```

akan dihasilkan InsertApp.java

Langkah ke-4

Set CLASSPATH agar memuat JDBC driver yang dibutuhkan untuk mengakses MySQL database server ...

```
$ export CLASSPATH=.
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-
bin.jar
```

Langkah ke-5

Launch ...

```
$ java InsertApp
mysql >
INSERT INTO EMPLOYEE_TBL (ID, NAME, DEPARTMENT, JOB_TITLE, HIRE_DATE,
PERMANENT_EMPLOYEE, SALARY ) VALUES ('032','Muhammad Sigit
```

Arifianto','Business Development','Account Manager','2000-12-30',1,
8500000.0)

Query OK, 1 row affected

CS-051-173

Merupakan latihan untuk meng-select data dari sebuah table di sebuah database ...

Persiapan

Anda perlu melakukan praktikum CS-051-171 terlebih dahulu. Directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis SelectApp.java, simpan di directory yang sudah dipersiapkan.

SelectApp.java

```
import java.io.*;
import java.sql.*;
import java.text.*;

public class SelectApp
{
    public static void main(String[] args)
    {
        Connection conn = null;
        try
        {
            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

            String url = "jdbc:mysql://localhost:3306/MYAPP_DB";
            String user = "ekobs";
            String pwd = "j2ee";
            conn = DriverManager.getConnection(
                url, user, pwd);

            String sqlSelect =
                "SELECT * FROM EMPLOYEE_TBL ";
            Statement stmt = conn.createStatement();

            ResultSet rs = stmt.executeQuery(sqlSelect);

            SimpleDateFormat format
                = new SimpleDateFormat("dd/MMMM/yyyy");
            while(rs.next())
            {
                String id = rs.getString("ID");
                String name = rs.getString("NAME");
                String department = rs.getString("DEPARTMENT");
                Date hireDate = rs.getDate("HIRE_DATE");
                boolean permanentEmployee
                    = rs.getBoolean("PERMANENT_EMPLOYEE");
                double salary = rs.getDouble("SALARY");

                System.out.println("E M P L O Y E E");
                System.out.println("Id : " + id);
                System.out.println("Name : " + name);
                System.out.println
```

```

        ("Department :" + department);
        System.out.println
        ("Hire Date : " + format.format(hireDate));
        System.out.println
        ("Permanent Emp.: " + permanentEmployee);
        System.out.println("Salary : " + salary);
        System.out.println();
    }
}
catch(ClassNotFoundException cnfe)
{
    cnfe.printStackTrace();
}
catch(SQLException sqle)
{
    sqle.printStackTrace();
}
finally
{
    try
    {
        if(conn != null)
        {
            conn.close();
        }
    }
    catch(SQLException sqle)
    {
        sqle.printStackTrace();
    }
}
}
}

```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac SelectApp.java
```

akan dihasilkan SelectApp.java

Langkah ke-4

Set CLASSPATH agar memuat JDBC driver yang dibutuhkan untuk mengakses MySQL database server ...

```
$ export CLASSPATH=.
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-
bin.jar
```

Langkah ke-5

Launch ...

```
$ java SelectApp
```

```

E M P L O Y E E
Id                :          033

```

Name : Wawan Setiawan
Department : Technical Support
Hire Date : 31/January/2003
Permanent Emp. : false
Salary : 5000000.0

E M P L O Y E E

Id : 032
Name : Muhammad Sigit Arifianto
Department : Business Development
Hire Date : 30/December/2000
Permanent Emp. : true
Salary : 8500000.0

CS-051-174

Merupakan latihan untuk meng-update data di sebuah table di sebuah database ...

Persiapan

Anda perlu melakukan praktikum CS-051-171 terlebih dahulu. Directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis UpdateApp.java, simpan di directory yang sudah dipersiapkan.

UpdateApp.java

```
import java.io.*;
import java.sql.*;

public class UpdateApp
{
    public static void main(String[] args)
    {
        Connection conn = null;
        try
        {
            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

            String url = "jdbc:mysql://localhost:3306/MYAPP_DB";
            String user = "ekobs";
            String pwd = "j2ee";
            conn = DriverManager.getConnection(
                url, user, pwd);

            String id = "032";
            String department = "Professional Service";
            String jobTitle = "Software Engineer";

            String sqlUpdate =
                "UPDATE EMPLOYEE_TBL "
                + "SET "
                + "DEPARTMENT = '" + department + "', "
                + "JOB_TITLE = '" + jobTitle + "' "
                + "WHERE "
                + "ID = '" + id + "'";

            System.out.println("mysql >");
            System.out.println(sqlUpdate);

            Statement stmt = conn.createStatement();
            int count = stmt.executeUpdate(sqlUpdate);
            System.out.println();
            System.out.println
                ("Query OK, " + count + " row affected");
        }
        catch(ClassNotFoundException cnfe)
        {
            cnfe.printStackTrace();
        }
    }
}
```

```

    }
    catch(SQLException sqle)
    {
        sqle.printStackTrace();
    }
    finally
    {
        try
        {
            if(conn != null)
            {
                conn.close();
            }
        }
        catch(SQLException sqle)
        {
            sqle.printStackTrace();
        }
    }
}
}
}

```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac UpdateApp.java
```

akan dihasilkan UpdateApp.java

Langkah ke-4

Set CLASSPATH agar memuat JDBC driver yang dibutuhkan untuk mengakses mySQL database server ...

```
$ export CLASSPATH=.
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-
bin.jar
```

Langkah ke-6

Launch ...

```
$ java UpdateApp
mysql >
UPDATE EMPLOYEE_TBL SET DEPARTMENT = 'Professional Service',JOB_TITLE =
'Software Engineer' WHERE ID = '032'
```

Query OK, 1 row affected

CS-051-175

Merupakan latihan untuk men-delete data dari sebuah table di sebuah database ...

Persiapan

Anda perlu melakukan praktikum CS-051-171 terlebih dahulu. Directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis DeleteApp.java, simpan di directory yang sudah dipersiapkan.

DeleteApp.java

```
import java.io.*;
import java.sql.*;

public class DeleteApp
{
    public static void main(String[] args)
    {
        Connection conn = null;
        try
        {
            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

            String url = "jdbc:mysql://localhost:3306/MYAPP_DB";
            String user = "ekobs";
            String pwd = "j2ee";
            conn = DriverManager.getConnection(
                url, user, pwd);

            String id = "032";
            String sqlDelete =
                "DELETE FROM EMPLOYEE_TBL "
                + "WHERE "
                + "ID = '" + id + "'";

            System.out.println("mysql >");
            System.out.println(sqlDelete);

            Statement stmt = conn.createStatement();
            int count = stmt.executeUpdate(sqlDelete);
            System.out.println();
            System.out.println("Query OK, " + count + " row affected");

        }
        catch(ClassNotFoundException cnfe)
        {
            cnfe.printStackTrace();
        }
        catch(SQLException sqle)
        {
            sqle.printStackTrace();
        }
        finally
        {
```

```

        try
        {
            if(conn != null)
            {
                conn.close();
            }
        }
        catch(SQLException sqle)
        {
            sqle.printStackTrace();
        }
    }
}
}

```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac DeleteApp.java
```

akan dihasilkan DeleteApp.java

Langkah ke-4

Set CLASSPATH agar memuat JDBC driver yang dibutuhkan untuk mengakses mySQL database server ...

```
$ export CLASSPATH=.
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-
bin.jar
```

Langkah ke-5

Launch ...

```
$ java DeleteApp
mysql >
DELETE FROM EMPLOYEE_TBL WHERE ID = '032'
```

Query OK, 1 row affected

CS-051-176

Merupakan latihan untuk meng-insert data dari sebuah table di sebuah database ...

Persiapan

Anda perlu melakukan praktikum CS-051-171 terlebih dahulu. Directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis InsertApp.java, simpan di directory yang sudah dipersiapkan.

InsertApp.java

```
import java.io.*;
import java.sql.*;
import java.text.*;

public class InsertApp
{
    public static void main(String[] args)
    {
        Connection conn = null;
        try
        {
            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

            String url
                = "jdbc:mysql://localhost:3306/MYAPP_DB";
            String user = "ekobs";
            String pwd = "j2ee";
            conn = DriverManager.getConnection(
                url, user, pwd);

            InsertApp app = new InsertApp(conn);
            app.service();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
        finally
        {
            try
            {
                if(conn != null)
                {
                    conn.close();
                }
            }
            catch(SQLException sqle)
            {
                sqle.printStackTrace();
            }
        }
    }

    public InsertApp(Connection conn)
```

```

    {
        this.conn = conn;
        this.in
            = new BufferedReader(
                new InputStreamReader(System.in));
    }

    private Connection conn;
    private SimpleDateFormat format
        = new SimpleDateFormat("ddMMyyyy");
    private BufferedReader in;

    private void service()
        throws IOException, ParseException, SQLException
    {
        String id = readLine("Id");
        String name = readLine("Name");
        String department = readLine("Department");
        String jobTitle = readLine("Job Title");
        java.util.Date hireDate
            = format.parse(readLine("Hire Date (ddMMyyyy)"));
        boolean permanentEmployee
            = Boolean.valueOf(
                readLine("Permanent Employee (True/False)")
                    .booleanValue());
        double salary = Double.parseDouble(readLine("Salary"));

        String sqlInsert
            = "INSERT INTO EMPLOYEE_TBL VALUES (?, ?, ?, ?, ?, ?, ?)";
        PreparedStatement stmt
            = conn.prepareStatement(sqlInsert);

        stmt.setString(1, id);
        stmt.setString(2, name);
        stmt.setString(3, department);
        stmt.setString(4, jobTitle);
        stmt.setDate(5, new java.sql.Date(hireDate.getTime()));
        stmt.setBoolean(6, permanentEmployee);
        stmt.setDouble(7, salary);

        int count = stmt.executeUpdate();
        System.out.println("Query OK, " + count + " row affected");
        stmt.close();
    }

    private String readLine(String parameterName)
        throws IOException
    {
        System.out.print(parameterName + " : ");
        return in.readLine();
    }
}

```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac InsertApp.java
```

akan dihasilkan InsertApp.java

Langkah ke-4

Set CLASSPATH agar memuat JDBC driver yang dibutuhkan untuk mengakses MySQL database server ...

```
$ export CLASSPATH=.
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-
bin.jar
```

Langkah ke-5

Launch ...

```
$ java InsertApp
Id : 024
Name : Yudo Ekanata
Department : Tool Development
Job Title : Software Engineering
Hire Date (ddMMyyyy) : 17082003
Permanent Employee (True/False) : False
Salary : 3000000
Query OK, 1 row affected
```

CS-051-177

Merupakan latihan untuk meng-select data dari sebuah table di sebuah database ...

Persiapan

Anda perlu melakukan praktikum CS-051-171 terlebih dahulu. Directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis SelectApp.java, simpan di directory yang sudah dipersiapkan.

SelectApp.java

```
import java.io.*;
import java.sql.*;
import java.text.*;

public class SelectApp
{
    public static void main(String[] args)
    {
        Connection conn = null;
        try
        {
            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

            String url = "jdbc:mysql://localhost:3306/MYAPP_DB";
            String user = "ekobs";
            String pwd = "j2ee";
            conn = DriverManager.getConnection(
                url, user, pwd);

            SelectApp app = new SelectApp(conn);
            app.service(args[0]);
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
        finally
        {
            try
            {
                if(conn != null)
                {
                    conn.close();
                }
            }
            catch(SQLException sqle)
            {
                sqle.printStackTrace();
            }
        }
    }

    public SelectApp(Connection conn)
```



```

    {
        this.conn = conn;
        this.in = new BufferedReader(new
InputStreamReader(System.in));
    }

    private Connection conn;
    private SimpleDateFormat format = new SimpleDateFormat("ddMMyyyy");
    private BufferedReader in;

    private void service(String id)
        throws IOException, ParseException, SQLException
    {
        String sqlSelect =
            "SELECT * FROM EMPLOYEE_TBL WHERE ID = ?";

        PreparedStatement stmt = conn.prepareStatement(sqlSelect);

        stmt.setString(1, id);

        ResultSet rs = stmt.executeQuery();

        SimpleDateFormat format = new SimpleDateFormat("dd/MMMM/yyyy");
        while(rs.next())
        {
            id = rs.getString("ID");
            String name = rs.getString("NAME");
            String department = rs.getString("DEPARTMENT");
            Date hireDate = rs.getDate("HIRE_DATE");
            boolean permanentEmployee
                = rs.getBoolean("PERMANENT_EMPLOYEE");
            double salary = rs.getDouble("SALARY");

            System.out.println();
            System.out.println("E M P L O Y E E");
            System.out.println("Id : " + id);
            System.out.println("Name : " + name);
            System.out.println("Department : " + department);
            System.out.println
                ("Hire Date : " + format.format(hireDate));
            System.out.println
                ("Permanent Emp. : " + permanentEmployee);
            System.out.println("Salary : " + salary);
        }
        rs.close();
        stmt.close();
    }

    private String readLine(String parameterName)
        throws IOException
    {
        System.out.print(parameterName + " : ");
        return in.readLine();
    }
}

```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac SelectApp.java
```

akan dihasilkan SelectApp.java

Langkah ke-4

Set CLASSPATH agar memuat JDBC driver yang dibutuhkan untuk mengakses MySQL database server ...

```
$ export CLASSPATH=.
```

```
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-bin.jar
```

Langkah ke-5

Launch ...

```
$ java SelectApp 024
```

```
E M P L O Y E E
```

```
Id : 024
```

```
Name : Yudo Ekanata
```

```
Department : Tool Development
```

```
Hire Date : 17/August/2003
```

```
Permanent Emp. : false
```

```
Salary : 3000000.0
```

CS-051-178

Merupakan latihan untuk meng-update data dari sebuah table di sebuah database ...

Persiapan

Anda perlu melakukan praktikum CS-051-171 terlebih dahulu. Directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis UpdateApp.java, simpan di directory yang sudah dipersiapkan.

UpdateApp.java

```
import java.io.*;
import java.sql.*;
import java.text.*;

public class UpdateApp
{
    public static void main(String[] args)
    {
        Connection conn = null;
        try
        {
            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

            String url = "jdbc:mysql://localhost:3306/MYAPP_DB";
            String user = "ekobs";
            String pwd = "j2ee";
            conn = DriverManager.getConnection(
                url, user, pwd);

            UpdateApp app = new UpdateApp(conn);
            app.service();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
        finally
        {
            try
            {
                if(conn != null)
                {
                    conn.close();
                }
            }
            catch(SQLException sqle)
            {
                sqle.printStackTrace();
            }
        }
    }

    public UpdateApp(Connection conn)
    {
```

```

        this.conn = conn;
        this.in =
            new BufferedReader(new InputStreamReader(System.in));
    }

    private Connection conn;
    private SimpleDateFormat format = new SimpleDateFormat("ddMMyyyy");
    private BufferedReader in;

    private void service()
        throws IOException, ParseException, SQLException
    {
        String id = readLine("Id");
        String name = readLine("Name");
        String department = readLine("Department");
        String jobTitle = readLine("Job Title");
        java.util.Date hireDate
            = format.parse(readLine("Hire Date (ddMMyyyy)"));
        boolean permanentEmployee
            = Boolean.valueOf(
                readLine("Permanent Employee (True/False)"))
                .booleanValue();
        double salary = Double.parseDouble(readLine("Salary"));

        String sqlUpdate =
            "UPDATE EMPLOYEE_TBL "
            + "SET "
            + "NAME = ?, "
            + "DEPARTMENT = ?, "
            + "JOB_TITLE = ?, "
            + "HIRE_DATE = ?, "
            + "PERMANENT_EMPLOYEE = ?, "
            + "SALARY = ? "
            + "WHERE "
            + "ID = ?";
        PreparedStatement stmt
            = conn.prepareStatement(sqlUpdate);

        stmt.setString(7, id);
        stmt.setString(1, name);
        stmt.setString(2, department);
        stmt.setString(3, jobTitle);
        stmt.setDate(4, new java.sql.Date(hireDate.getTime()));
        stmt.setBoolean(5, permanentEmployee);
        stmt.setDouble(6, salary);

        int count = stmt.executeUpdate();
        System.out.println
            ("Query OK, " + count + " row affected");
        stmt.close();
    }

    private String readLine(String parameterName)
        throws IOException
    {
        System.out.print(parameterName + " : ");
        return in.readLine();
    }
}

```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac UpdateApp.java
```

akan dihasilkan UpdateApp.java

Langkah ke-4

Set CLASSPATH agar memuat JDBC driver yang dibutuhkan untuk mengakses MySQL database server ...

```
$ export CLASSPATH=.
```

```
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-bin.jar
```

Langkah ke-5

Launch ...

```
$ java UpdateApp
```

```
Id : 024
```

```
Name : Yudo Ekanata
```

```
Department : Tool Development
```

```
Job Title : Tester
```

```
Hire Date (ddMMyyyy) : 17082003
```

```
Permanent Employee (True/False) : True
```

```
Salary : 5000000
```

```
Query OK, 1 row affected
```

CS-051-179

Merupakan latihan untuk men-delete data dari sebuah table di sebuah database ...

Persiapan

Anda perlu melakukan praktikum CS-051-171 terlebih dahulu. Directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis DeleteApp.java, simpan di directory yang sudah dipersiapkan.

DeleteApp.java

```
import java.io.*;
import java.sql.*;
import java.text.*;

public class DeleteApp
{
    public static void main(String[] args)
    {
        Connection conn = null;
        try
        {
            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

            String url = "jdbc:mysql://localhost:3306/MYAPP_DB";
            String user = "ekobs";
            String pwd = "j2ee";
            conn = DriverManager.getConnection(
                url, user, pwd);

            DeleteApp app = new DeleteApp(conn);
            app.service();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
        finally
        {
            try
            {
                if(conn != null)
                {
                    conn.close();
                }
            }
            catch(SQLException sqle)
            {
                sqle.printStackTrace();
            }
        }
    }

    public DeleteApp(Connection conn)
    {
```

```

        this.conn = conn;
        this.in
            = new BufferedReader(
                new InputStreamReader(System.in));
    }

    private Connection conn;
    private SimpleDateFormat format
        = new SimpleDateFormat("ddMMyyyy");
    private BufferedReader in;

    private void service()
        throws IOException, ParseException, SQLException
    {
        String id = readLine("Id");
        String sqlDelete =
            "DELETE FROM EMPLOYEE_TBL WHERE ID = ?";
        PreparedStatement stmt = conn.prepareStatement(sqlDelete);

        stmt.setString(1, id);

        int count = stmt.executeUpdate();
        System.out.println("Query OK, " + count + " row affected");
        stmt.close();
    }

    private String readLine(String parameterName)
        throws IOException
    {
        System.out.print(parameterName + " : ");
        return in.readLine();
    }
}

```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac DeleteApp.java
```

akan dihasilkan DeleteApp.java

Langkah ke-4

Set CLASSPATH agar memuat JDBC driver yang dibutuhkan untuk mengakses MySQL database server ...

```
$ export CLASSPATH=.
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-
bin.jar
```

Langkah ke-5

Launch ...

```
$ java DeleteApp
Id : 024
Query OK, 1 row affected
```


18Data Access Object

...

19Desktop

Untuk mengembangkan aplikasi Desktop dengan Java, Anda bisa menggunakan Swing dan AWT yang merupakan library yang menyediakan komponen-komponen GUI.

Komponen utama dari Java untuk Desktop adalah JFrame. Dengan JFrame, Anda bisa membuat sebuah window, di mana di atasnya bisa diletakkan komponen-komponen GUI.

Di antara komponen yang lumrah digunakan adalah JLabel, JButton, JText, JTextArea, JComboBox, JRadioButton, JCheckBox, JMenuBar dan JMenu.

Dalam penanganan event, Java menerapkan event-delegation model. Dalam model ini, satu atau lebih obyek dapat didaftarkan sebagai listener dari sebuah event. Saat event tersebut terjadi, maka method dari obyek tersebut yang bersesuaian dengan event yang terjadi, akan dijalankan.

CS-051-191

Merupakan pengenalan dengan Swing ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis SalamKeadilanJFrame.java, simpan di directory yang sudah dipersiapkan.

SalamKeadilanJFrame.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class SalamKeadilanJFrame
    extends JFrame
    implements ActionListener
{
    JLabel labelPesan;

    public SalamKeadilanJFrame()
    {
        JButton tombolPesan = new JButton("Click me !");
        tombolPesan.addActionListener(this);
        getContentPane().add(tombolPesan, BorderLayout.NORTH);

        labelPesan = new JLabel();
        getContentPane().add(labelPesan, BorderLayout.CENTER);

        WindowListener wlSalamKeadilanFrame =
            new WindowAdapter()
            {
                public void windowClosing(WindowEvent we)
                {
                    System.exit(0);
                }
            };
        addWindowListener(wlSalamKeadilanFrame);
    }

    public void actionPerformed(ActionEvent ae)
    {
        labelPesan.setText("Salam Keadilan !");
    }
}
```

Langkah ke-2

Tulis SalamKeadilanSwingApp.java, simpan di directory yang sudah dipersiapkan.

SalamKeadilanSwingApp.java

```
import javax.swing.*;

public class SalamKeadilanSwingApp
{
```

```

public static void main(String[] args)
{
    JFrame frame = new SalamKeadilanJFrame();

    String lookAndFeel
        = "javax.swing.plaf.metal.MetalLookAndFeel";
    if (args.length > 0)
    {
        if (args[0].equals("metal"))
        {
            //
        }
        else
        if (args[0].equals("window"))
        {
            lookAndFeel
                = "com.sun.java.swing.plaf.windows.WindowsLookAndFeel";
        }
        else
        if (args[0].equals("motif"))
        {
            lookAndFeel
                = "com.sun.java.swing.plaf.motif.MotifLookAndFeel";
        }
    }

    try
    {
        UIManager.setLookAndFeel(lookAndFeel);
        SwingUtilities.updateComponentTreeUI(frame);
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }

    frame.setLocation(300, 300);
    frame.setSize(300, 200);
    frame.show();
}
}

```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

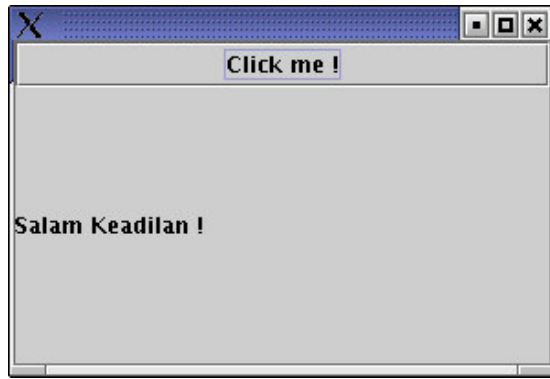
```
$ javac SalamKeadilanSwingApp.java
```

akan dihasilkan SalamKeadilanSwingApp.java

Langkah ke-5

Launch ...

```
$ java SalamKeadilanSwingApp
```



CS-051-192

Contoh menggunakan JTextField ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis MyJFrame.java, simpan di directory yang sudah dipersiapkan.

MyJFrame.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJFrame
    extends JFrame
    implements ActionListener
{
    JTextField inputTxt;
    MyJDialog myDlg;

    public MyJFrame()
    {
        JButton commandBtn = new JButton("Click me !");
        commandBtn.addActionListener(this);
        inputTxt = new JTextField();

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(2,2));

        panel.add(new JLabel("Nama : "));
        panel.add(inputTxt);
        panel.add(new JLabel(""));
        panel.add(commandBtn);

        getContentPane().add(panel, BorderLayout.CENTER);

        myDlg = new MyJDialog(this);
        myDlg.setLocation(400, 300);

        WindowListener wl =
            new WindowAdapter()
            {
                public void windowClosing(WindowEvent we)
                {
                    System.exit(0);
                }
            };
        addWindowListener(wl);
    }

    public void actionPerformed(ActionEvent ae)
    {
        String value = inputTxt.getText();

        myDlg.setMessage(value);
        myDlg.pack();
    }
}
```

```
        myDlg.show();
    }
}
```

Langkah ke-2

Tulis MyJDialog.java, simpan di directory yang sudah dipersiapkan.

MyJDialog.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJDialog
    extends JDialog
{
    private JLabel messageLbl;

    public MyJDialog(JFrame parent)
    {
        super(parent, null, true);

        messageLbl = new JLabel();
        getContentPane().add(messageLbl, BorderLayout.NORTH);

        JButton closeBtn = new JButton("Close");
        closeBtn.addActionListener
            (
                new ActionListener()
                {
                    public void actionPerformed(ActionEvent e)
                    {
                        dispose();
                    }
                }
            );
        getContentPane().add(closeBtn, BorderLayout.SOUTH);
    }

    public void setMessage(String message)
    {
        messageLbl.setText(message);
    }
}
```

Langkah ke-3

Tulis SwingApp.java, simpan di directory yang sudah dipersiapkan.

SwingApp.java

```
import javax.swing.*;

public class SwingApp
{
    public static void main(String[] args)
    {
        JFrame frame = new MyJFrame();
        frame.setLocation(300, 300);
    }
}
```

```
        frame.pack();  
        frame.show();  
    }  
}
```

Langkah ke-4

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-5

Compile ...

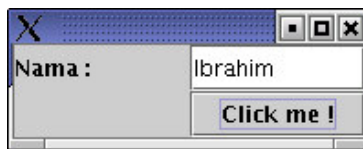
```
$ javac SwingApp.java
```

akan dihasilkan MyJFrame.class, MyJDialog.class dan SwingApp.java

Langkah ke-6

Launch ...

```
$ java SwingApp
```



CS-051-193

Contoh menggunakan JCheckBox ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis MyJFrame.java, simpan di directory yang sudah dipersiapkan.

MyJFrame.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJFrame
    extends JFrame
    implements ActionListener
{
    MyJDialog myDlg;

    JCheckBox bananaCheckBox;
    JCheckBox mangoCheckBox;
    JCheckBox orangeCheckBox;

    public MyJFrame()
    {
        JButton commandBtn = new JButton("Click me !");
        commandBtn.addActionListener(this);

        bananaCheckBox = new JCheckBox("Banana");
        mangoCheckBox = new JCheckBox("Mango");
        orangeCheckBox = new JCheckBox("Orange");

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4,1));
        panel.add(bananaCheckBox);
        panel.add(mangoCheckBox);
        panel.add(orangeCheckBox);
        panel.add(commandBtn);

        getContentPane().add(panel, BorderLayout.CENTER);

        myDlg = new MyJDialog(this);
        myDlg.setLocation(400, 300);

        WindowListener wl =
            new WindowAdapter()
            {
                public void windowClosing(WindowEvent we)
                {
                    System.exit(0);
                }
            };
        addWindowListener(wl);
    }

    public void actionPerformed(ActionEvent ae)
```

```

{
    String selected = null;
    if(bananaCheckBox.isSelected())
    {
        selected = bananaCheckBox.getActionCommand();
    }
    if(mangoCheckBox.isSelected())
    {
        if(selected==null)
            selected = "";
        else
            selected += " , ";

        selected += mangoCheckBox.getActionCommand();
    }
    if(orangeCheckBox.isSelected())
    {
        if(selected==null)
            selected = "";
        else
            selected += " , ";
        selected += orangeCheckBox.getActionCommand();
    }

    myDlg.setMessage(selected);
    myDlg.pack();
    myDlg.show();
}
}

```

Langkah ke-2

Tulis MyJDialog.java, simpan di directory yang sudah dipersiapkan.

MyJDialog.java

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJDialog
    extends JDialog
{
    private JLabel messageLbl;

    public MyJDialog(JFrame parent)
    {
        super(parent, null, true);

        messageLbl = new JLabel();
        getContentPane().add(messageLbl, BorderLayout.NORTH);

        JButton closeBtn = new JButton("Close");
        closeBtn.addActionListener
            (
                new ActionListener()
                {
                    public void actionPerformed(ActionEvent e)
                    {
                        dispose();
                    }
                }
            )
    }
}

```

```

        }
    );
    getContentPane().add(closeBtn, BorderLayout.SOUTH);
}

public void setMessage(String message)
{
    messageLbl.setText(message);
}
}

```

Langkah ke-3

Tulis SwingApp.java, simpan di directory yang sudah dipersiapkan.

SwingApp.java

```

import javax.swing.*;

public class SwingApp
{
    public static void main(String[] args)
    {
        JFrame frame = new MyJFrame();
        frame.setLocation(300, 300);
        frame.pack();
        frame.show();
    }
}

```

Langkah ke-4

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-5

Compile ...

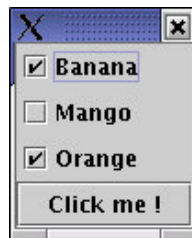
```
$ javac SwingApp.java
```

akan dihasilkan MyJFrame.class, MyJDialog.class dan SwingApp.java

Langkah ke-6

Launch ...

```
$ java SwingApp
```





CS-051-194

Contoh menggunakan JRadioButton ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis MyJFrame.java, simpan di directory yang sudah dipersiapkan.

MyJFrame.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJFrame
    extends JFrame
    implements ActionListener
{
    MyJDialog myDlg;

    JRadioButton bananaRadioButton;
    JRadioButton mangoRadioButton;
    JRadioButton orangeRadioButton;

    public MyJFrame()
    {
        JButton commandBtn = new JButton("Click me !");
        commandBtn.addActionListener(this);

        bananaRadioButton = new JRadioButton("Banana");
        mangoRadioButton = new JRadioButton("Mango");
        orangeRadioButton = new JRadioButton("Orange");

        ButtonGroup group = new ButtonGroup();
        group.add(bananaRadioButton);
        group.add(mangoRadioButton);
        group.add(orangeRadioButton);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4,1));
        panel.add(bananaRadioButton);
        panel.add(mangoRadioButton);
        panel.add(orangeRadioButton);
        panel.add(commandBtn);

        getContentPane().add(panel, BorderLayout.CENTER);

        myDlg = new MyJDialog(this);
        myDlg.setLocation(400, 300);

        WindowListener wl =
            new WindowAdapter()
            {
                public void windowClosing(WindowEvent we)
                {
                    System.exit(0);
                }
            }
    }
}
```

```

        } ;
        addWindowListener(wl);
    }

    public void actionPerformed(ActionEvent ae)
    {
        String selected = null;
        if(bananaRadioButton.isSelected())
        {
            selected = bananaRadioButton.getActionCommand();
        }
        if(mangoRadioButton.isSelected())
        {
            if(selected==null)
                selected = "";
            else
                selected += " , ";

            selected += mangoRadioButton.getActionCommand();
        }
        if(orangeRadioButton.isSelected())
        {
            if(selected==null)
                selected = "";
            else
                selected += " , ";
            selected += orangeRadioButton.getActionCommand();
        }

        myDlg.setMessage(selected);
        myDlg.pack();
        myDlg.show();
    }
}

```

Langkah ke-2

Tulis MyJDialog.java, simpan di directory yang sudah dipersiapkan.

MyJDialog.java

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJDialog
    extends JDialog
{
    private JLabel messageLbl;

    public MyJDialog(JFrame parent)
    {
        super(parent, null, true);

        messageLbl = new JLabel();
        getContentPane().add(messageLbl, BorderLayout.NORTH);

        JButton closeBtn = new JButton("Close");
        closeBtn.addActionListener
            (
                new ActionListener()
                {

```

```

        public void actionPerformed(ActionEvent e)
        {
            dispose();
        }
    }
);
getContentPane().add(closeBtn, BorderLayout.SOUTH);
}

public void setMessage(String message)
{
    messageLbl.setText(message);
}
}

```

Langkah ke-3

Tulis SwingApp.java, simpan di directory yang sudah dipersiapkan.

SwingApp.java

```

import javax.swing.*;

public class SwingApp
{
    public static void main(String[] args)
    {
        JFrame frame = new MyJFrame();
        frame.setLocation(300, 300);
        frame.pack();
        frame.show();
    }
}

```

Langkah ke-4

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-5

Compile ...

```
$ javac SwingApp.java
```

akan dihasilkan MyJFrame.class, MyJDialog.class dan SwingApp.java

Langkah ke-6

Launch ...

```
$ java SwingApp
```





CS-051-195

Contoh menggunakan JComboBox ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis MyJFrame.java, simpan di directory yang sudah dipersiapkan.

MyJFrame.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJFrame
    extends JFrame
    implements ActionListener
{
    MyJDialog myDlg;

    JComboBox comboBox;

    public MyJFrame()
    {
        JButton commandBtn = new JButton("Click me !");
        commandBtn.addActionListener(this);

        comboBox = new JComboBox();
        comboBox.addItem("Banana");
        comboBox.addItem("Mango");
        comboBox.addItem("Orange");

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(2,1));
        panel.add(comboBox);
        panel.add(commandBtn);

        getContentPane().add(panel, BorderLayout.CENTER);

        myDlg = new MyJDialog(this);
        myDlg.setLocation(400, 300);

        WindowListener wl =
            new WindowAdapter()
            {
                public void windowClosing(WindowEvent we)
                {
                    System.exit(0);
                }
            };
        addWindowListener(wl);
    }

    public void actionPerformed(ActionEvent ae)
    {
        String selected = (String)comboBox.getSelectedItem();
    }
}
```

```

        myDlg.setMessage(selected);
        myDlg.pack();
        myDlg.show();
    }
}

```

Langkah ke-2

Tulis MyJDialog.java, simpan di directory yang sudah dipersiapkan.

MyJDialog.java

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJDialog
    extends JDialog
{
    private JLabel messageLbl;

    public MyJDialog(JFrame parent)
    {
        super(parent, null, true);

        messageLbl = new JLabel();
        getContentPane().add(messageLbl, BorderLayout.NORTH);

        JButton closeBtn = new JButton("Close");
        closeBtn.addActionListener
            (
                new ActionListener()
                {
                    public void actionPerformed(ActionEvent e)
                    {
                        dispose();
                    }
                }
            );
        getContentPane().add(closeBtn, BorderLayout.SOUTH);
    }

    public void setMessage(String message)
    {
        messageLbl.setText(message);
    }
}

```

Langkah ke-3

Tulis SwingApp.java, simpan di directory yang sudah dipersiapkan.

SwingApp.java

```

import javax.swing.*;

public class SwingApp
{
    public static void main(String[] args)
    {

```

```
        JFrame frame = new MyJFrame();
        frame.setLocation(300, 300);
        frame.pack();
        frame.show();
    }
}
```

Langkah ke-4

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-5

Compile ...

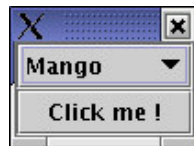
```
$ javac SwingApp.java
```

akan dihasilkan MyJFrame.class, MyJDialog.class dan SwingApp.java

Langkah ke-6

Launch ...

```
$ java SwingApp
```



CS-051-196

Contoh menggunakan JButton ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis MyJFrame.java, simpan di directory yang sudah dipersiapkan.

MyJFrame.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJFrame
    extends JFrame
    implements ActionListener
{
    MyJDialog myDlg;

    JList list;

    public MyJFrame()
    {
        JButton commandBtn = new JButton("Click me !");
        commandBtn.addActionListener(this);

        DefaultListModel listModel = new DefaultListModel();
        listModel.addElement("Banana");
        listModel.addElement("Mango");
        listModel.addElement("Orange");

        list = new JList(listModel);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(2,1));
        panel.add(list);
        panel.add(commandBtn);

        getContentPane().add(panel, BorderLayout.CENTER);

        myDlg = new MyJDialog(this);
        myDlg.setLocation(400, 300);

        WindowListener wl =
            new WindowAdapter()
            {
                public void windowClosing(WindowEvent we)
                {
                    System.exit(0);
                }
            };
        addWindowListener(wl);
    }

    public void actionPerformed(ActionEvent ae)
    {

```

```

String selected = null;
Object[] values = list.getSelectedValues();
for(int i=0;i<values.length;i++)
{
    if(selected == null )
        selected = "";
    else
        selected += ",";
    selected += (String) values[i];
}

myDlg.setMessage(selected);
myDlg.pack();
myDlg.show();
}
}

```

Langkah ke-2

Tulis MyJDialog.java, simpan di directory yang sudah dipersiapkan.

MyJDialog.java

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJDialog
    extends JDialog
{
    private JLabel messageLbl;

    public MyJDialog(JFrame parent)
    {
        super(parent, null, true);

        messageLbl = new JLabel();
        getContentPane().add(messageLbl, BorderLayout.NORTH);

        JButton closeBtn = new JButton("Close");
        closeBtn.addActionListener
            (
                new ActionListener()
                {
                    public void actionPerformed(ActionEvent e)
                    {
                        dispose();
                    }
                }
            );
        getContentPane().add(closeBtn, BorderLayout.SOUTH);
    }

    public void setMessage(String message)
    {
        messageLbl.setText(message);
    }
}

```

Langkah ke-3

Tulis `SwingApp.java`, simpan di directory yang sudah dipersiapkan.

SwingApp.java

```
import javax.swing.*;

public class SwingApp
{
    public static void main(String[] args)
    {
        JFrame frame = new MyJFrame();
        frame.setLocation(300, 300);
        frame.pack();
        frame.show();
    }
}
```

Langkah ke-4

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-5

Compile ...

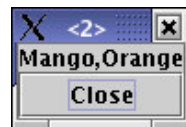
```
$ javac SwingApp.java
```

akan dihasilkan `MyJFrame.class`, `MyJDialog.class` dan `SwingApp.java`

Langkah ke-6

Launch ...

```
$ java SwingApp
```



CS-051-197

Contoh menggunakan JMenu ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis MyJFrame.java, simpan di directory yang sudah dipersiapkan.

MyJFrame.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJFrame
    extends JFrame
    implements ActionListener
{
    MyJDialog myDlg;

    public MyJFrame()
    {
        JMenu menu0 = new JMenu("Asia");
        JMenu menu1 = new JMenu("Africa");
        JMenu menu2 = new JMenu("Europe");

        JMenuItem menuItem01 = new JMenuItem("Beijing");
        menuItem01.addActionListener(this);
        menu0.add(menuItem01);
        JMenuItem menuItem02 = new JMenuItem("Jakarta");
        menuItem02.addActionListener(this);
        menu0.add(menuItem02);
        JMenuItem menuItem03 = new JMenuItem("Tokyo");
        menuItem03.addActionListener(this);
        menu0.add(menuItem03);

        JMenuItem menuItem11 = new JMenuItem("Cairo");
        menuItem11.addActionListener(this);
        menu1.add(menuItem11);
        JMenuItem menuItem12 = new JMenuItem("Chartoum");
        menuItem12.addActionListener(this);
        menu1.add(menuItem12);

        JMenuItem menuItem21 = new JMenuItem("Amsterdam");
        menuItem21.addActionListener(this);
        menu2.add(menuItem21);
        JMenuItem menuItem22 = new JMenuItem("Frankfurt");
        menuItem22.addActionListener(this);
        menu2.add(menuItem22);
        JMenuItem menuItem23 = new JMenuItem("Helsinki");
        menuItem23.addActionListener(this);
        menu2.add(menuItem23);
        JMenuItem menuItem24 = new JMenuItem("London");
        menuItem24.addActionListener(this);
        menu2.add(menuItem24);
    }
}
```



```

        JMenuBar bar = new JMenuBar();
        bar.add(menu0);
        bar.add(menu1);
        bar.add(menu2);

        setJMenuBar(bar);

        myDlg = new MyJDialog(this);
        myDlg.setLocation(400, 300);

        WindowListener wl =
            new WindowAdapter()
            {
                public void windowClosing(WindowEvent we)
                {
                    System.exit(0);
                }
            };
        addWindowListener(wl);
    }

    public void actionPerformed(ActionEvent ae)
    {
        JMenuItem item = (JMenuItem) ae.getSource();
        String command = item.getText();
        myDlg.setMessage(command);
        myDlg.pack();
        myDlg.show();
    }
}

```

Langkah ke-2

Tulis MyJDialog.java, simpan di directory yang sudah dipersiapkan.

MyJDialog.java

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJDialog
    extends JDialog
{
    private JLabel messageLbl;

    public MyJDialog(JFrame parent)
    {
        super(parent, null, true);

        messageLbl = new JLabel();
        getContentPane().add(messageLbl, BorderLayout.NORTH);

        JButton closeBtn = new JButton("Close");
        closeBtn.addActionListener
            (
                new ActionListener()
                {
                    public void actionPerformed(ActionEvent e)
                    {
                        dispose();
                    }
                }
            );
    }
}

```

```

        }
    );
    getContentPane().add(closeBtn, BorderLayout.SOUTH);
}

public void setMessage(String message)
{
    messageLbl.setText(message);
}
}

```

Langkah ke-3

Tulis SwingApp.java, simpan di directory yang sudah dipersiapkan.

SwingApp.java

```

import javax.swing.*;

public class SwingApp
{
    public static void main(String[] args)
    {
        JFrame frame = new MyJFrame();
        frame.setLocation(300, 300);
        frame.pack();
        frame.show();
    }
}

```

Langkah ke-4

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-5

Compile ...

```
$ javac SwingApp.java
```

akan dihasilkan MyJFrame.class, MyJDialog.class dan SwingApp.java

Langkah ke-6

Launch ...

```
$ java SwingApp
```



CS-051-198

Contoh menggunakan JTable ...

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis MyJFrame.java, simpan di directory yang sudah dipersiapkan.

MyJFrame.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class MyJFrame
    extends JFrame
{
    public MyJFrame()
    {
        String[] columnNames =
            {"Id", "Name", "Department", "Job Title"};
        String[][] rowData =
            {
                {"024", "Yudo Ekanata",
                 "Software Engineering", "Programmer"},
                {"032", "Muhammad Arifianto",
                 "Business Development", "Account Manager"},
                {"033", "Wawan Setiawan",
                 "Technical Support", "System Administrator"},
            };

        JTable table = new JTable(rowData, columnNames);
        JScrollPane pane = new JScrollPane(table);
        getContentPane().add(pane, BorderLayout.CENTER);

        WindowListener wl =
            new WindowAdapter()
            {
                public void windowClosing(WindowEvent we)
                {
                    System.exit(0);
                }
            };
        addWindowListener(wl);
    }
}
```

Langkah ke-2

Tulis SwingApp.java, simpan di directory yang sudah dipersiapkan.

SwingApp.java

```
import javax.swing.*;

public class SwingApp
```

```
{
    public static void main(String[] args)
    {
        JFrame frame = new MyJFrame();
        frame.setLocation(300, 300);
        frame.pack();
        frame.show();
    }
}
```

Langkah ke-3

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-4

Compile ...

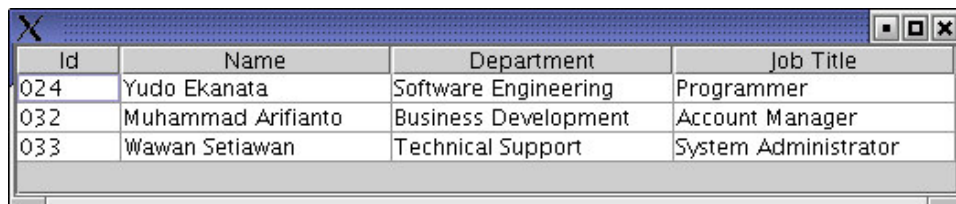
```
$ javac SwingApp.java
```

akan dihasilkan MyJFrame.class, MyJDialog.class dan SwingApp.java

Langkah ke-5

Launch ...

```
$ java SwingApp
```



The screenshot shows a Java Swing window with a table containing three rows of employee data. The table has four columns: Id, Name, Department, and Job Title. The data is as follows:

| Id | Name | Department | Job Title |
|-----|--------------------|----------------------|----------------------|
| 024 | Yudo Ekanata | Software Engineering | Programmer |
| 032 | Muhammad Arifianto | Business Development | Account Manager |
| 033 | Wawan Setiawan | Technical Support | System Administrator |

20 Aplikasi Database di Desktop

Latihan yang memadukan JDBC dan Swing ...

21Security

Java mempunyai apa yang disebut sebagai sandbox security model. Dalam model ini, aplikasi yang dikembangkan dengan Java, akan bekerja dalam Java Virtual Machine dengan keterbatasan-keterbatasan tertentu berdasarkan security policy. Melalui security policy ini, Anda bisa menentukan direktori mana yang boleh diakses oleh aplikasi tersebut, atau apakah aplikasi tsb boleh mengadakan network connection.

CS-051-211

Merupakan latihan menggunakan Java Security Manager. Dengan menggunakan Java Security Manager maka aplikasi Java yang dijalankan akan dibatasi di dalam Sandbox.

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis FilePrintlnApp.java, simpan di directory yang sudah dipersiapkan.

FilePrintlnApp.java

```
import java.io.*;

public class FilePrintlnApp
{
    public static void main(String[] args)
        throws Exception
    {
        String fileName = args[0];

        RandomAccessFile raf
            = new RandomAccessFile(fileName, "r");
        String line;
        while((line = raf.readLine()) != null)
        {
            System.out.println(line);
        }
        raf.close();
    }
}
```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac FilePrintlnApp.java
```

akan dihasilkan FilePrintlnApp.class

Langkah ke-4

Launch ... tanpa Security Manager. Misal untuk men-cetak isi file /home/lab/.bashrc

```
$ java FilePrintlnApp /home/lab/.bashrc
# .bashrc

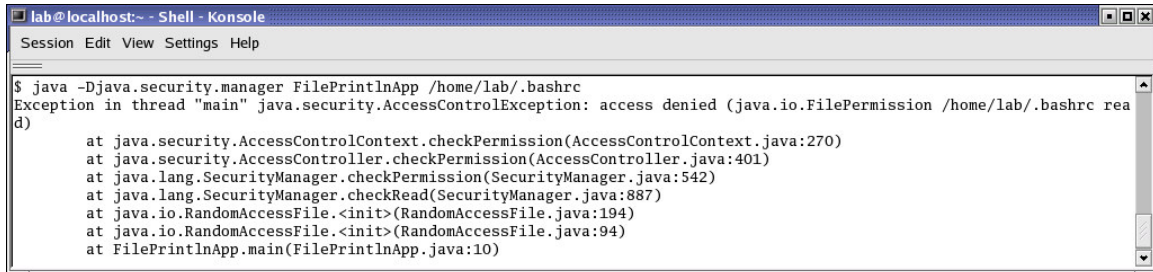
# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
```


Langkah ke-4

Launch ... dengan Security Manager. Misal untuk men-cetak isi file /home/lab/.bashrc

```
$ java -Djava.security.manager FilePrintlnApp /home/lab/.bashrc
Exception in thread "main" java.security.AccessControlException: access
denied (java.io.FilePermission /home/lab/.bashrc read)
```



```
lab@localhost:~ - Shell - Konsole
Session Edit View Settings Help

$ java -Djava.security.manager FilePrintlnApp /home/lab/.bashrc
Exception in thread "main" java.security.AccessControlException: access denied (java.io.FilePermission /home/lab/.bashrc read)
    at java.security.AccessControlContext.checkPermission(AccessControlContext.java:270)
    at java.security.AccessController.checkPermission(AccessController.java:401)
    at java.lang.SecurityManager.checkPermission(SecurityManager.java:542)
    at java.lang.SecurityManager.checkRead(SecurityManager.java:887)
    at java.io.RandomAccessFile.<init>(RandomAccessFile.java:194)
    at java.io.RandomAccessFile.<init>(RandomAccessFile.java:94)
    at FilePrintlnApp.main(FilePrintlnApp.java:10)
```

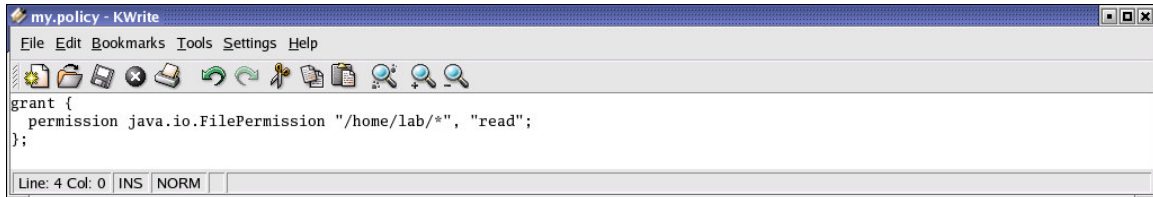
Dapat ditemukan bahwa aplikasi FilePrintlnApp mendapatkan java.security.AccessControlException

Langkah ke-6

Tulis my.policy dan simpan di directory kerja ...

my.policy

```
grant {
    permission java.io.FilePermission "/home/lab/*", "read";
};
```



```
my.policy - KWrite
File Edit Bookmarks Tools Settings Help

grant {
    permission java.io.FilePermission "/home/lab/*", "read";
};

Line: 4 Col: 0 | INS | NORM
```

Langkah ke-7

Launch ... dengan Security policy my.policy ...

```
$ java -Djava.security.manager -Djava.security.policy=my.policy
FilePrintlnApp /home/lab/.bashrc
# .bashrc

# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
```

CS-051-212

Merupakan latihan menggunakan Java Security Manager. Dengan menggunakan Java Security Manager maka aplikasi Java yang dijalankan akan dibatasi di dalam Sandbox.

Persiapan

Buat sebuah directory untuk bekerja, misalnya /home/lab/myjava.

Langkah

Langkah ke-1

Tulis ConnectionApp.java, simpan di directory yang sudah dipersiapkan.

ConnectionApp.java

```
import java.io.*;
import java.sql.*;

public class ConnectionApp
{
    public static void main(String[] args)
    {
        Connection conn = null;
        try
        {
            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

            String url = "jdbc:mysql://localhost:3306/MYAPP_DB";
            String user = "ekobs";
            String pwd = "j2ee";

            System.out.println
                ("Mencoba membangun connection ke '" + url
                 + "' dengan user '" + user
                 + "' dan password '" + pwd + "' ...");
            conn = DriverManager.getConnection(
                url, user, pwd);
            System.out.println("Success.");
        }
        catch(ClassNotFoundException cnfe)
        {
            cnfe.printStackTrace();
        }
        catch(SQLException sqle)
        {
            sqle.printStackTrace();
        }
        finally
        {
            try
            {
                if(conn != null)
                {
                    System.out.println("Menutup connection.");
                    conn.close();
                }
            }
            catch(SQLException sqle)
            {
            }
        }
    }
}
```

```

        {
            sqle.printStackTrace();
        }
    }
}

```

Langkah ke-2

Buka sebuah terminal, dan change directory ke directory kerja tersebut

Langkah ke-3

Compile ...

```
$ javac ConnectionApp.java
```

akan dihasilkan ConnectionApp.class

Langkah ke-4

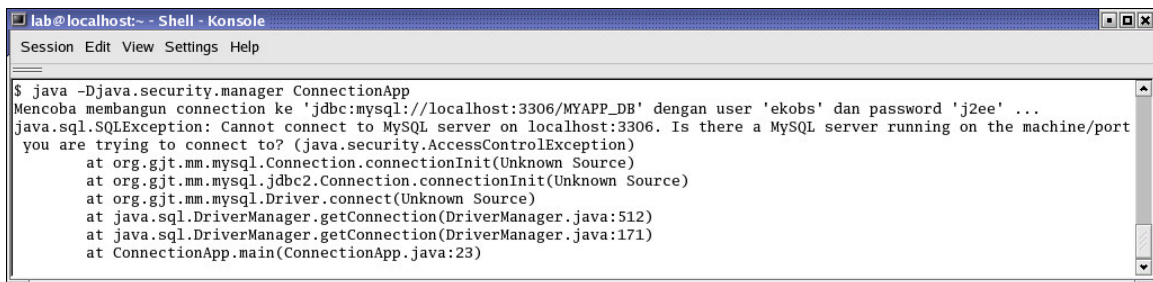
Launch ... tanpa Security Manager

```
$ export CLASSPATH=.
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-
bin.jar
$ java ConnectionApp
Mencoba membangun connection ke 'jdbc:mysql://localhost:3306/MYAPP_DB'
dengan user 'ekobs' dan password 'j2ee' ...
Success.
Menutup connection.
```

Langkah ke-5

Launch ... dengan Security Manager.

```
$ java -Djava.security.manager ConnectionApp
Mencoba membangun connection ke 'jdbc:mysql://localhost:3306/MYAPP_DB'
dengan user 'ekobs' dan password 'j2ee' ...
java.sql.SQLException: Cannot connect to MySQL server on localhost:3306.
Is there a MySQL server running on the machine/port you are trying to
connect to? (java.security.AccessControlException)
```



```

lab@localhost:~ - Shell - Konsole
Session Edit View Settings Help
$ java -Djava.security.manager ConnectionApp
Mencoba membangun connection ke 'jdbc:mysql://localhost:3306/MYAPP_DB' dengan user 'ekobs' dan password 'j2ee' ...
java.sql.SQLException: Cannot connect to MySQL server on localhost:3306. Is there a MySQL server running on the machine/port
you are trying to connect to? (java.security.AccessControlException)
    at org.gjt.mm.mysql.Connection.connectionInit(Unknown Source)
    at org.gjt.mm.mysql.jdbc2.Connection.connectionInit(Unknown Source)
    at org.gjt.mm.mysql.Driver.connect(Unknown Source)
    at java.sql.DriverManager.getConnection(DriverManager.java:512)
    at java.sql.DriverManager.getConnection(DriverManager.java:171)
    at ConnectionApp.main(ConnectionApp.java:23)

```

Dapat ditemukan bahwa aplikasi ConnectionApp mendapatkan java.security.AccessControlException

Langkah ke-6

Tulis my.policy dan simpan di directory kerja ...

my.policy

```
grant {  
    permission java.net.SocketPermission "localhost:3306", "connect";  
};
```

Langkah ke-7

Launch ... dengan Security policy my.policy ...

```
$ java -Djava.security.manager -Djava.security.policy=my.policy  
ConnectionApp  
Mencoba membangun connection ke 'jdbc:mysql://localhost:3306/MYAPP_DB'  
dengan user 'ekobs' dan password 'j2ee' ...  
Success.  
Menutup connection.
```