

Dasar-Dasar Pemrograman Java

Eko Budhi Suprasetiawan

ekobs@developerforce.net

Lisensi Dokumen:

Copyright © 2003-2006 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

1. Meng-install Java Development Kit

Pengantar

Java Development Kit (JDK) merupakan perlengkapan tempur yang mendasar dalam pengembangan aplikasi dengan Java. Dua program utama yang disediakan dalam JDK adalah :

- javac, yaitu program untuk meng-compile kode sumber
- java, yaitu program untuk meluncurkan aplikasi

Proses

Mendapatkan installation file

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.

Installation dengan executable binary

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_3_1_01-linux-i386.bin`. Untuk menjalankannya, melalui console :

```
$ ./j2sdk-1_3_1_01-linux-i386.bin
```

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_3_1_01-linux-i386.bin` dari directory `/home/lab` maka sebuah directory `jdk1.3.1_01` akan dibuat dan di directory inilah file-file JDK berada.

```
Unpacking...
Checksumming...
0
0
Extracting...
UnZipSFX 5.40 of 28 November 1998, by Info-ZIP (Zip-Bugs@lists.wku.edu).
  creating: jdk1.3.1_01/
  creating: jdk1.3.1_01/jre/
  creating: jdk1.3.1_01/jre/bin/
  creating: jdk1.3.1_01/jre/bin/i386/
  creating: jdk1.3.1_01/jre/bin/i386/green_threads/
  inflating: jdk1.3.1_01/jre/bin/i386/green_threads/java
sehingga akhirnya :
  inflating: jdk1.3.1_01/man/ja_JP.eucJP/man1/native2ascii.1
  inflating: jdk1.3.1_01/man/ja_JP.eucJP/man1/rmic.1
  inflating: jdk1.3.1_01/man/ja_JP.eucJP/man1/serialver.1
Done.
```

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

2. Susunan Kode Sumber .java

Dalam mengembangkan program Java, Anda menulis class dan class. Class ini ditulis dalam kode sumber yang disimpan sebagai file teks biasa berekstension .java.

Dalam file .java, dapat dideklarasikan :

1. package
2. import
3. satu atau lebih class

Deklarasi Package

Deklarasi package digunakan untuk mengelompokkan class-class. Sebuah package dapat mempunyai satu atau lebih sub-package ini sehingga dapat menyusun sebuah hirarki.

Tata cara penulisan :

```
package <namapackage>.<namasubpackage>.<namasubsubpackage>;
```

Contoh deklarasi package adalah :

```
package java.awt;
```

atau

```
package org.apache.tomcat.core;
```

atau

```
package net.developerforce.relieve.dataaccess;
```

Deklarasi package bersifat opsional, tidak harus ada. Tanpa deklarasi package maka sebuah class dikelompokkan ke dalam default package.

Deklarasi Import

Deklarasi import, bersifat opsional, digunakan untuk menunjukkan package atau class yang digunakan dalam sebuah program Java.

Contohnya adalah :

```
import java.io.RandomAccessFile;
```

```
import java.net.*;
```

Deklarasi import tidak akan menjadikan class atau package yang Anda cantumkan digabungkan dengan program Anda saat kompilasi atau saat diluncurkan. Deklarasi import digunakan untuk menunjukkan java atau javac dalam menemukan class yang Anda gunakan.

Deklarasi Class

Deklarasi class merupakan kandungan utama sebuah file berekstension .java. Deklarasi class terutama memuat :

1. nama class, bisa dilengkapi dengan kendali akses, deklarasi extends maupun deklarasi implements. Pola yang lumrah adalah :
2. `public class ClassName`
3. `extends SuperClassName`

4. implements Interface1Name, Interface2Name, Interface3Name
5. constructor, yang dipanggil pada saat dibuat instans dari class.
6. deklarasi variabel-variabel
7. deklarasi prosedur-prosedur, yang di dalam Java disebut method.

Contoh

Berikut adalah sebuah contoh sebuah kode sumber untuk class DataAccessPatient. Class ini bertanggung jawab untuk menyimpan obyek Patient ke sistem basis data.

```
package net.developerforce.relieve.dataaccess;

import java.sql.*;
import net.developerforce.relieve.value.Patient;

public class DataAccessPatient
{
    private Connection conn;

    public DataAccessPatient(Connection conn)
    {
        this.conn = conn;
    }

    public void insertPatient(Patient patient)
        throws SQLException
    {
        String insertStr =
            "INSERT INTO PATIENTTBL "
            + " VALUES("
            + "'" + patient.getId() + "'"
            + ","
            + "'" + patient.getName() + "'"
            + ","
            + "'" + patient.getIc() + "'"
            + ","
            + "'" + patient.getAddress() + "'"
            + ","
            + "'" + patient.getTelephone() + "'"
            + ","
            + "'" + patient.getEmail() + "'"
            + ")";

        Statement stmt = conn.createStatement();
        stmt.executeUpdate(insertStr);
    }

    public void updatePatient(Patient patient)
        throws SQLException
    {
        String updateStr =
            "UPDATE PATIENTTBL "
            + "SET "
            + "NAME = '" + patient.getName() + "',"
            + "IC = '" + patient.getIc() + "',"
            + "ADDRESS = '" + patient.getAddress() + "',"
            + "TELEPHONE = '" + patient.getTelephone() + "',"
```

```
        + "EMAIL = '" + patient.getEmail() + "' "
        + "WHERE "
        + "ID = '" + patient.getId() + "'";
Statement stmt = conn.createStatement();
stmt.executeUpdate(updateStr);
}

public Patient selectPatientWhereIdEquals(String id)
    throws SQLException
{
    String selectStr =
        "SELECT ID, NAME, IC, ADDRESS, TELEPHONE, EMAIL "
        + "FROM PATIENTTBL "
        + "WHERE ID = '" + id + "'";
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery(selectStr);
    if(rs.next())
    {
        return new Patient (
            rs.getString("ID"),
            rs.getString("NAME"),
            rs.getString("IC"),
            rs.getString("ADDRESS"),
            rs.getString("TELEPHONE"),
            rs.getString("EMAIL")
        );
    }
    else
    {
        return null;
    }
}
}
```

Dengan mengamati kode sumber `DataAccessPatient`, Anda dapat menemukan bahwa

1. Package dari class ini adalah `net.developerforce.relieve.dataaccess`, sesuai dengan deklarasi package yang ada yaitu :
2. `package net.developerforce.relieve.dataaccess;`
3. Import yang dilakukan class ini adalah mencakup package `java.sql` dan sebuah class `net.developerforce.relieve.value.Patient`, sesuai dengan deklarasi import yang bisa ditemukan :
4. `import java.sql.*;`
5. `import net.developerforce.relieve.value.Patient;`

Anda bisa saja membuang deklarasi import, dan sebagai penggantinya Anda menuliskan deklarasi class-class yang digunakan secara lengkap. Bandingkan perubahan yang terjadi dalam potongan berikut :

```
package net.developerforce.relieve.dataaccess;

public class DataAccessPatient
{
    private java.sql.Connection conn;
```

```
public DataAccessPatient(java.sql.Connection conn)
{
    ...
}

public void
insertPatient(net.developerforce.relieve.value.Patient patient)
    throws java.sql.SQLException
{
    ...
    java.sql.Statement stmt = conn.createStatement();
    ...
}

public void
updatePatient(net.developerforce.relieve.value.Patient patient)
    throws java.sql.SQLException
{
    ...
    java.sql.Statement stmt = conn.createStatement();
    ...
}

public Patient selectPatientWhereIdEquals(String id)
    throws java.sql.SQLException
{
    ...
    java.sql.Statement stmt = conn.createStatement();
    java.sql.ResultSet rs = stmt.executeQuery(selectStr);
    if(rs.next())
    {
        return new net.developerforce.relieve.value.Patient (
            ...
        );
    }
    ...
}
}
```

Anda boleh membuang deklarasi import tetapi sebagai penggantinya Anda harus mendeklarasikan class-class yang digunakan secara lengkap : Connection sebagai java.sql.Connection, Statement sebagai java.sql.Statement, ResultSet sebagai java.sql.ResultSet dan Patient sebagai net.developerforce.relieve.value.Patient.

6. Deklarasi class yang ada dalam kode sumber ini adalah :
7. public class DataAccessPatient

Menunjukkan bahwa class DataAccessPatient diturunkan dari java.lang.Object dan bukan dari class lain.

```
8. Untuk meng-instantiate class DataAccessPatient dibutuhkan parameter bertipe
   java.sql.Connection, sebagaimana deklarasi constructor :
9.   public DataAccessPatient(Connection conn)
10.  {
11.      ...
12.  }
13. Class ini mempunyai satu variable yang dideklarasikan sebagai :
14.   private Connection conn;
15. Class DataAccessPatient mendeklarasikan tiga method :
16.   public void insertPatient(Patient patient)
17.       throws SQLException
18.   {
19.       ...
20.   }
21.
22.   public void updatePatient(Patient patient)
23.       throws SQLException
24.   {
25.       ...
26.   }
27.
28.   public Patient selectPatientWhereIdEquals(String id)
29.       throws SQLException
30.   {
31.       ...
32.   }
```

3. Proyek 1 : Membuat SalamKeadilanApp

Tujuan

Proyek SalamKeadilanApp bisa merupakan sentuhan pertama dengan teknologi Java, sebelum Anda menyelaminya lebih dalam atau mendakinya lebih tinggi. Melalui proyek ini Anda akan mengetahui bagaimana menulis aplikasi dengan bahasa pemrograman Java, meng-compile dan meluncurkannya.

Rancangan

Anda akan mengembangkan aplikasi bernama SalamKeadilanApp, yang ketika dijalankan akan menuliskan pesan "Salam Keadilan !" ke console.

Pembekalan

Pengembangan aplikasi dengan bahasa pemrograman Java pada dasarnya melalui beberapa langkah.

1. Menulis kode sumber dalam bahasa pemrograman Java, dan disimpan dalam file berekstension .java.
2. Meng-compile kode sumber tersebut menggunakan compiler yang disediakan JDK, yaitu javac. Hasilnya adalah sebuah Java class yang disimpan sebagai file berekstension .class.
3. Meluncurkan aplikasi dengan program java yang disediakan JDK.

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Langkah

Langkah 1 : Luncurkan text editor pilihan Anda

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda. Di Linux Anda dapat menggunakan kwrite. Di Windows Anda dapat menggunakan Notepad.

Langkah 2 : Menulis kode sumber

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

Langkah 3 : Simpan sebagai SalamKeadilanApp.java

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanApp.java

Kegagalan peluncuran aplikasi

- `Exception in thread "main" java.lang.NoClassDefFoundError: SalamKeadilanApp/class`
- Bisa terjadi karena Anda menjalankan :
- `$ java SalamKeadilanApp.class.`

Yang benar adalah :

- `$ java SalamKeadilanApp`
- `Exception in thread "main" java.lang.NoClassDefFoundError: SalamKeadilanApp`
- Dapat terjadi karena variabel lingkungan `CLASSPATH` yang tidak benar. Anda harus menge-set variabel lingkungan `CLASSPATH` agar mencakup directory dimana `SalamKeadilanApp.class` berada.
- `Exception in thread "main" java.lang.NoSuchMethodError: main`
- Bisa terjadi karena Anda tidak benar dalam menulis kode sumber, yaitu dalam menuliskan metoda `main`. Contoh yang benar adalah
- `public static void main(String[] args)`

Contoh yang salah adalah

```
public static void main()
```

atau

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

atau

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

4. Proyek 2 : Membuat SalamKeadilanApplet

Tujuan

Proyek SalamKeadilanApplet dirancang sebagai sentuhan pertama Anda dengan JavaApplet. Melalui proyek ini Anda akan mengetahui tahap-tahap dalam mengembangkan JavaApplet dan bagaimana agar dapat berjalan di atas browser.

Rancangan

Anda akan mengembangkan JavaApplet bernama SalamKeadilanApplet, yang ketika dijalankan akan menuliskan pesan "Salam Keadilan !" di atas browser.

Pembekalan

JavaApplet adalah program Java yang dijalankan di atas Web browser. Untuk dapat menjalankannya, Web browser terlebih dahulu harus men-download JavaApplet, baik di local machine, Intranet maupun Internet.

Berbeda dengan stand alone application yang dapat dijalankan melalui console, JavaApplet membutuhkan Web browser untuk menjalankannya. Dibandingkan dengan JSP atau JavaServlet, JavaApplet harus didownload terlebih dahulu ke Web browser, sementara JSP dan JavaServlet dijalankan di sisi server dan hanya mengembalikan HTML ke Web browser untuk ditampilkan.

Langkah-langkah dalam mengembangkan JavaApplet mencakup :

1. Menulis kode sumber dari JavaApplet sebagaimana lumrahnya kode sumber Java lainnya di simpan sebagai file dengan extension .java. Satu panduan khusus, class yang ditulis sebagai JavaApplet, harus meng-extends java.applet.Applet.
2. Meng-compile kode sumber tersebut menggunakan compiler yang disediakan JDK, yaitu javac. Hasilnya adalah sebuah Java class yang disimpan sebagai file berekstension .class.
3. Menuliskan file HTML yang memuat JavaApplet. Tag yang digunakan adalah <applet></applet>;
4. Memanggil file HTML tersebut melalui Web browser, baik dari local machine, Intranet maupun Internet.

Persiapan

- Anda membutuhkan Web browser yang Java-enable, yaitu yang mempunyai kemampuan menjalankan JavaApplet.
- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Langkah

Langkah 1 : Tulislah kode sumber

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
import java.applet.*;  
import java.awt.*;
```

```
public class SalamKeadilanApplet extends Applet
{
    public void paint(Graphics g)
    {
        Font font = new Font("Courier", Font.BOLD, 20);
        g.setFont(font);
        g.drawString("Salam keadilan !", 20, 30);
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanApplet.java

Langkah 2 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

```
$ javac SalamKeadilanApplet.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file SalamKeadilanApplet.class di dalam directory yang sama.

Langkah 3 : Luncurkan text editor lainnya untuk menulis file HTML

Untuk menulis file HTML yang dibutuhkan. Untuk memuat SalamKeadilanApplet, digunakan tag <applet> </applet>

```
<html>
<body>
    <APPLET code="SalamKeadilanApplet"
            width="400"
            height="100">
    </APPLET>
</body>
</html>
```

Simpanlah di directory yang sama dengan SalamKeadilan.class hasil proses compilation di atas.

Langkah 4 : Jalankan di atas browser

Untuk menjalankannya, bukalah file SalamKeadilan.html dari Web browser. Untuk percobaan di local machine, bukalah dari Web browser URL file:/home/lab/touch/SalamKeadilan.html



Troubleshooting

Terdapat beberapa kegagalan yang dapat terjadi :

- `SalamKeadilanapplet could not be loaded`

Dapat terjadi karena Anda salah menuliskan nama JavaApplet dalam file HTML, atau file .class tidak berada pada directory yang seharusnya.

- `java.lang.ClassCastException : SalamKeadilanApplet is not an applet`

Dapat terjadi karena Anda salah menuliskan kode sumber JavaApplet. Sebagai panduan, sebuah JavaApplet haruslah meng-extends `java.applet.Applet`.

5. Deklarasi Variable

Tujuan

Dalam latihan ini, Anda akan mencoba mendeklarasikan variable-variable dalam bahasa pemrograman Java.

Rancangan

Dalam sebuah program LatDeklarasiVariable Anda berlatih mendeklarasikan variable-variable untuk selanjutnya dicetak ke console.

Pembekalan

Di dalam bahasa pemrograman Java, sebuah variable mempunyai nama, tipe dan harga. Dalam deklarasi variable setidaknya dibutuhkan nama dan tipe.

Tata cara penulisan pertama adalah :

```
type name;
```

Dengan deklarasi ini, variable belum mempunyai harga. Contoh :

```
int var1;  
var1 = 3;
```

di mana variable bernama var1 mempunyai tipe int. Dalam statement selanjutnya, variable var1 diberi harga 3;

Tata cara penulisan lainnya, variable dapat secara langsung diberi nilai awal :

```
type name = value;
```

Contoh :

```
String var5 = "Purwokerto";
```

di mana variable bernama var5 mempunyai tipe String dan mempunyai harga awal "Purwokerto";

Contoh lainnya :

```
int var2 = 5;  
double var3 = 1.99;  
boolean var4 = false;
```

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Langkah

Langkah 1 : Tulislah kode sumber

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

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

```
{  
    int var1;  
    var1 = 3;  
  
    int var2 = 5;  
  
    double var3 = 1.99;  
  
    boolean var4 = false;  
  
    String var5= "Purwokerto";  
  
    System.out.println(var1);  
    System.out.println(var2);  
    System.out.println(var3);  
    System.out.println(var4);  
    System.out.println(var5);  
}  
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatDeklarasiVariable.java

Langkah 2 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

```
$ javac LatDeklarasiVariable.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatDeklarasiVariable.class di dalam directory yang sama.

Langkah 3 : Luncurkan dengan java

```
$ java LatDeklarasiVariable
```

```
3
```

```
5
```

```
1.99
```

```
false
```

```
Purwokerto
```

6. Bekerja dengan String dan StringBuffer

Tujuan

Dalam latihan ini, Anda akan menggunakan beberapa method dari class `java.lang.String` dan `java.lang.StringBuffer`

Rancangan

Sebuah program yang menunjukkan bagaimana menggunakan `java.lang.String` dan `java.lang.StringBuffer`

Pembekalan

Dalam bahasa pemrograman Java tersedia dua class untuk menangani string, yaitu `java.lang.String` dan `java.lang.StringBuffer`. Perbedaannya adalah bahwa `java.lang.String` adalah immutable, yaitu tidak dapat dimodifikasi, sedangkan `java.lang.StringBuffer` dapat dimodifikasi.

Sebagai gambaran, dalam potongan program berikut :

```
String str = "Salam";  
str = str + " keadilan !";
```

Meskipun tampak sederhana, dalam proses concatenation ini, JVM akan meng-create obyek baru untuk menampung hasil proses concatenation. Dalam kasus ini, obyek `str` pada baris ke-2 bukanlah obyek `str` pada baris ke-1 yang diubah, tetapi obyek `str` pada baris ke-2 adalah obyek baru yang berbeda.

Beberapa method dari class `java.lang.String` adalah :

- `equals()`
- `equalsIgnoreCase()`
- `compareTo()`
- `toUpperCase()`
- `toLowerCase()`
- `charAt()`
- `substring()`
- `indexOf()`
- `lastIndexOf()`
- `startsWith()`
- `trim()`

Beberapa method dari class `java.lang.StringBuffer` adalah :

- `append()`
- `insert()`
- `delete()`

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Latihan dengan java.lang.String

Langkah 1 : Tulislah kode sumber

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
public class LatString
{
    public static void main(String[] args)
    {
        String str1 = "Salam Keadilan !";
        String str2 = "Salam keadilan !";
        System.out.println("str1"                = " + str1);
        System.out.println("str2"                = " + str2);
        System.out.println();

        boolean boolean1 = str1.equals(str2);
        System.out.println("str1.equals(str2)"    = " + boolean1);

        boolean boolean2 = str1.equalsIgnoreCase(str2);
        System.out.println("str1.equalsIgnoreCase(str2) = " + boolean2);

        int int1 = str1.compareTo(str2);
        System.out.println("str1.compareTo(str2)"  = " + int1);

        String str3 = str1.toUpperCase();
        System.out.println("str1.toUpperCase()"    = " + str3);

        String str4 = str1.toLowerCase();
        System.out.println("str1.toLowerCase()"    = " + str4);

        char c1 = str1.charAt(6);
        System.out.println("str1.charAt(6)"        = " + c1);

        String str5 = str1.substring(6);
        System.out.println("str1.substring(6)"     = " + str5);

        String str6 = str1.substring(0,5);
        System.out.println("str1.substring(0,5)"   = " + str6);

        int int2 = str1.indexOf('l');
        System.out.println("str1.indexOf('l')"     = " + int2);

        int int3 = str1.indexOf('l', 6);
        System.out.println("str1.indexOf('l', 6)"  = " + int3);

        int int4 = str1.indexOf("adil");
        System.out.println("str1.indexOf(\"adil\")" = " + int4);

        int int5 = str1.lastIndexOf('l');
        System.out.println("str1.lastIndexOf('l')" = " + int5);

        boolean boolean3 = str1.startsWith("salam");
    }
}
```

```
System.out.println("str1.startsWith(\"salam\")    = \" + boolean3);

boolean boolean4  = str1.startsWith("Salam");
System.out.println("str1.startsWith(\"Salam\")    = \" + boolean4);

System.out.println();
String str7 = "                Salam Keadilan !                ";
System.out.println("str7                        = \" + str7);

String str8 = str7.trim();
System.out.println("str7.trim()                  = \" + str8);
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatString.java

Langkah 2 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

```
$ javac LatString.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatString.class di dalam directory yang sama.

Langkah 3 : Luncurkan dengan java

```
$ java LatString
str1                        = Salam Keadilan !
str2                        = Salam keadilan !

str1.equals(str2)           = false
str1.equalsIgnoreCase(str2) = true
str1.compareTo(str2)        = -32
str1.toUpperCase()          = SALAM KEADILAN !
str1.toLowerCase()          = salam keadilan !
str1.charAt(6)              = K
str1.substring(6)           = Keadilan !
str1.substring(0,5)         = Salam
str1.indexOf('l')           = 2
str1.indexOf('l', 6)        = 11
str1.indexOf("adil")        = 8
str1.lastIndexOf('l')       = 11
str1.startsWith("salam")    = false
str1.startsWith("Salam")    = true

str7                        =                Salam Keadilan !
str7.trim()                 = Salam Keadilan !
```

Latihan dengan java.lang.StringBuffer

Langkah 1 : Tulislah kode sumber

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

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

```
{
String str1 = "Salam keadilan !";

StringBuffer sb1 = new StringBuffer(str1);
System.out.println("Awal mula          : " + sb1.toString());
System.out.println("capacity()         = " + sb1.capacity());
System.out.println("length()           = " + sb1.length());

sb1.setCharAt(15, '?');
System.out.println("Usai setCharAt()      : " + sb1.toString());

sb1.append(" Yes, salam keadilan !");
System.out.println("Usai append()        : " + sb1.toString());

sb1.insert(14, " untuk bumi Indonesia ");
System.out.println("Usai insert()        : " + sb1.toString());

sb1.delete(6, 14);
System.out.println("Usai delete()        : " + sb1.toString());
}
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatStringBuffer.java

Langkah 2 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

```
$ javac LatStringBuffer.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatStringBuffer.class di dalam directory yang sama.

Langkah 3 : Luncurkan dengan java

```
$ java LatStringBuffer
Awal mula          : Salam keadilan !
capacity()         = 32
length()           = 16
Usai setCharAt()   : Salam keadilan ?
Usai append()      : Salam keadilan ? Yes, salam keadilan !
Usai insert()      : Salam keadilan untuk bumi Indonesia ? Yes, salam
keadilan !
Usai delete()      : Salam  untuk bumi Indonesia  ? Yes, salam keadilan
!
```

7. If

Tujuan

Dalam latihan ini, Anda akan mencoba bekerja dengan kendali aliran if.

Rancangan

Dalam sebuah program LatIfElse Anda akan menggunakan pernyataan kendali aliran if untuk memecahkan persamaan kuadrat dengan rumus abc.

Pembekalan

Pernyataan kendali aliran if akan menguji sebuah keadaan, apakah keadaan tersebut true atau false. Jika keadaan tersebut true maka kumpulan pernyataan yang berkaitan akan dijalankan, dan jika false maka tidak dijalankan.

Tata cara penulisan if adalah :

```
if(keadaan)
{
    pernyataan-pernyataan;
}
```

Contoh

```
if(x<0)
{
    System.out.println("x adalah kurang dari 0");
}
```

Maka jika $x < 0$, program akan menuliskan pesan "x adalah kurang dari 0", jika $x \geq 0$ maka program tidak melakukan apa-apa.

Kendali aliran if dapat mempunyai bagian else, dengan penulisan :

```
if(keadaan)
{
    pernyataan-pernyataan;
}
else
{
    pernyataan-pernyataan lain;
}
```

Dalam hal ini, pernyataan-pernyataan lain akan dijalankan jika keadaan adalah false. Contoh :

```
if(x<0)
{
    System.out.println("x adalah kurang dari 0");
}
else
{
    System.out.println("x adalah tidak kurang dari 0");
}
```

Maka jika $x < 0$, program akan menuliskan pesan "x adalah kurang dari 0", jika $x \geq 0$ maka program akan menuliskan pesan "x adalah tidak kurang dari 0".

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Langkah

Langkah 1 : Tulislah kode sumber

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
public class LatIfElse
{
    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);
        }
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatIfElse.java

Langkah 2 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

```
$ javac LatIfElse.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatIfElse.class di dalam directory yang sama.

Langkah 3 : Luncurkan dengan java

```
$ java LatIfElse
```

Persamaan kuadrat $a \cdot x^2 + b \cdot x + c = 0$, 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

8. Switch-case

Tujuan

Dalam latihan ini, Anda akan mencoba bekerja dengan kendali aliran switch-case.

Rancangan

Dalam sebuah program LatSwitchCase Anda akan menggunakan pernyataan kendali aliran switch-case untuk menebak nama yang diketahui inisialnya.

Pembekalan

Tata cara penulisan switch-case adalah :

```
switch(variabel)
{
    case harga1 : pernyataan-1; break;
    case harga2 : pernyataan-2; break;
    case harga3 : pernyataan-3; break;

    default : pernyataan-default;
}
yang dapat dipandang sebagai
if(variabel==harga1)
{
    pernyataan-1;
}
else
if(variabel==harga2)
{
    pernyataan-2;
}
else
if(variabel==harga3)
{
    pernyataan-3;
}
else
{
    pernyataan-default;
}
```

Dalam kendali aliran switch-case, variabel penentu harus bertipe integer atau yang kompatibel dengan integer, yaitu boolean, character, byte, short dan long.

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Langkah

Langkah 1 : Tulislah kode sumber

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
public class LatSwitchCase
{
    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);
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatSwitchCase.java

Langkah 2 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

```
$ javac LatSwitchCase.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatSwitchCase.class di dalam directory yang sama.

Langkah 3 : Luncurkan dengan java

```
$ java LatSwitchCase
Nama Anda adalah Millati
```


9. For

Tujuan

Dalam latihan ini, Anda akan mengendalikan aliran program dengan for.

Rancangan

Dalam sebuah program LatFor, Anda akan menuliskan program untuk menghitung harga factorial sebuah bilangan dengan menggunakan perulangan for.

Pembekalan

Pernyataan perulangan for akan menjalankan sekumpulan pernyataan-pernyataan, dan kemudian mengulanginya lagi selama persyaratan terpenuhi.

Tata cara penulisan for adalah :

```
for(keadaan-awal, persyaratan, pernyataan-perulangan)
{
    pernyataan-pernyataan;
}
```

Perulangan for akan menjalankan pernyataan-pernyataan mulai dari keadaan awal, selama persyaratan terpenuhi. Usai menjalankan pernyataan-pernyataan, pernyataan-perulangan akan dijalankan. Selanjutnya perulangan dimulai kembali dengan menguji persyaratan. Contoh

```
for(int i=0;i<=3;i++)
{
    System.out.println(i);
}
```

Maka perulangan akan dimulai dengan variable i berharga 0. Selanjutnya karena persyaratan $i \leq 3$ terpenuhi, maka pernyataan `System.out.println(i)` akan dijalankan. Akhirnya pernyataan `i++` dijalankan. Dan kemudian persyaratan $i \leq 3$ diuji lagi. Demikian seterusnya.

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya `/home/lab/touch`

Langkah

Langkah 1 : Tulislah kode sumber

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
public class LatFor
{
    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);  
    }  
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatFor.java

Langkah 2 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

```
# javac LatFor.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatFor.class di dalam directory yang sama.

Langkah 3 : Luncurkan dengan java

```
# java LatFor
```

Hasil

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

10. Do While

Tujuan

Dalam latihan ini, Anda akan mengendalikan aliran program dengan do-while.

Rancangan

Dalam sebuah program LatDoWhile, Anda akan menuliskan program untuk menghitung harga factorial sebuah bilangan.

Pembekalan

Pernyataan perulangan do-while akan menjalankan sekumpulan pernyataan-pernyataan, dan kemudian mengulanginya lagi selama persyaratan terpenuhi.

Tata cara penulisan do-while adalah :

```
do
{
    pernyataan pernyataan;
}
while(persyaratan terpenuhi)
```

Pernyataan perulangan do-while setidaknya akan menjalankan kumpulan pernyataan-pernyataan sebanyak satu kali, untuk kemudian menguji apakah persyaratan terpenuhi. Jika persyaratan terpenuhi maka kumpulan pernyataan-pernyataan dijalankan lagi, dan kemudian menguji kembali apakah persyaratan terpenuhi. Demikian seterusnya, kumpulan pernyataan-pernyataan akan dijalankan berulang selama persyaratan terpenuhi.

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Langkah

Langkah 1 : Tulislah kode sumber

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
public class LatDoWhile
{
    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);
    }
}

```

Simpanlah di directory yang telah Anda persiapkan sebagai LatDoWhile.java

Langkah 2 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

```
# javac LatDoWhile.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatDoWhile.class di dalam directory yang sama.

Langkah 3 : Luncurkan dengan java

```
# java LatDoWhile
```

Hasil

```

1      * 2      = 2
2      * 3      = 6
6      * 4      = 24
24     * 5      = 120
5 !          = 120

```

11. While

Tujuan

Dalam latihan ini, Anda akan mengendalikan aliran program dengan while.

Rancangan

Dalam sebuah program LatWhile, Anda akan menuliskan program untuk menghitung harga factorial sebuah bilangan.

Pembekalan

Pernyataan perulangan while akan menguji sebuah persyaratan, dan kemudian menjalankan sekumpulan pernyataan jika persyaratan terpenuhi. Usai menjalankan kumpulan pernyataan, persyaratan akan diuji kembali, dan jika terpenuhi kumpulan pernyataan akan dijalankan kembali. demikian seterusnya.

Tata cara penulisan while adalah :

```
while(persyaratan terpenuhi)
{
    pernyataan pernyataan;
}
```

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Langkah

Langkah 1 : Tulislah kode sumber

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
public class LatWhile
{
    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);
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatWhile.java

Langkah 2 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

```
# javac LatWhile.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatWhile.class di dalam directory yang sama.

Langkah 3 : Luncurkan dengan java

```
# java LatWhile
```

Hasil

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

12. Menggunakan java.lang.Math

Tujuan

Dalam latihan ini, Anda akan menggunakan beberapa method dari class java.lang.Math.

Rancangan

Sebuah program yang menunjukkan bagaimana menggunakan java.lang.Math.

Pembekalan

Class java.lang.Math mempunyai sejumlah static method yang dapat digunakan untuk melakukan perhitungan matematis. Di antaranya adalah :

- abs()
- max()
- min()
- random()
- round()
- cos()
- sin()
- tan()
- pow()

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Langkah

Langkah 1 : Tulislah kode sumber

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
public class LatMath
{
    public static void main(String[] args)
    {
        double x, y;

        x = -37.33;
        System.out.println();
        System.out.println("x" + x);
        System.out.println("Math.abs(x)" + Math.abs(x));

        x = 2.356194490192345;
        System.out.println();
        System.out.println("x" + x);
        System.out.println("Math.tan(x)" + Math.tan(x));

        x = 21;
        y = -37;
```

```
System.out.println();
System.out.println("x          = " + x);
System.out.println("y          = " + y);
System.out.println("Math.max(x,y) = " + Math.max(x,y));

x = 2;
y = 3;
System.out.println();
System.out.println("x          = " + x);
System.out.println("y          = " + y);
System.out.println("Math.pow(x,y) = " + Math.pow(x,y));

System.out.println();
System.out.print("Math.random()      :");
for(int i=0;i<3;i++)
{
    System.out.print("    " + Math.random());
}
System.out.println();

x = 225;
System.out.println();
System.out.println("x          = " + x);
System.out.println("Math.sqrt(x)    = " + Math.sqrt(x));

x = 135;
System.out.println();
System.out.println("x          = " + x);
System.out.println("Math.toRadians()= " + Math.toRadians(x));

    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatMath.java

Langkah 2 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

```
$ javac LatMath.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatMath.class di dalam directory yang sama.

Langkah 3 : Luncurkan dengan java

```
$ java LatMath
```

```
x          = -37.33
Math.abs(x) = 37.33

x          = 2.356194490192345
Math.tan(x) = -1.0000000000000002
```



```
x          = 21.0  
y          = -37.0  
Math.max(x,y) = 21.0
```

```
x          = 2.0  
y          = 3.0  
Math.pow(x,y) = 8.0
```

```
Math.random() : 0.04970817076751444 0.40194521371113345  
0.003979911650069612
```

```
x          = 225.0  
Math.sqrt(x) = 15.0
```

```
x          = 135.0  
Math.toRadians() = 2.356194490192345
```

13. Penanganan Exception

Tujuan

Anda akan mengenal beberapa tindakan yang dapat Anda lakukan berkaitan dengan Exception

Rancangan

Terdapat 3 contoh program yang akan Anda kembangkan

1. LatExceptionTryCatchApp yang merupakan contoh penggunaan try-catch-finally
2. LatExceptionThrowsApp yang merupakan contoh penggunaan throws
3. LatExceptionThrowsNewApp yang merupakan contoh men-throw java.lang.Exception baru

Pembekalan

Exception adalah pesan yang sengaja dilemparkan oleh Java Virtual Machine ketika terjadi kegagalan dalam menjalankan statement. Keadaan-keadaan yang dilaporkan sebagai exception di antaranya :

- NullPointerException akan dilemparkan jika Anda memanggil sebuah method dari sebuah variabel yang bernilai null.
- ArrayOutOfBoundsException akan dilemparkan jika Anda mencoba berinteraksi dengan elemen sebuah array yang indeksnya di luar batas.
- FileNotFoundException akan dilemparkan jika Anda mencoba mengakses sebuah File yang tidak ada.
- SQLException dilemparkan jika terjadi kesalahan, atau kegagalan dalam berinteraksi dengan basis data.
- ClassCastException dilemparkan jika Anda mencoba meng-casting sebuah obyek ke obyek lain yang tidak sesuai.

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Latihan dengan Blok Try-Catch

Langkah 1 : Dengan text editor tulislah kode sumber LatExceptionTryCatch

```
import java.io.RandomAccessFile;
import java.io.IOException;

public class LatExceptionTryCatch
{
    public void print(String fileName)
    {
        try
        {
```

```
        RandomAccessFile file = new RandomAccessFile(fileName, "r");
        String line = null;
        while((line=file.readLine()) != null)
        {
            System.out.println(line);
        }
    }
    catch(IOException ioe)
    {
        System.out.println(
            "Tertangkap di LatExceptionTryCatch.print() : "
            + ioe.toString());
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatExceptionTryCatch.java

Langkah 2 : Dengan text editor tulislah kode sumber LatExceptionTryCatchApp

```
public class LatExceptionTryCatchApp
{
    public static void main(String[] args)
    {
        if(args.length <1)
        {
            System.out.println("Cara penggunaan:
                java LatExceptionTryCatchApp <fileName>");
            System.exit(0);
        }

        String fileName = args[0];
        LatExceptionTryCatch o = new LatExceptionTryCatch();
        o.print(fileName);
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatExceptionTryCatchApp.java

Langkah 3 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Langkah 4 : Menge-set variabel lingkungan PATH dan CLASSPATH

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

```
$ export CLASSPATH=/home/lab/touch
```

Langkah 5 : Meng-compile

Melalui console, jalankan javac :

```
$ javac LatExceptionTryCatchApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatExceptionTryCatchApp.class di dalam directory yang sama.

Langkah 6 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java LatExceptionTryCatchApp myfile.txt
```

Hasilnya adalah sebuah pesan dituliskan ke console :

```
Tertangkap di LatExceptionTryCatch.print()  
:java.io.FileNotFoundException:  
myfile.txt (No such file or directory)
```

Latihan men-throws Exception

Langkah 1 : Dengan text editor tulislah kode sumber LatExceptionThrows

```
import java.io.RandomAccessFile;  
import java.io.IOException;  
  
public class LatExceptionThrows  
{  
    public void print(String fileName)  
        throws IOException  
    {  
        RandomAccessFile file = new RandomAccessFile(fileName, "r");  
        String line = null;  
        while((line=file.readLine()) != null)  
        {  
            System.out.println(line);  
        }  
    }  
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatExceptionThrows.java

Langkah 2 : Dengan text editor tulislah kode sumber LatExceptionThrowsApp

```
import java.io.IOException;  
  
public class LatExceptionThrowsApp  
{  
    public static void main(String[] args)  
    {  
        if(args.length <1)  
        {  
            System.out.println("Cara penggunaan  
                                : java LatExceptionThrowsApp ");  
            System.exit(0);  
        }  
  
        String fileName = args[0];
```

```
LatExceptionThrows o = new LatExceptionThrows();

try
{
    o.print(fileName);
}
catch(IOException ioe)
{
    System.out.println(
        "Tertangkap di LatExceptionThrowsNewApp.main() : "
        + ioe.toString());
}
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatExceptionThrowsApp.java

Langkah 3 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Langkah 4 : Menge-set variabel lingkungan PATH dan CLASSPATH

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

```
$ export CLASSPATH=/home/lab/touch
```

Langkah 5 : Meng-compile

Melalui console, jalankan javac :

```
$ javac LatExceptionThrowsApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatExceptionThrowsApp.class di dalam directory yang sama.

Langkah 6 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java LatExceptionThrowsApp myfile.class
```

Hasilnya adalah sebuah pesan dituliskan ke console :

```
Tertangkap di LatExceptionThrowsNewApp.main()
```

```
:java.io.FileNotFoundException: myfile.txt (No such file or directory)
```

Latihan men-throw Exception Baru

Langkah 1 : Dengan text editor tulislah kode sumber LatExceptionThrowsNew

```
import java.io.RandomAccessFile;
import java.io.IOException;
```

```
public class LatExceptionThrowsNew
{
    public void print(String fileName)
        throws IOException, IllegalArgumentException
    {
        int dot = fileName.lastIndexOf(".");
        String ext = fileName.substring(dot);
        boolean allowedFile = ext.equals(".txt") || ext.equals(".html")
            || ext.equals(".xml");

        if(!allowedFile)
        {
            throw new IllegalArgumentException(
                "Hanya membaca file dengan extension .txt, .html and
.xml");
        }

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

Simpanlah di directory yang telah Anda persiapkan sebagai LatExceptionThrowsNew.java

Langkah 2 : Dengan text editor tulislah kode sumber LatExceptionThrowsNewApp

```
import java.io.IOException;

public class LatExceptionThrowsNewApp
{
    public static void main(String[] args)
    {
        if(args.length <1)
        {
            System.out.println("Cara penggunaan:
            java LatExceptionThrowsNewApp <fileName>");
            System.exit(0);
        }

        String fileName = args[0];
        LatExceptionThrowsNew o = new LatExceptionThrowsNew();

        try
        {
            o.print(fileName);
        }
        catch(IOException ioe)
        {
            System.out.println(
                "Tertangkap di LatExceptionThrowsNewApp.main() : "
                + ioe.toString());
        }
    }
}
```

```
    }  
    catch(IllegalArgumentException iae)  
    {  
        System.out.println(  
            "Tertangkap di LatExceptionThrowsNewApp.main() :"  
            + iae.toString());  
    }  
}  
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai LatExceptionThrowsNewApp.java

Langkah 3 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Langkah 4 : Menge-set variabel lingkungan PATH dan CLASSPATH

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

```
$ export CLASSPATH=/home/lab/touch
```

Langkah 5 : Meng-compile

Melalui console, jalankan javac :

```
$ javac LatExceptionThrowsNewApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatExceptionThrowsNewApp.class di dalam directory yang sama.

Langkah 6 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java LatExceptionThrowsNewApp myfile.class
```

Hasilnya adalah sebuah pesan dituliskan ke console :

```
Tertangkap di LatExceptionThrowsNewApp.main()
```

```
:java.lang.IllegalArgumentException:
```

```
Hanya membaca file dengan extension .txt, .html and .xml
```

Troubleshooting

```
LatExceptionThrows.java:11: unreported exception
```

```
java.io.IOException;
```

```
must be caught or declared to be thrown
```

```
while((line=file.readLine()) != null)java: command not found
```

Menunjukkan bahwa terdapat Exception yang perlu ditangani, baik dengan blok try-catch-finally maupun dengan throws.

14. Pemrograman Multithreading

Tujuan

Dalam latihan ini Anda akan mengenal pemrograman multithreading dalam Java.

Rancangan

Anda akan mengembangkan aplikasi bernama `SalamKeadilanThreadApp` yang akan meng-instantiate beberapa Thread dan menjalankannya. Masing-masing Thread akan secara mandiri menuliskan pesan "Salam Keadilan !" ke console.

Pembekalan

Sebuah thread adalah sebuah bagian program yang bisa berjalan mandiri, sehingga dua atau lebih thread dapat berjalan bersamaan, tanpa yang satu harus menunggu selesainya yang lain. Java, sebagai pemrograman yang mendukung pemrograman multithreading, membolehkan Anda membuat dua atau lebih bagian program yang mengalir mandiri.

Untuk memberikan gambaran tentang thread, Anda perhatikanlah tempat transaksi bank, atau loket karcis kereta api atau tempat pemilihan umum, di mana terdapat antrian pelanggan untuk mendapatkan sebuah layanan. Jika ada dua atau lebih pegawai yang melayani pelanggan maka ini adalah seperti pemrograman multithreading, di mana lebih dari satu pelanggan dapat mendapat layanan pada saat bersamaan. Bandingkanlah dengan yang hanya mempunyai satu pegawai, dimana pelanggan harus mendapatkan layanan satu per satu, mengalir satu menunggu lainnya.

Gambaran lainnya, dalam pemrograman multithreading, Anda dapat membayangkan seolah alam bebas, di mana obyek-obyek seolah binatang-binatang yang bergerak masing-masing, ada yang terbang, ada yang berenang, ada yang makan dan lain-lain. Berbeda dengan beberapa rangkaian kereta api tanpa rel dua arah, maka satu rangkaian kereta api harus bergerak sesudah rangkaian kereta api lainnya.

Pemrograman multithreading merupakan sesuatu yang intrinsik di dalam bahasa Java. Setiap bagian program bekerja sebagai bagian dari suatu thread. Setidaknya dalam program stand-alone yang berjalan, terdapat sebuah thread, yaitu thread di mana metoda `main()` dipanggil.

Thread dapat dibuat dengan salah satu dari dua cara :

- meng-extend `java.lang.Thread` atau
- meng-implement `java.lang.Runnable`.

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya `/home/lab/touch`

Langkah

Langkah 1 : Dengan text editor tulislah SalamKeadilanThread.java

```
public class SalamKeadilanThread extends Thread
{
    public SalamKeadilanThread(String instanceName)
    {
        super(instanceName);
    }

    public void run()
    {
        int count =0;
        int maxCount = (int)(7*Math.random());
        while(count<maxCount)
        {
            System.out.println(
                getName()
                + "          : Salam keadilan !");
            yield();

            try
            {
                int t = (int)(1000*Math.random());
                sleep(t);
            }
            catch(InterruptedException ie)
            {
                ie.printStackTrace();
            }
            count++;
        }
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanThread.java

Langkah 2 : Dengan text editor tulislah SalamKeadilanRunnable.java

```
public class SalamKeadilanRunnable implements Runnable
{
    private String instanceName;

    public SalamKeadilanRunnable(String instanceName)
    {
        this.instanceName = instanceName;
    }

    public void run()
    {
        int count =0;
        int maxCount = (int)(7*Math.random());
        while(count<maxCount)
        {
            System.out.println(
                instanceName
```

```
        + "                : Salam keadilan !");  
        Thread.currentThread().yield();  
  
        try  
        {  
            int t = (int)(1000*Math.random());  
            Thread.currentThread().sleep(t);  
        }  
        catch(InterruptedException ie)  
        {  
            ie.printStackTrace();  
        }  
  
        count++;  
    }  
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanRunnable.java

Langkah 3 : Dengan text editor tuliskan SalamKeadilanThreadApp.java

```
public class SalamKeadilanThreadApp  
{  
    public static void main(String[] args)  
    {  
        Thread thread1  
            = new SalamKeadilanThread("Thread Pertama");  
        Thread thread2  
            = new SalamKeadilanThread("Thread Kedua");  
  
        Runnable runnable1  
            = new SalamKeadilanRunnable("Thread Ketiga");  
        Thread thread3 = new Thread(runnable1);  
  
        thread1.start();  
        thread2.start();  
        thread3.start();  
    }  
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanThreadApp.java

Langkah 4 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Langkah 5 : Menge-set variabel lingkungan PATH dan CLASSPATH

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel

lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

```
$ export CLASSPATH=/home/lab/touch
```

Langkah 6 : Meng-compile

Melalui console, jalankan javac :

```
$ javac SalamKeadilanThreadApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file SalamKeadilanThreadApp.class, SalamKeadilanRunnable.class dan SalamKeadilanThread.class di dalam directory yang sama.

Langkah 7 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java SalamKeadilanThreadApp
```

Hasilnya adalah sebuah pesan dituliskan ke console :

```
Thread Pertama      : Salam keadilan !
Thread Kedua        : Salam keadilan !
Thread Ketiga       : Salam keadilan !
Thread Ketiga       : Salam keadilan !
Thread Kedua        : Salam keadilan !
Thread Pertama      : Salam keadilan !
Thread Pertama      : Salam keadilan !
Thread Kedua        : Salam keadilan !
Thread Ketiga       : Salam keadilan !
Thread Pertama      : Salam keadilan !
Thread Pertama      : Salam keadilan !
Thread Ketiga       : Salam keadilan !
Thread Pertama      : Salam keadilan !
Thread Kedua        : Salam keadilan !
```

15. Membaca Dari Text File Dengan Random Access File

Tujuan

Dalam latihan ini Anda akan mencoba membaca plain-text file dengan `RandomAccessFile` dari package `java.io`

Rancangan

Aplikasi yang dibuat akan membaca plain-text file dan mencetak ke console.

Pembekalan

Untuk dapat bekerja dengan package `java.io`, Anda perlu meng-import ke dalam code :

```
import java.io.*;
atau boleh dituliskan untuk setiap class yang Anda butuhkan :
import java.io.File;
import java.io.RandomAccessFile;
import java.io.IOException;
```

Untuk membuat instance dari `RandomAccessFile`, dapat digunakan constructor

```
public RandomAccessFile(File file, String mode)
public RandomAccessFile(String fileName, String mode)
dengan mode bisa berupa "r" untuk read-only atau "rw" untuk read-write. Contoh :
String fileName = "/home/ekobs/todo.txt";
RandomAccessFile raf = new RandomAccessFile(fileName, "r");
atau :
String fileName = "/home/ekobs/todo.txt";
File file = new File(fileName);
RandomAccessFile raf = new RandomAccessFile(file, "r");
```

Karena operasi-operasi IO men-throw `java.io.IOException`, dan `java.io.FileNotFoundException`, Anda perlu mengembangkan exception handling untuk ini. Yaitu dengan men-throw, ataupun dengan menyusun try-catch block.

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Siapkan sebuah directory untuk latihan Anda, misalnya `/home/lab/touch`
- juga sebuah sub directory `src/touch/` di bawah sub directory tsb
- dan sebuah sub directory `classes/` di bawah sub directory tsb

Langkah

Langkah 1 : Luncurkan text editor pilihan Anda

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda. Di Linux Anda dapat menggunakan `kwrite`. Di Windows Anda dapat menggunakan Notepad.

Langkah 2 : Menulis kode sumber

```
package touch;

import java.io.*;

public class LatRandomAccessFileApp1
{
    public void printLine(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 main(String[] args)
    {
        String fileName = args[0];
        LatRandomAccessFileApp1 app = new LatRandomAccessFileApp1();
        try
        {
            app.printLine(fileName);
        }
        catch(IOException ioe)
        {
            ioe.printStackTrace();
        }
    }
}
```

Langkah 3 : Simpan sebagai LatRandomAccessFileApp1.java

Simpanlah di directory yang telah Anda persiapkan sebagai LatRandomAccessFileApp1.java di sub-directory src/touch.

Langkah 4 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

```
$ cd /home/lab/touch
```

Langkah 5 : Menge-set variabel lingkungan PATH dan CLASSPATH

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

```
$ export CLASSPATH=/home/lab/touch/classes
```

Langkah 6 : Meng-compile

Melalui console, jalankan javac :

```
$ javac -d classes/ src/touch/LatRandomAccessFileAppl.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file `LatRandomAccessFileAppl.class` di dalam directory `classes/touch`.

Langkah 7 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java. Sebagai contoh untuk membaca file `/home/lab/.ymessenger/messages.dat`, dapat dijalankan perintah berikut.

```
$ java touch.LatRandomAccessFileAppl /home/lab/.ymessenger/messages.dat
```

```
# Yahoo! Messenger Message Archive
```

```
# Copyright (c) 1999-2002 Yahoo! Inc. All Rights Reserved
```

```
# Generated: Sat Feb 8 12:09:04 2003
```

```
# This file is auto-generated by Yahoo! Messenger.
```

```
# EDIT AT YOUR OWN RISK!!!
```

16. Menulis ke Text File Dengan RandomAccessFile

Tujuan

Dalam latihan ini Anda akan mencoba menulis plain-text file dengan RandomAccessFile dari package java.io

Rancangan

Aplikasi yang dibuat akan membaca plain-text file, melakukan pembalikan baris dan selanjutnya menyimpan ke file lain.

Pembekalan

Untuk menulis sebuah baris ke dalam file dengan RandomAccessFile, digunakan :

```
writeBytes(String s)
```

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Siapkan sebuah directory untuk latihan Anda, misalnya /home/lab/touch
- juga sebuah sub directory src/touch/ di bawah sub directory tsb
- dan sebuah sub directory classes/ di bawah sub directory tsb

Langkah

Langkah 1 : Luncurkan text editor pilihan Anda

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda. Di Linux Anda dapat menggunakan kwrite. Di Windows Anda dapat menggunakan Notepad.

Langkah 2 : Menulis kode sumber

```
package touch;

import java.io.*;

public class LatRandomAccessFileApp2
{
    public void inverseLine(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();
}
public static void main(String[] args)
{
    String sourceFileName = args[0];
    String destinationFileName = args[1];
    LatRandomAccessFileApp2 app = new LatRandomAccessFileApp2();
    try
    {
        app.inverseLine(sourceFileName, destinationFileName);
    }
    catch(IOException ioe)
    {
        ioe.printStackTrace();
    }
}
}
```

Langkah 3 : Simpan sebagai LatRandomAccessFileApp2.java

Simpanlah di directory yang telah Anda persiapkan sebagai LatRandomAccessFileApp2.java di sub-directory src/touch.

Langkah 4 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

```
$ cd /home/lab/touch
```

Langkah 5 : Menge-set variabel lingkungan PATH dan CLASSPATH

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

```
$ export CLASSPATH=/home/lab/touch/classes
```

Langkah 6 : Meng-compile

Melalui console, jalankan javac :

```
$ javac -d classes/ src/touch/LatRandomAccessFileApp2.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatRandomAccessFileApp2.class di dalam directory classes/touch

Langkah 7 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java. Contoh berikut akan membalikkan file /home/lab/.ymessenger/messages.dat.

Lihat muatan dari file messages.dat :

```
$ more /home/lab/.ymessenger/messages.dat
# Yahoo! Messenger Message Archive
# Copyright (c) 1999-2002 Yahoo! Inc. All Rights Reserved

# Generated: Sat Feb  8 12:09:04 2003
# This file is auto-generated by Yahoo! Messenger.
# EDIT AT YOUR OWN RISK!!!
```

Balikkan setiap baris dan simpan ke file1.txt :

```
$ java touch.LatRandomAccessFileApp2 /home/lab/.ymessenger/messages.dat
file1.txt
```

Lihat muatan dari file file1.txt :

```
$ more file1.txt
evihcrA egasseM regnesseM !ooHaY #
devreseR sthgIR l1A .cnI !ooHaY 2002-9991 )c( thgirypOC #

3002 40:90:21 8 beF taS :detareneG #
.regnesseM !ooHaY yb detareneg-otua si elif sihT #
!!!KSIR NWO RUOY TA TIDE #
```

17. Bekerja dengan AWT

Tujuan

Mengenal contoh sederhana aplikasi berbasis GUI dalam Java dengan AWT (Abstract Windowing Toolkit). Class yang akan digunakan adalah `java.awt.Frame`, `java.awt.Button`, `java.awt.Label`. Juga akan digunakan interface `java.awt.event.EventListener`, dan class terkait `java.awt.event.ActionEvent` untuk menangani event yang terjadi jika komponen `java.awt.Button` ditekan.

Rancangan

Dalam latihan ini Anda akan mencoba membuat sebuah window. Di atas window tersebut Anda meletakkan sebuah tombol, yang jika ditekan akan menampilkan pesan "Salam Keadilan !"

Pembekalan

AWT (Abstract Windowing Toolkit) merupakan package yang diperkenalkan untuk mengembangkan aplikasi berbasis GUI. Dalam AWT, tercakup :

1. Class-class yang digunakan untuk meletakkan class-class lain, atau disebut container. Di antaranya adalah `JFrame`, `JPanel`, `JDialog`.
2. Class-class yang berkaitan dengan komponen-komponen untuk GUI seperti `Button`, `TextField`, `Label`, `RadioButton`.
3. Class, interface dan adapter yang digunakan untuk penanganan event, seperti `EventListener`, dan `ActionEvent`.

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya `/home/lab/touch`

Langkah

Langkah 1 : Luncurkan text editor pilihan Anda

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda. Di Linux Anda dapat menggunakan `kwrite`. Di Windows Anda dapat menggunakan `Notepad`.

Langkah 2 : Menulis kode sumber untuk Frame

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

public class SalamKeadilanFrame
    extends Frame
    implements ActionListener
{
    Label labelPesan;

    public SalamKeadilanFrame()
    {
```

```
        Button tombolPesanan = new Button("Tombol Pesanan");
        tombolPesanan.addActionListener(this);
        add(tombolPesanan, BorderLayout.NORTH);

        labelPesanan = new Label();
        add(labelPesanan, BorderLayout.CENTER);

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

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

Langkah 3 : Simpan sebagai SalamKeadilanFrame.java

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanFrame.java

Langkah 4 : Dengan teks editor mulai tulis file untuk SalamKeadilanAWTApp.java

Ini adalah main class yang ketika dijalankan akan meng-instantiate SalamKeadilanFrame dan menampilkannya.

```
import java.awt.*;

public class SalamKeadilanAWTApp
{
    public static void main(String[] args)
    {
        Frame frame = new SalamKeadilanFrame();
        frame.setLocation(300, 300);
        frame.setSize(200, 100);
        frame.show();
    }
}
```

Langkah 5 : Simpan sebagai SalamKeadilanAWTApp.java

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanAWTApp.java

Langkah 6 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Langkah 7 : Menge-set variabel lingkungan PATH dan CLASSPATH

Lakukan set atas variabel lingkungan yang dibutuhkan.

Langkah 8 : Meng-compile

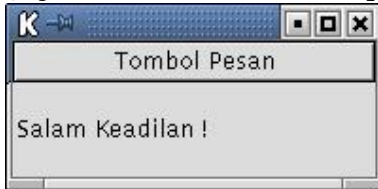
Dengan :.

```
$ javac SalamKeadilanAWTApp.java
```

Langkah 9 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java SalamKeadilanAWTApp
```



Hasilnya adalah sebuah window dengan tombol di sana, yang jika ditekan menampilkan teks "Salam Keadilan !"

18. Bekerja dengan Swing

Tujuan

Mengenal contoh sederhana aplikasi berbasis GUI dalam Java dengan Swing Class yang akan digunakan adalah `javax.swing.JFrame`, `javax.swing.JButton`, `java.swing.JLabel`. Serta pengenalan tentang Look and Feel dari aplikasi Swing yang dapat diatur ke Window, Motif atau Metal.

Rancangan

Dalam latihan ini Anda akan mencoba membuat sebuah window. Di atas window tersebut Anda meletakkan sebuah tombol, yang jika ditekan akan menampilkan pesan "Salam Keadilan !"

Pembekalan

Swing pengembangan lebih lanjut dari AWT untuk mengembangkan aplikasi berbasis GUI. Class-class dari AWT mempunyai padanannya dalam Swing, seperti `java.awt.Frame` dengan `javax.swing.JFrame`, `java.awt.Button` dengan `javax.swing.JButton` dan `java.awt.TextField` dengan `javax.swing.JTextField`.

Swing memperkenalkan teknologi baru, di antaranya :

1. Look and feel.

Swing juga merupakan teknologi yang lightweight, dalam makna lebih independen terhadap platform. Berbeda dengan AWT yang heavyweight, di mana tingkah laku AWT sangat tergantung kepada platform dan bisa menghasilkan perbedaan tingkah laku di atas platform yang berbeda.

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya `/home/lab/touch`

Langkah

Langkah 1 : Luncurkan text editor pilihan Anda

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda. Di Linux Anda dapat menggunakan `kwrite`. Di Windows Anda dapat menggunakan Notepad.

Langkah 2 : Menulis kode sumber untuk Frame

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

public class SalamKeadilanJFrame
    extends JFrame
    implements ActionListener
```

```
{
    JLabel labelPesana;

    public SalamKeadilanJFrame()
    {
        JButton tombolPesana = new JButton("Tombol Pesana");
        tombolPesana.addActionListener(this);
        getContentPane().add(tombolPesana, BorderLayout.NORTH);

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

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

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

Langkah 3 : Simpan sebagai SalamKeadilanJFrame.java

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanJFrame.java

Langkah 4 : Dengan teks editor mulai tulis file untuk SalamKeadilanSwingApp.java

Ini adalah main class yang ketika dijalankan akan meng-instantiate SalamKeadilanJFrame dan menampilkannya.

```
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 5 : Simpan sebagai SalamKeadilanSwingApp.java

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanSwingApp.java

Langkah 6 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Langkah 7 : Menge-set variabel lingkungan PATH dan CLASSPATH

Lakukan set atas variabel lingkungan yang dibutuhkan.

Langkah 8 : Meng-compile

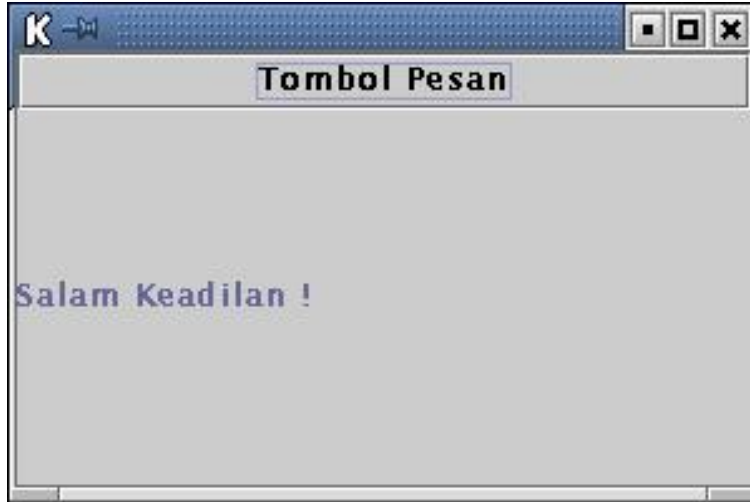
Dengan :.

```
$ javac SalamKeadilanSwingApp.java
```

Langkah 9 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java SalamKeadilanSwingApp
```

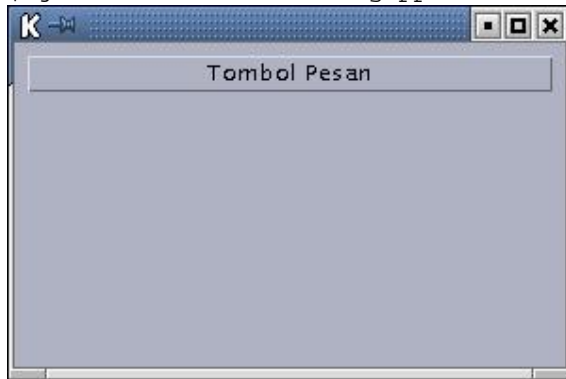


Hasilnya adalah sebuah window dengan tombol di sana, yang jika ditekan menampilkan teks "Salam Keadilan !" Look and feel yang digunakan dalam hal ini adalah metal

Langkah 9 : Mencoba look and feel lainnya

Untuk mencoba menggunakan look and feel lainnya, misalnya motif dapat digunakan

```
$ java SalamKeadilanSwingApp motif
```



Hasilnya adalah sebuah window dengan look and feel yang berbeda.

19. Menampilkan Data dari DB dengan JTable

Tujuan

Dalam latihan ini Anda akan menggunakan `javax.swing.JTable` untuk menampilkan data dari Database.

Rancangan

Anda akan mengembangkan aplikasi yang mengakses database dengan JDBC kemudian menampilkannya dengan `javax.swing.JTable`.

Pembekalan

...

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya `/home/lab/touch`
- Anda membutuhkan JDBC Driver

Langkah

Langkah 1 : Dengan text editor tulislah Patient.java

```
public class Patient
{
    private String name;
    private String pin;
    public Patient(String pin, String name)
    {
        this.name = name;
        this.pin = pin;
    }

    public String getPin()
    {
        return this.pin;
    }

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

Simpanlah di directory yang telah Anda persiapkan sebagai Patient.java

Langkah 2 : Dengan text editor tulislah PatientDataAccess.java

```
import java.util.Vector;
import java.sql.*;
```

```
public class PatientDataAccess
{
    public Vector getAllPatients()
    {
        Vector v = new Vector();

        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        try
        {

            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

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

            conn = DriverManager.getConnection(
                url, user, pwd);
            stmt = conn.createStatement();

            String selectSQL =
                "SELECT PIN, NAME FROM PATIENTTBL";
            rs = stmt.executeQuery(selectSQL);

            while(rs.next())
            {
                String pin = rs.getString("PIN");
                String name = rs.getString("NAME");
                Patient patient = new Patient(pin, name);
                v.addElement(patient);
            }

        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
        return v;
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai PatientDataAccess.java

Langkah 3 : Dengan text editor tulislah PatientTableModel.java

```
import javax.swing.table.*;
import java.util.Vector;

public class PatientTableModel extends AbstractTableModel
{
    private String[] daftarKolom
        = {"Patient ID", "Name"};
    private String[][] data;
```

```
public PatientTableModel(Vector patients)
{
    int jumlahBaris = patients.size();
    data = new String[jumlahBaris][];

    for(int iterator=0;iterator<jumlahBaris;iterator++)
    {
        Patient patient =
            (Patient)patients.elementAt(iterator);
        data[iterator] = new String[2];
        data[iterator][0] = patient.getPin();
        data[iterator][1] = patient.getName();
    }
}

public int getRowCount()
{
    return this.data.length;
}

public int getColumnCount()
{
    return daftarKolom.length;
}

public String getColumnName(int columnIndex)
{
    return daftarKolom[columnIndex];
}

public Object getValueAt(int rowIndex, int columnIndex)
{
    return data[rowIndex][columnIndex];
}
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai PatientTableModel.java

Langkah 4 : Dengan text editor tulislah PatientJTableApp.java

```
import java.util.Vector;
import java.sql.SQLException;

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

public class PatientJTableApp
{
    public static void main(String[] args)
    {
        PatientDataAccess dataAccess = new PatientDataAccess();

        Vector patients
            = dataAccess.getAllPatients();
```

```
PatientTableModel model
    = new PatientTableModel(patients);
JTable table = new JTable(model);
JScrollPane pane
    = new JScrollPane(table);

JFrame frame
    = new JFrame(
        "Daftar Patient dengan JTable");
frame.getContentPane().add(pane);

WindowAdapter wa = new WindowAdapter()
{
    public void windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
};

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

Simpanlah di directory yang telah Anda persiapkan sebagai PatientJTableApp.java

Langkah 5 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Langkah 6 : Menge-set variabel lingkungan PATH dan CLASSPATH

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

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

Langkah 7 : Meng-compile

Melalui console, jalankan javac :

```
$ javac PatientJTableApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file PatientJTableApp.class, Patient.class, PatientDataAccess.class dan PatientTableModel.class di dalam directory yang sama.

Langkah 8 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java PatientJTableApp
```

Hasilnya adalah sebuah pesan dituliskan ke console :



The image shows a screenshot of a Java Swing window titled "Daftar Patient dengan jTable". The window contains a table with two columns: "Patient ID" and "Name". The table has four rows of data. The window has a standard Mac OS X-style title bar with a red close button, a yellow maximize button, and a green window button.

Patient ID	Name
M201	Muhammad Ilham
K099	Kwik Tan Tong
C208	Cheerla Srinivasan
D898	Diana Laila

20. Menampilkan Data dari DB dengan JApplet

Tujuan

Dalam latihan ini Anda akan menggunakan `javax.swing.JTable` untuk menampilkan data dari Database di atas `javax.swing.JApplet`

Rancangan

Anda akan mengembangkan aplikasi yang mengakses database dengan JDBC kemudian menampilkannya dengan `javax.swing.JTable` di atas `javax.swing.JApplet`

Pembekalan

...

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya `/home/lab/touch`
- Anda membutuhkan JDBC Driver
- Anda perlu meng-install Java Plugin sehingga browser Anda siap menangani JApplet

Langkah

Langkah 1 : Persiapkan `Patient.java`, `PatientDataAccess.java` dan `PatientTableModel.java`.

Kode sumber untuk class-class ini dapat didapatkan dari bahan tutorial yang lalu.

Langkah 2 : Dengan text editor tulislah `PatientJTableJApplet.java`

```
import java.util.Vector;
import java.sql.SQLException;

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

public class PatientJTableJApplet extends JApplet
{
    public void init()
    {
        PatientDataAccess dataAccess = new PatientDataAccess();

        Vector patients
            = dataAccess.getAllPatients();
        PatientTableModel model
            = new PatientTableModel(patients);
        JTable table = new JTable(model);
        JScrollPane pane
```

```
        = new JScrollPane(table);  
        getContentPane().add(pane);  
    }  
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai PatientJTableApp.java

Langkah 3 : Melalui console lakukan compile

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Untuk dapat sukses melakukan kompilasi , Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Melalui console, jalankan javac :

```
$ javac PatientJTableJApplet.java
```

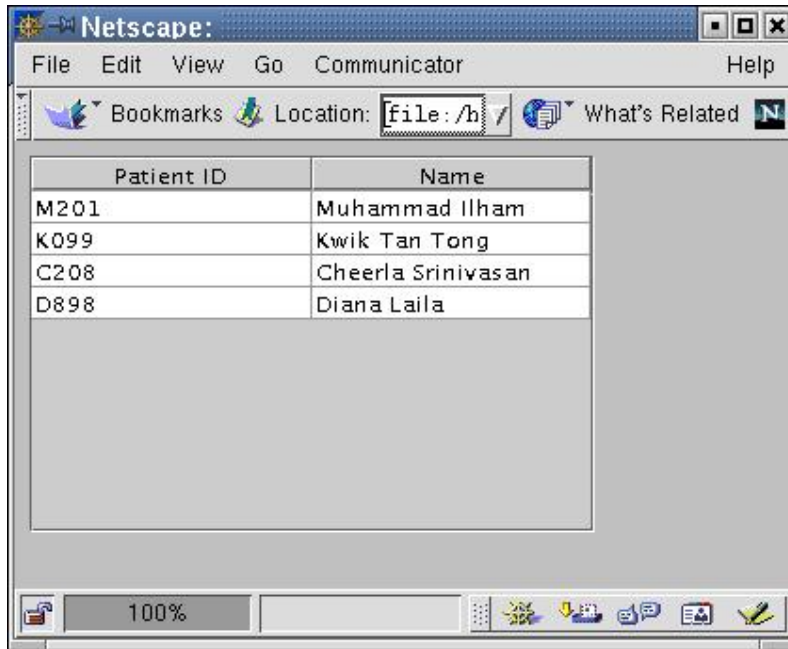
Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file PatientJTableApp.class, Patient.class, PatientDataAccess.class dan PatientTableModel.class, PatientJTableApplet.class di dalam directory yang sama.

Langkah 4 : Dengan text editor tulislah patientjtablejapplet.html

```
<html>  
<body>  
    <OBJECT  
classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"  
    width="300" height="200" align="middle"  
    codebase="http://java.sun.com/products/plugin/1.2/  
        jinstall-12-win32.cab#Version=1,1,2,0">  
        <PARAM NAME="code"  
VALUE="PatientJTableJApplet.class">  
        <PARAM NAME="codebase"  
VALUE="mm.mysql-2.0.8-bin.jar">  
        <COMMENT>  
        <EMBED type="application/x-java-applet;version=1.2"  
            width="300" height="200" align="middle"  
            code="PatientJTableJApplet.class"  
            codebase=".">  
        </EMBED>  
        </COMMENT>  
        <NOEMBED>  
        </COMMENT>  
        No support for JDK 1.2 applets found!  
        </NOEMBED>  
        </EMBED>  
    </OBJECT>  
</body>  
</html>
```

Langkah 5 : Meluncurkan Applet melalui browser

Untuk meluncurkan aplikasi Anda, melalui console, dengan Netscape buka file:/home/lab/touch/patientjtablejapplet.html. Hasilnya dapat diamati di layar Netscape :



Troubleshooting

```
java.lang.NoClassDefFoundError: javax/swing/JApplet
```

Anda perlu meng-install Java Plug In. Untuk Linux dapat didownload dari <http://blackdown.org>, bernama JavaPlugIn-1.2.2-FCS-linux-i386-glibc-2.1.3.run

```
$ sh JavaPlugIn-1.2.2-FCS-linux-i386-glibc-2.1.3.run
```

```
Java(TM) Plug-in 1.2.2 Pre-Release  
Binary Code Evaluation License
```

```
SUN MICROSYSTEMS, INC., THROUGH ITS JAVASOFT BUSINESS ("SUN") IS  
WILLING TO LICENSE THE ACCOMPANYING JAVA(tm) PLUG-IN 1.2.2 PRE-RELEASE  
SOFTWARE AND DOCUMENTATION INCLUDING AUTHORIZED COPIES OF EACH (THE  
"SOFTWARE") TO LICENSEE ONLY ON THE CONDITION THAT LICENSEE ACCEPTS  
ALL OF THE TERMS IN THIS AGREEMENT.
```

```
...
```

```
Do you agree to the above license terms?  
If you do not agree to the terms, installation cannot proceed  
Please type yes or no.
```

```
yes  
Unpacking into install.sfx.1418  
Unpacking...  
Validating checksum...  
Installing Java(tm) Plug-in into /home/lab/.netscape ...
```



```
Netscape directory is writable. Install proceeding  
Removing old versions...  
UnTarring file  
Prodding navigator to welcome page. Assume navigator is running...  
Setting bookmarks in navigator. Assumes navigator is running...  
Java(tm) Plug-in installation done.  
java.lang.ClassNotFoundException : org.gjt.mm.mysql.Driver
```

Anda perlu menjadikan JDBC Driver untuk mySQL terjangkau oleh Netscape, dengan meng-copy ke dalam directory terkait :

```
$ cp /home/lab/lib/mm.mysql-2.0.8/mm.mysql-2.0.8-bin.jar  
/home/lab/.netscape/java/lib/
```

21. Menjalankan Applet sebagai Aplikasi Stand-alone

Tujuan

Dalam latihan ini Anda akan mencoba menjalankan sebuah applet, dikembangkan dari JApplet, sebagai standalone application.

Rancangan

Anda akan menggunakan applet yang telah dikembangkan pada saat yang lalu, yaitu PatientJTableJApplet. Applet ini akan di-instantiate di sebuah main() dan diletakkan diatas JFrame.

Pembekalan

Applet adalah subclass dari Panel, sehingga Anda dapat meletakkan Applet di atas Frame sebagaimana Anda meletakkan Panel di atas Frame. Hal yang sama berlaku dengan JApplet, JPanel dan JFrame.

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch
- Anda membutuhkan JDBC Driver

Langkah

Langkah 1 : Persiapkan Patient.java, PatientDataAccess.java dan PatientTableModel.java, dan Patient JTableJApplet.java

Kode sumber untuk class-class ini dapat didapatkan dari bahan tutorial yang lalu.

Langkah 2 : Dengan text editor tulislah PatientJTableJAppletApp.java

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

public class PatientJTableJAppletApp
{
    public static void main(String[] args)
    {
        PatientJTableJApplet jApplet = new PatientJTableJApplet();

        JFrame frame
            = new JFrame(
                "Daftar Patient dengan jTable");

        frame.getContentPane().add(jApplet);
        jApplet.init();

        WindowAdapter wa = new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
```

```
        {  
            System.exit(0);  
        }  
    };  
  
    frame.pack();  
    frame.show();  
}  
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai PatientJTableJAppletApp.java

Langkah 3 : Melalui console lakukan compile

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Untuk dapat sukses melakukan kompilasi , Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Melalui console, jalankan javac :

```
$ javac PatientJTableJAppletApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file PatientJTableApp.class, Patient.class, PatientDataAccess.class dan PatientTableModel.class, PatientJTableApplet.class dan PatientJTableJAppletApp.class di dalam directory yang sama.

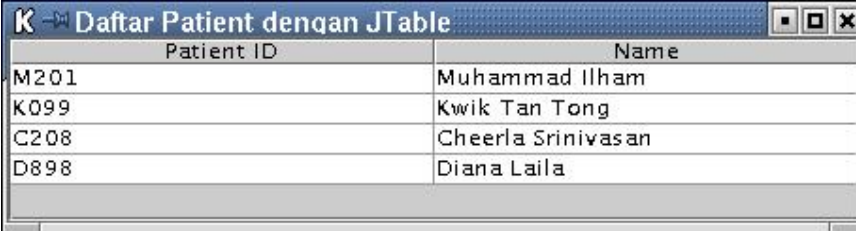
Langkah 4 : Menjalankan aplikasi

Pertama, Anda perlu menge-set CLASSPATH agar memuat JDBC Driver yang dibutuhkan :

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

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java PatientJTableJAppletApp
```



Patient ID	Name
M201	Muhammad Ilham
K099	Kwik Tan Tong
C208	Cheerla Srinivasan
D898	Diana Laila

Troubleshooting

Diskusi

Dapat diperhatikan bahwa untuk menjadikan Applet sebagai sebuah aplikasi stand-alone, maka applet perlu di-instantiate dan di-init dalam method main(). Dalam latihan ini, Anda mengembangkan satu class terpisah untuk memuat main(), dengan alasan kesederhanaan untuk belajar. Anda bisa memasukkan main() ini ke dalam deklarasi class dari Applet yang Anda kembangkan, sedemikian sehingga sebuah class yang sama dapat dijalankan sebagai Applet dan sebagai aplikasi stand-alone.

22. Pemrograman Grafis dengan Java

Tujuan

Mengenalkan pemrograman grafis dalam Java yang membolehkan Anda menggambar line, rectangle dan lain-lain.

Rancangan

Anda akan mengembangkan sebuah program yang menampilkan jam manual berbentuk oval.

Pembekalan

...

Persiapan

- Anda membutuhkan Web browser yang Java-enable, yaitu yang mempunyai kemampuan menjalankan JavaApplet.
- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Class Utama : Jam Oval

Langkah 1 : Tulislah kode sumber JamOval.java

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
import java.util.Date;

import java.applet.*;
import java.awt.*;

public class JamOval
{
    public JamOval(Image buffer)
    {
        this.buffer = buffer;
        preparedJamOval();
    }

    private int lebar = 350;
    private int tinggi = 250;

    private int titikTengahX = 175;
    private int titikTengahY = 125;

    private int jariJariX = 150;
    private int jariJariY = 100;

    private double jarumDetik = 0.9;
    private double jarumMenit = 0.8;
    private double jarumJam = 0.5;

    private double detik, detikLama;
```

```
private double menit, menitLama;
private double jam, jamLama;

Image buffer = null;

public void preparedJamOval()
{
    Graphics g = buffer.getGraphics();

    g.setColor(Color.white);

    g.fillOval(0,0, lebar, tinggi);

    for(int iterator=0;iterator<12;iterator++)
    {
        double sudut = iterator*(Math.PI/6);
        int xPos      = getX(sudut,1);
        int yPos      = getY(sudut,1);
        g.setColor(Color.blue);
        g.drawRoundRect(xPos-5,yPos-5,10,10,3,3);
    }
}

public void repaintBuffer()
{
    Graphics g = buffer.getGraphics();

    double sudutDetik
        = (Math.PI*2*detikLama)/60-Math.PI*0.5;
    double sudutMenit
        = (Math.PI*2*menitLama)/60-Math.PI*0.5;
    double sudutJam
        = (Math.PI*2*jamLama)/12-Math.PI*0.5;

    g.setColor(Color.white);
    g.drawLine(getXCenter(), getYCenter(),
        getX(sudutDetik, jarumDetik),
        getY(sudutDetik, jarumDetik));
    g.drawLine(getXCenter(), getYCenter(),
        getX(sudutMenit, jarumMenit),
        getY(sudutMenit, jarumMenit));
    g.drawLine(getXCenter(), getYCenter(),
        getX(sudutJam, jarumJam),
        getY(sudutJam, jarumJam));

    g.setColor(Color.magenta);
    sudutDetik
        = (Math.PI*2*detik)/60-Math.PI*0.5;
    sudutMenit
        = (Math.PI*2*menit)/60-Math.PI*0.5;
    sudutJam
        = (Math.PI*2*jam)/12-Math.PI*0.5;

    g.drawLine(getXCenter(), getYCenter(),
        getX(sudutDetik, jarumDetik),
        getY(sudutDetik, jarumDetik));
```

```
        g.drawLine(getXCenter(), getYCenter(),
                    getX(sudutMenit, jarumMenit),
                    getY(sudutMenit, jarumMenit));
        g.drawLine(getXCenter(), getYCenter(),
                    getX(sudutJam, jarumJam),
                    getY(sudutJam, jarumJam));
    }

    public void start()
    {
        Date date = new Date();
        long miliSecond = date.getTime();

        detik = date.getSeconds();
        menit = date.getMinutes();
        jam = date.getHours();
    }

    public void tick()
    {
        detikLama = detik;
        detik      += 0.1;
        detik      = detik%60;

        menitLama = menit;
        menit += 0.1/60;
        menit = menit%60;

        jamLama = jam;
        jam += (0.1/60)/60;
        jam = jam%12;
        repaintBuffer();
    }

    public int getXCenter()
    {
        return titikTengahX;
    }

    public int getYCenter()
    {
        return titikTengahY;
    }

    public int getX(double sudut, double jarum)
    {
        double d = titikTengahX
            + (jariJariX*jarum*Math.cos(sudut));
        return (int)Math rint(d);
    }

    public int getY(double sudut, double jarum)
    {
        double d = titikTengahY
            + (jariJariY*jarum*Math.sin(sudut));
        return (int)Math rint(d);
    }
}
```

```
}  
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai JamOval.java

Jam Oval sebagai stand alone GUI application

Langkah 2 : Tulislah kode sumber JamOvalPanel.java

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
import java.awt.*;  
  
public class JamOvalPanel  
    extends Panel  
    implements Runnable  
{  
    private int lebar = 350;  
    private int tinggi = 250;  
  
    JamOval jamOval = null;  
    Image buffer = null;  
  
    public void init()  
    {  
        buffer = createImage(lebar, tinggi);  
        jamOval = new JamOval(buffer);  
        Thread t = new Thread(this);  
        t.start();  
    }  
  
    public void paint(Graphics g)  
    {  
        g.drawImage(buffer, 0,0, null);  
    }  
  
    public void update(Graphics g)  
    {  
        paint(g);  
    }  
  
    public void run()  
    {  
        jamOval.start();  
        while(true)  
        {  
            try  
            {  
                Thread.sleep(100);  
            }  
            catch(InterruptedException ie)  
            {  
                ie.printStackTrace();  
            }  
            jamOval.tick();  
            repaint();  
        }  
    }  
}
```


Simpanlah di directory yang telah Anda persiapkan sebagai JamOvalPanel.java

Langkah 3 : Tulislah kode sumber JamOvalApp.java

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
import java.awt.*;

public class JamOvalApp
{
    public static void main(String[] args)
    {
        Frame frame = new Frame();
        frame.setSize(400,300);
        frame.show();
        JamOvalPanel jamOvalPanel = new JamOvalPanel();
        frame.add(jamOvalPanel);
        jamOvalPanel.init();
        frame.validate();
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai JamOvalApp.java

Langkah 4 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

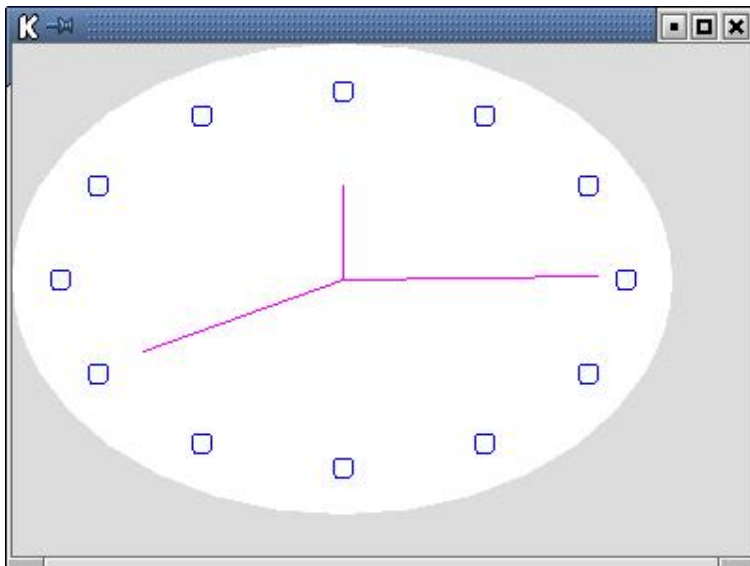
```
$ javac JamOvalApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file JamOvalApp.class, JamOvalPanel.class dan JamOval.class di dalam directory yang sama.

Langkah 5 : Luncurkan aplikasi

```
$ java JamOvalApp
```

Hasilnya :



Jam Oval sebagai Applet

Langkah 6 : Tulislah kode sumber JamOvalApplet.java

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda.

```
import java.util.Date;

import java.applet.*;
import java.awt.*;

public class JamOvalApplet
    extends Applet
    implements Runnable
{
    private int lebar  = 350;
    private int tinggi = 250;

    JamOval jamOval = null;
    Image buffer = null;

    public void init()
    {
        buffer = createImage(lebar, tinggi);
        jamOval = new JamOval(buffer);
        Thread t = new Thread(this);
        t.start();
    }

    public void paint(Graphics g)
    {
        g.drawImage(buffer, 0,0, null);
    }

    public void update(Graphics g)
    {
        paint(g);
    }

    public void run()
    {
        jamOval.start();
        while(true)
        {
            try
            {
                Thread.sleep(100);
            }
            catch(InterruptedException ie)
            {
                ie.printStackTrace();
            }
            jamOval.tick();
            repaint();
        }
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai JamOvalApplet.java

Langkah 7 : Lakukan compilation dengan javac

Setelah terlebih dahulu set variabel lingkungan PATH dan CLASSPATH.

```
$ javac JamOvalApplet.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file JamOvalApplet.class, dan JamOval.class di dalam directory yang sama.

Langkah 8 : Luncurkan text editor lainnya untuk menulis file HTML

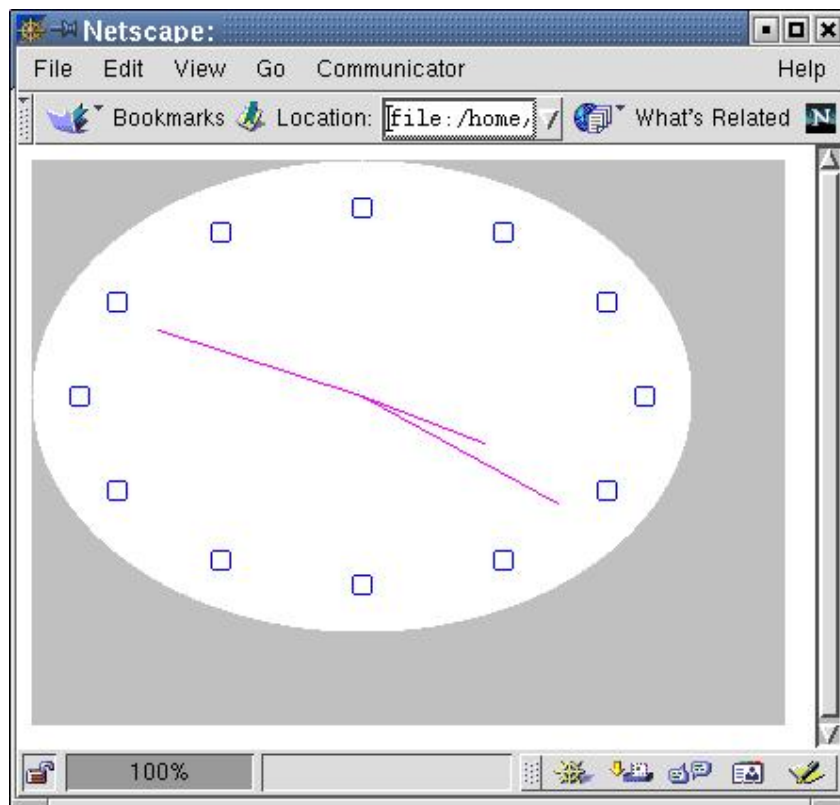
Untuk menulis file HTML yang dibutuhkan. Untuk memuat SalamKeadilanApplet, digunakan tag <applet> </applet>

```
<html>
<body>
    <APPLET code="JamOvalApplet"
            width="400"
            height="300">
    </APPLET>
</body>
</html>
```

Simpanlah di directory yang sama dengan JamOvalApplet.class hasil proses compilation di atas.

Langkah 9 : Jalankan di atas browser

Untuk menjalankannya, bukalah file SalamKeadilan.html dari Web browser. Untuk percobaan di local machine, bukalah dari Web browser URL file:/home/lab/touch/JamOvalApplet.html



23. Pemrograman Jaringan

Tujuan

Dalam latihan ini Anda akan mengenal pemrograman jaringan dalam Java.

Rancangan

Anda akan mengembangkan aplikasi client server sederhana, di mana sebuah server berjalan dan jika dihubungi oleh sebuah client akan mengirimkan pesan "Salam Keadilan !"

Pembekalan

Pemrograman jaringan dikembangkan untuk menyambungkan proses-proses komputasi yang lumrahnya berjalan dalam mesin-mesin yang berbeda yang terhubung dalam suatu jaringan komputer. Contohnya adalah :

- Web browser mengakses Web server melalui jaringan.
- Aplikasi basis data, di mana terdapat client dan server yang berada di mesin komputer yang terpisah, dan berinteraksi dalam dalam jaringan.
- Aplikasi e-mail di mana terdapat mail client berkomunikasi dengan mail server.

Dalam pemrograman jaringan, inti tantangannya adalah bagaimana Anda dapat mengirimkan data dari sebuah proses ke proses lainnya melalui pipa yang menghubungkan keduanya. Pipa antara dua proses ini dapat dianalogikan dengan saluran telepon antara dua pembicara. Dalam analogi sambungan telepon, seorang pembicara dapat bercakap di satu sisi , kemudian suaranya akan mengalir melalui kabel telepon ke lawan bicaranya, sehingga lawan bicara dapat mendengarnya, dan sebaliknya.

Dalam pemrograman jaringan, setelah Anda membuat sebuah pipa data antara dua proses, maka masing-masing proses akan memegang satu ujung. Setiap ujung dari pipa data ini disebut socket. Aliran data antara kedua socket ini disebut data stream. Masing-masing socket menggenggam dua stream, yaitu output stream dan input stream.

Sebuah proses dapat mengirim data ke proses lain dengan menuliskannya ke output stream yang dimiliki sebuah socket. Sebuah proses dapat menerima data dengan membacanya dari input stream yang dimiliki sebuah socket. Dengan cara ini maka dapat terjadi pertukaran data, komunikasi dan interaksi antara dua proses yang berbeda meskipun mereka terpisah dalam jaringan. Proses-proses ini dapat terpisah tetapi terjaring, baik dalam intranet, extranet ataupun internet, tidak peduli apakah satu proses berada di Afrika Tengah dan proses lainnya berada di tengah Samudra Pasifik.

Class-class untuk mengembangkan pemrograman jaringan dikumpulkan dalam package java.net, yang mempunyai dua class utama yaitu dan

- java.net.ServerSocket, digunakan untuk membangun server
- java.net.Socket, digunakan untuk membangun pipa antara server dan client

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch

Langkah

Langkah 1 : Dengan text editor tulislah SalamKeadilanServer.java

```
import java.io.*;
import java.net.*;

public class SalamKeadilanServer
{
    private String namaServer;
    private ServerSocket serverSocket;

    public SalamKeadilanServer(String namaServer, int port)
        throws IOException
    {
        this.namaServer = namaServer;
        System.out.println("Menghidupkan server bernama :
                           " + namaServer + " di port : " + port);
        serverSocket = new ServerSocket(port);
    }

    public void operateService()
        throws IOException
    {
        System.out.println(namaServer + " menanti connection ...
");
        Socket socket = serverSocket.accept();
        String alamatClient =
socket.getInetAddress().getHostAddress();

        System.out.println(namaServer + " menangkap connection
dari "
                           + alamatClient);

        InputStream inputStream = socket.getInputStream();
        InputStreamReader inputStreamReader = new
InputStreamReader(inputStream);
        BufferedReader bufferedReader = new
BufferedReader(inputStreamReader);

        OutputStream outputStream = socket.getOutputStream();
        PrintWriter printWriter = new PrintWriter(outputStream,
true);

        String namaClient = bufferedReader.readLine();
        System.out.println(namaServer + " membaca nama client :
"
                           + namaClient);

        System.out.println(namaServer + " mengirim pesan : ");
```

```
String pesanDikirim = "Pesan dari " + namaServer + ", untuk  
"  
                + namaClient + "di " + alamatClient +  
                " : Salam Keadilan !";  
System.out.println(pesanDikirim);  
printWriter.println(pesanDikirim);  
  
bufferedReader.close();  
printWriter.close();  
socket.close();  
System.out.println(namaServer + " menutup connection.");  
    }  
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanServer.java

Langkah 2 : Dengan text editor tulislah SalamKeadilanServerApp.java

```
public class SalamKeadilanServerApp  
{  
    public static void main(String[] args)  
    {  
        if(args.length<2)  
        {  
            System.out.println("Cara penggunaan  
:java SalamKeadilanServerApp  
<namaServer> <port>");  
            System.exit(0);  
        }  
        try  
        {  
            String namaServer = args[0];  
            String portStr = args[1];  
            int port = Integer.parseInt(portStr);  
            SalamKeadilanServer server =  
new SalamKeadilanServer(namaServer, port);  
            server.operateService();  
        }  
        catch(java.io.IOException ioe)  
        {  
            ioe.printStackTrace();  
        }  
    }  
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanServerApp.java

Langkah 3 : Dengan text editor tulislah SalamKeadilanClient.java

```
import java.io.*;  
import java.net.*;  
  
public class SalamKeadilanClient  
{
```

```
private String namaClient;
private String alamatServer;
private int port;

public SalamKeadilanClient(String namaClient, String
alamatServer, int port)
{
    System.out.println("Membuat client dengan nama : " +
namaClient);
    this.namaClient = namaClient;
    this.alamatServer = alamatServer;
    this.port = port;
}

public void connectToServer()
    throws IOException
{
    System.out.println(namaClient + " menghubungi server di
alamat "
                        + alamatServer + " di " + port);
    Socket socket = new Socket(alamatServer, port);

    InputStream inputStream = socket.getInputStream();
    InputStreamReader inputStreamReader = new
InputStreamReader(inputStream);
    BufferedReader bufferedReader = new
BufferedReader(inputStreamReader);

    OutputStream outputStream = socket.getOutputStream();
    PrintWriter printWriter = new PrintWriter(outputStream,
true);

    System.out.println(namaClient + " mengirim data tentang
nama ke server.");
    printWriter.println(namaClient);

    System.out.println(namaClient + " menangkap pesan dari
server : ");
    String pesan = bufferedReader.readLine();

    System.out.println(pesan);

    bufferedReader.close();
    printWriter.close();
    System.out.println(namaClient + " menutup connection.");
    socket.close();
}
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanClient.java

Langkah 4 : Dengan text editor tulishlah SalamKeadilanClientApp.java

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



```
        if(args.length<3)
        {
            System.out.println("Cara penggunaan:
java SalamKeadilanClientApp
<namaClient> <alamatServer> <nomorPortdariServer>");
            System.exit(0);
        }

        try
        {
            String namaClient = args[0];
            String alamatServer = args[1];
            String portStr = args[2];
            int port = Integer.parseInt(portStr);
            SalamKeadilanClient client = new
SalamKeadilanClient
                                (namaClient,
alamatServer, port);
                                client.connectToServer();
        }
        catch(java.io.IOException ioe)
        {
            ioe.printStackTrace();
        }
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai SalamKeadilanClientApp.java

Langkah 5 : Persiapkan dua buah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan dua buah console, dan lakukan change directory di kedua console tsb ke directory yang sudah dipersiapkan.

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

```
$ export CLASSPATH=/home/lab/touch
```

Langkah 6 : Meng-compile SalamKeadilanServerApp.java di console pertama

Melalui console, jalankan javac :

```
$ javac SalamKeadilanServerApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file SalamKeadilanServerApp.class, dan SalamKeadilanServer.class di dalam directory yang sama.

Langkah 6 : Meng-compile SalamKeadilanClientApp.java di console kedua

Melalui console, jalankan javac :

```
$ javac SalamKeadilanClientApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file SalamKeadilanClientApp.class, dan SalamKeadilanClient.class di dalam directory yang sama.

Langkah 7 : Meluncurkan aplikasi SalamKeadilanServerApp di console pertama

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java SalamKeadilanServerApp SalamKeadilanServer 8080
```

Log di console :

Menghidupkan server bernama : SalamKeadilanServer di port : 8080

SalamKeadilanServer menanti connection ...

Langkah 8 : Meluncurkan aplikasi SalamKeadilanClientApp di console kedua

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java SalamKeadilanClientApp ekobs@developerforce.net localhost 8080
```

Log di console :

Membuat client dengan nama : ekobs@developerforce.net

ekobs@developerforce.net menghubungi server di alamat localhost di 8080

Langkah 9 : Perkembangan komunikasi client-server

Saat SalamKeadilanServerApp mendapat connection, akan ditemukan log :

SalamKeadilanServer menangkap connection dari 127.0.0.1

SalamKeadilanServer membaca nama client : ekobs@developerforce.net

SalamKeadilanServer mengirim pesan :

Pesan dari SalamKeadilanServer, untuk ekobs@developerforce.net
di 127.0.0.1 : Salam Keadilan !

Di sisi client, dapat ditemukan log :

ekobs@developerforce.net mengirim data tentang nama ke server.

ekobs@developerforce.net menangkap pesan dari server :

Pesan dari SalamKeadilanServer, untuk ekobs@developerforce.net
di 127.0.0.1 : Salam Keadilan !

Selanjutnya keduanya menutup connection. Di sisi client :

ekobs@developerforce.net menutup connection.

Di sisi server :

SalamKeadilanServer menutup connection.

24. Mengembangkan Web Server Sederhana

Tujuan

Mendayagunakan salah satu kekayaan yang disediakan dalam J2SE yaitu package java.net

Rancangan

Anda akan mengembangkan Web server sederhana dengan Java. Web server ini akan mampu melayani permintaan atas static pages dari Web browser.

Pembekalan

...

Langkah

Langkah 1 : Dengan text editor tulislah PrimitiveWebServer.java

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

public class PrimitiveWebServer
{
    private File docRoot;
    private ServerSocket serverSocket;

    public PrimitiveWebServer(int port, String docRoot)
        throws IOException
    {
        this.docRoot = new File(docRoot);
        if(!this.docRoot.isDirectory())
        {
            throw new IOException(docRoot + " is not a valid directory.");
        }

        System.out.println("Menghidupkan primitive Web server ");
        System.out.println("port : " + port);
        System.out.println("docRoot : " + docRoot);
        serverSocket = new ServerSocket(port);
    }

    public void operateService()
        throws IOException
    {
        while(true)
        {
            System.out.println("Menanti connection ... ");
            Socket socket = serverSocket.accept();
            String alamatClient =
socket.getInetAddress().getHostAddress();
            =
```

```
        System.out.println("Menangkap connection dari " +
alamatClient);

        InputStream inputStream = socket.getInputStream();
        InputStreamReader inputStreamReader =
            new InputStreamReader(inputStream);
        BufferedReader bufferedReader =
            new BufferedReader(inputStreamReader);

        OutputStream outputStream = socket.getOutputStream();

        String barisPertama = bufferedReader.readLine();
        System.out.println("Membaca header : " + barisPertama);

        StringTokenizer st = new StringTokenizer(barisPertama);
        String method = st.nextToken();
        String url = st.nextToken();

        File file = new File(docRoot, url);
        System.out.println("File yang dibaca :
            " + file.getCanonicalPath());

        if(file.exists())
        {
            FileInputStream fis = new FileInputStream(file);

            int sebuahByte = fis.read();
            while(sebuahByte != -1)
            {
                outputStream.write(sebuahByte);
                sebuahByte = fis.read();
            }
        }
        else
        {
            System.out.println("File tak ada");
        }

        bufferedReader.close();

        outputStream.close();
        socket.close();
        System.out.println("Menutup connection.");
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai PrimitiveWebServer.java

Langkah 2 : Dengan teks editor mulai tulis file untuk PrimitiveWebServerApp.java

Ini adalah main class yang ketika dijalankan akan meng-instantiate PrimitiveWebServer dan menjalankannya.

```
public class PrimitiveWebServerApp
{
    public static void main(String[] args)
    {
        if(args.length<2)
        {
            System.out.println("Cara penggunaan:
                                java PrimitiveWebServer
                                <port> <docRoot>");
            System.exit(0);
        }

        try
        {
            String portStr = args[0];
            String docRoot = args[1];
            int port = Integer.parseInt(portStr);
            PrimitiveWebServer server =
                new PrimitiveWebServer(port, docRoot);
            server.operateService();
        }
        catch(java.io.IOException ioe)
        {
            ioe.printStackTrace();
        }
    }
}
```

Simpanlah di directory yang telah Anda persiapkan sebagai PrimitiveWebServerApp.java

Langkah 3 : Persiapkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Lakukan set atas variabel lingkungan yang dibutuhkan, yaitu PATH dan CLASSPATH.

Langkah 4 : Meng-compile

Dengan :

```
$ javac PrimitiveWebServerApp.java
```

Langkah 5 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java PrimitiveWebServerApp 8080 /home/web/developerforce.net
```

Log di console :

Menghidupkan primitive Web server

port : 8080

docRoot : /home/web/developerforce.net

Menanti connection ...

Langkah 6 : Pengujian

Untuk mencoba, Anda dapat mem-browse ke machine Anda pada IP 8080 Cobalah <http://localhost:8080/index.html>

Di console, Anda dapat menemukan log :

```
Menangkap connection dari 127.0.0.1
Membaca header : GET /index.html HTTP/1.0
File yang dibaca : /home/web/developerforce.net/index.html
Menutup connection.
Menanti connection ...
Menangkap connection dari 127.0.0.1
Membaca header : GET /img/import.gif HTTP/1.0
File yang dibaca : /home/web/developerforce.net/img/import.gif
Menutup connection.
Menanti connection ...
Menangkap connection dari 127.0.0.1
Membaca header : GET /img/communityservice.gif HTTP/1.0
File yang dibaca : /home/web/developerforce.net/img/communityservice.gif
Menutup connection.
Menanti connection ...
Menangkap connection dari 127.0.0.1
Membaca header : GET /img/developerforcenet.gif HTTP/1.0
File yang dibaca : /home/web/developerforce.net/img/developerforcenet.gif
Menutup connection.
Menanti connection ...
Menangkap connection dari 127.0.0.1
Membaca header : GET /img/decisivecreativity.gif HTTP/1.0
File yang dibaca : /home/web/developerforce.net/img/decisivecreativity.gif
Menutup connection.
Menanti connection ...
```

Hasilnya :



25. Pengenalan JDBC

Tantangan dalam Mengakses DB

Tersedianya berbagai pilihan vendor di satu sisi membolehkan Anda untuk memilih produk RDBMS sesuai kebutuhan dan keadaan yang dihadapi. Di sisi lain, secara tradisional, ini merupakan tantangan tersendiri berkaitan dengan portabilitas kode program yang Anda kembangkan.

Misalkan Anda mengembangkan program dengan PHP yang semula dikembangkan untuk mengakses MySQL, maka apakah program Anda dapat kemudian dijalankan dengan database berbeda seperti Oracle, DB2 atau Postgress ? Pertanyaan yang serupa dapat ditanyakan untuk aplikasi yang Anda kembangkan dengan Visual Basic yang awalnya mengakses Microsoft SQL Server, atau dengan Borland C++ yang semula mengakses DB2.

Java memecahkan tantangan ini dengan mengenalkan apa yang dinamakan Java Database Connectivity, disingkat JDBC. JDBC adalah sebuah API yang menyediakan antar muka yang standar untuk berinteraksi dengan berbagai database. Program yang Anda kembangkan dengan JDBC untuk berinteraksi dengan MySQL, kini dapat Anda gunakan untuk berinteraksi dengan Postgress, Oracle, DB2, MS Access, ataupun MS SQL Server.

Bagaimana JDBC Bekerja ?

Kata kunci dari feature unggulan ini adalah JDBC driver. JDBC driver adalah driver yang dikembangkan untuk database vendor tertentu yang akan menerjemahkan perintah-perintah yang Anda tulis dalam bahasa Java, dengan JDBC, ke dalam perintah-perintah yang spesifik untuk sebuah database vendor. Dengan cara ini, dapat dikatakan bahwa program Anda dapat berinteraksi dengan database apapun yang tersedia JDBC drivernya.

Terdapat 4 tipe JDBC driver. Perbedaannya terletak pada apakah JDBC driver berhubungan dengan client-side native library yang disediakan oleh database vendor tertentu ataupun melalui protokol jaringan, dan terletak pada apakah JDBC driver ini vendor independent ataupun tidak.

Di mata programmer, perbedaan-perbedaan ini tidak berpengaruh terhadap kode-kode program Anda. Selama Anda mempunyai JDBC driver untuk sebuah database vendor pilihan Anda, Anda dapat mengembangkan program dalam bahasa Java untuk berinteraksi dengannya. Anda boleh tidak peduli dengan tipe berapakah JDBC driver Anda, dan sekali lagi, bahkan secara umum Anda boleh tidak peduli apakah database vendor Anda.

Bekerja dengan JDBC

Secara umum, untuk bekerja dengan JDBC, langkah yang dilakukan adalah :

1. Me-load JDBC Driver ke dalam JVM. JDBC Driver dapat dipandang sebagai library yang dibutuhkan untuk mengakses database tertentu.

2. Membuat koneksi ke DB yang direpresentasikan sebagai obyek `java.sql.Connection`.
3. Membuat obyek `java.sql.Statement` yang akan digunakan untuk mengirimkan perintah SQL ke DB.
4. Menjalankan metoda yang bersesuaian dari obyek `java.sql.Statement`, seperti `executeQuery()` atau `executeUpdate()`.

26. Membuat Connection ke Database

Tujuan

Dalam latihan ini Anda akan mencoba membuat Connection ke database dengan JDBC.

Rancangan

Anda akan membuat Connection ke sebuah database mySQL dengan nama RELIEVEDB.

Pembekalan

Untuk membuat Connection ke database, pertama Anda harus me-load JDBC Driver yang dibutuhkan. Pernyataannya untuk memuatkan JDBC driver ke dalam JVM adalah :

```
Class.forName(jdbcDriver);
```

JDBC driver yang dibutuhkan adalah spesifik untuk masing-masing database vendor. Tabel berikut adalah JDBC Driver yang dibutuhkan untuk mySQL, dan Oracle

DBMS	JDBC Driver
mySQL	org.gjt.mm.mysql.Driver
Oracle	oracle.jdbc.driver.OracleDriver

Lumrahnya JDBC Driver ini disebarluaskan dalam file dengan format .jar. Perlu Anda perhatikan bahwa JDBC driver ini harus ada di dalam CLASSPATH, agar dapat dimuatkan ke dalam JVM.

Selanjutnya, untuk membuat Connection ke DBMS digunakan pernyataan :

```
Connection conn  
= DriverManager.getConnection(url, user, pwd);
```

Parameter yang dibutuhkan adalah URL, nama user dan password. Untuk URL, masing-masing sistem basis data mempunyai pola masing-masing yang spesifik dan khas. Tabel berikut adalah pola dan contoh URL ke beberapa DBMS :

Database	Pola URL	Contoh URL
HyperSQL	jdbc:hsqldb:hsqldb://<hostName>:<port>	jdbc:hsqldb:hsqldb://localhost:7105
mySQL	jdbc:mysql://<hostName>:<port>/<dbName>	jdbc:mysql://localhost:3306/RELIEVEDB
Oracle	jdbc:oracle:thin:@<hostName>:<port>:<dbName>	jdbc:oracle:thin:@localhost:1521:RELIEVEDB

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch
- Anda membutuhkan JDBC Driver. Untuk mySQL, Anda dapat men-download mm.mysql-2.0.8-you-must-unjar-me.jar. Dengan meng-unjar file ini Anda mendapatkan mm.mysql-2.0.8/mm.mysql-2.0.8-bin.jar, yang memuat JDBC Driver untuk mySQL

Langkah

Langkah 1 : Luncurkan text editor pilihan Anda

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda. Di Linux

Anda dapat menggunakan kwrite. Di Windows Anda dapat menggunakan Notepad.

Langkah 2 : Menulis kode sumber

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

public class LatJDBCConnectApp
{
    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/RELIEVEDB";
            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 3 : Simpan sebagai LatJDBConnectApp.java

Simpanlah di directory yang telah Anda persiapkan sebagai LatJDBConnectApp.java

Langkah 4 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Langkah 5 : Menge-set variabel lingkungan PATH dan CLASSPATH

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

```
$ export CLASSPATH=/home/lab/touch
```

JDBC driver juga harus tercakup dalam CLASSPATH. Contoh

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

Langkah 6 : Meng-compile

Melalui console, jalankan javac :

```
$ javac LatJDBConnectApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatJDBConnectApp.class di dalam directory yang sama.

Langkah 7 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java LatJDBConnectApp
```

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

Success.

Menutup connection.

Troubleshooting

Kegagalan umum

JDBC Driver tidak ditemukan.

```
java.lang.ClassNotFoundException: org.gjt.mm.mysql.Driver  
    at java.net.URLClassLoader$1.run(URLClassLoader.java:200)  
    at java.security.AccessController.doPrivileged(Native  
Method)  
    at  
java.net.URLClassLoader.findClass(URLClassLoader.java:188)  
    at java.lang.ClassLoader.loadClass(ClassLoader.java:297)  
    at  
sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:286)
```

```
        at java.lang.ClassLoader.loadClass(ClassLoader.java:253)
        at
java.lang.ClassLoader.loadClassInternal(ClassLoader.java:313)
        at java.lang.Class.forName0(Native Method)
        at java.lang.Class.forName(Class.java:120)
        at LatJDBCCConnectApp.main(LatJDBCCConnectApp.java:14)
```

Anda

Solusi : Anda perlu menge-set CLASSPATH agar memuat JDBC driver.

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

Kegagalan authorization

```
java.sql.SQLException: Invalid authorization specification:
Access denied for user:
'ekobs@localhost.localdomain' (Using password: YES)
        at org.gjt.mm.mysql.MySQLIO.init(Unknown Source)
        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:517)
        at
java.sql.DriverManager.getConnection(DriverManager.java:177)
        at LatJDBCCConnectApp.main(LatJDBCCConnectApp.java:23)
```

Solusi adalah : memberikan GRANT ke user ini, root dari mysql harus melakukannya

27. Perintah SQL dengan JDBC : SELECT

Tujuan

Anda akan berinteraksi dengan sistem basis data, menjalankan perintah SQL yaitu SELECT dengan menggunakan JDBC API.

Rancangan

Anda akan mengembangkan aplikasi bernama LatJDBCQueryApp, yang ketika dijalankan akan membaca sebuah table dengan `java.sql.Statement.executeQuery()` dan menampilkannya ke console, dengan `System.out.println()`

Pembekalan

Untuk dapat membaca table dalam sistem basis data dengan menggunakan JDBC API, langkah yang dibutuhkan adalah :

1. Membuat database connection, yang dalam Java berupa obyek `java.sql.Connection`
2. Membuat obyek `java.sql.Statement` untuk mengirimkan perintah SQL ke basis data.
Contoh
3.

```
Statement stmt = conn.createStatement();
```
4. Menjalankan perintah `java.sql.Statement.executeQuery()` dengan melewati parameter berupa perintah SQL. Hasilnya adalah sebuah obyek `java.sql.ResultSet`. Contoh :
5.

```
String selectSQL =
```
6.

```
"SELECT PIN, NAME, IC, ADDRESS FROM PATIENTTBL";
```
7.

```
rs = stmt.executeQuery(selectSQL);
```
8. `java.sql.ResultSet` dapat dipandang sebagai kumpulan row-row dalam table, dan Anda dapat membaca hasil perintah SQL dengan mengakses obyek ini. Contoh :
9.

```
while(rs.next())
```
10.

```
{
```
11.

```
String pin = rs.getString("PIN");
```
12.

```
String name = rs.getString("NAME");
```
13.

```
String ic = rs.getString("IC");
```
14.

```
String address = rs.getString("ADDRESS");
```
15.

```
...
```
16.

```
}
```
- 17.
- 18.

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya `/home/lab/touch`
- Anda membutuhkan JDBC Driver. Untuk MySQL, Anda dapat men-download `mm.mysql-2.0.8-you-must-unjar-me.jar`. Dengan meng-unjar file ini Anda mendapatkan `mm.mysql-2.0.8/mm.mysql-2.0.8-bin.jar`, yang memuat JDBC Driver untuk MySQL

Langkah

Langkah 1 : Luncurkan text editor pilihan Anda

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda. Di Linux Anda dapat menggunakan `kwrite`. Di Windows Anda dapat menggunakan Notepad.

Langkah 2 : Menulis kode sumber

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

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

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

            conn = DriverManager.getConnection(
                url, user, pwd);
            stmt = conn.createStatement();

            String selectSQL =
                "SELECT PIN, NAME, IC, ADDRESS FROM PATIENTTBL";
            rs = stmt.executeQuery(selectSQL);

            while(rs.next())
            {
                String pin = rs.getString("PIN");
                String name = rs.getString("NAME");
                String ic = rs.getString("IC");
                String address = rs.getString("ADDRESS");

                System.out.println();
                System.out.println("P A T I E N T");
                System.out.println("Patient Identification Number
                    : " + pin);
                System.out.println("Name
                    : " + name);
                System.out.println("Identification Card
                    : " + ic);
                System.out.println("Address
                    : " + address);
            }
        }
        catch(ClassNotFoundException cnfe)
        {
            cnfe.printStackTrace();
        }
        catch(SQLException sqle)
        {
            sqle.printStackTrace();
        }
    }
}
```

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

Langkah 3 : Simpan sebagai LatJDBCQueryApp.java

Simpanlah di directory yang telah Anda persiapkan sebagai LatJDBCQueryApp.java

Langkah 4 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Langkah 5 : Menge-set variabel lingkungan PATH dan CLASSPATH

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

```
$ export CLASSPATH=/home/lab/touch
```

JDBC driver juga harus tercakup dalam CLASSPATH. Contoh

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

Langkah 6 : Meng-compile

Melalui console, jalankan javac :

```
$ javac LatJDBCQueryApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatJDBCQueryApp.class di dalam directory yang sama.

Langkah 7 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java LatJDBCQueryApp
```

Hasilnya adalah sebuah pesan dituliskan ke console :

```
P A T I E N T
Patient Identification Number    : M201
```

Name : Muhammad Ilham
Identification Card : KTP 209/8989/876/200969
Address : Jl Cimanggis 3 No 77,
Jakarta Selatan 12000 Indonesia

P A T I E N T

Patient Identification Number : K099
Name : Kwik Tan Tong
Identification Card : Passport S908080D
Address : Block 998L #20-2,
Red Hill 98962 Singapore
[lab@localhost touch]\$

Troubleshooting

```
java.sql.SQLException: Syntax error or access violation:
You have an error in your SQL syntax near '' at line 1
    at org.gjt.mm.mysql.MySQLIO.sendCommand(Unknown Source)
    at org.gjt.mm.mysql.MySQLIO.sqlQueryDirect(Unknown Source)
    at org.gjt.mm.mysql.MySQLIO.sqlQuery(Unknown Source)
    at org.gjt.mm.mysql.Connection.execSQL(Unknown Source)
    at org.gjt.mm.mysql.Connection.execSQL(Unknown Source)
    at org.gjt.mm.mysql.Statement.executeUpdate(Unknown Source)
    at org.gjt.mm.mysql.jdbc2.Statement.executeUpdate(Unknown
Source)
    at LatJDBCUpdateApp.main(LatJDBCUpdateApp.java:48)
```

Anda perlu meneliti kembali perintah SQL yang Anda kirimkan melalui `java.sql.Statement.executeQuery()`

```
java.sql.SQLException: Column 'ID' not found.
```

Anda perlu meneliti kembali perintah SQL yang Anda kirimkan melalui `java.sql.Statement.executeQuery()` maupun pernyataan dalam membaca hasil melalui `java.sql.ResultSet.getXxx()`

28. Perintah SQL dengan JDBC : INSERT, UPDATE, DELETE

Tujuan

Anda akan berinteraksi dengan sistem basis data, menjalankan perintah SQL yaitu INSERT, UPDATE dan DELETE dengan menggunakan JDBC API.

Rancangan

Anda akan mengembangkan aplikasi bernama LatJDBCUpdateApp, yang ketika dijalankan akan mengirimkan perintah SQL ke basis data, yaitu perintah INSERT, UPDATE dan DELETE. Perintah SQL dikirimkan melalui obyek `java.sql.Statement` yaitu metoda `executeUpdate()`. Usai sebuah perintah UPDATE, INSERT atau DELETE dikirimkan, Anda mengirimkan perintah SELECT untuk menguji perubahan yang terjadi dalam table.

Pembekalan

Untuk dapat mengirimkan perintah INSERT, UPDATE dan DELETE table dalam sistem basis data dengan menggunakan JDBC API, langkah yang dibutuhkan adalah tak berbeda jauh dengan mengirimkan perintah SELECT. Perbedaan utamanya adalah bahwa metoda yang digunakan bukan `java.sql.Statement.executeQuery()` melainkan `java.sql.Statement.executeUpdate()`. Hasil metoda ini juga bukan `java.sql.ResultSet` melainkan sebuah bilangan int yang menunjukkan berapa raw dalam table telah terpengaruh oleh perintah SQL yang dikirimkan.

1. Membuat database connection, yang dalam Java berupa obyek `java.sql.Connection`
2. Membuat obyek `java.sql.Statement` untuk mengirimkan perintah SQL ke basis data.
Contoh
3.

```
Statement stmt = conn.createStatement();
```
4. Menjalankan perintah `java.sql.Statement.executeQuery()` dengan melewati parameter berupa perintah SQL. Balikannya adalah sebuah bilangan int yang menunjukkan berapa raw dalam table telah berubah. Contoh :
5.

```
String insertSQL =
```
6.

```
    "INSERT INTO PATIENTTBL VALUES("
```
7.

```
    + "'H089', 'Hariman Siregar', 'KTP 0987.0089.89',
```
8.

```
    'Jl Gunung Salak 87 Bandung 43719 Indonesia')";
```
9.

```
int jumlahTerinsert = stmt.executeUpdate(insertSQL);
```

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya `/home/lab/touch`
- Anda membutuhkan JDBC Driver. Untuk `mySQL`, Anda dapat men-download `mm.mysql-2.0.8-you-must-unjar-me.jar`. Dengan meng-unjar file ini Anda mendapatkan `mm.mysql-2.0.8/mm.mysql-2.0.8-bin.jar`, yang memuat JDBC Driver untuk `mySQL`

Langkah

Langkah 1 : Luncurkan text editor pilihan Anda

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda. Di Linux Anda dapat menggunakan `kwrite`. Di Windows Anda dapat menggunakan Notepad.

Langkah 2 : Menulis kode sumber

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

public class LatJDBCUpdateApp
{
    public static void main(String[] args)
    {
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        try
        {

            String jdbcDriver
                = "org.gjt.mm.mysql.Driver";
            Class.forName(jdbcDriver);

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

            conn = DriverManager.getConnection(
                url, user, pwd);
            stmt = conn.createStatement();

            String selectSQL =
                "SELECT PIN, NAME, IC, ADDRESS FROM PATIENTTBL";
            rs = stmt.executeQuery(selectSQL);

            while(rs.next())
            {
                System.out.println();
                System.out.println("P A T I E N T");
                System.out.println("Patient Identification Number:
                    " + rs.getString("PIN"));
                System.out.println("Name
                    " + rs.getString("NAME"));
                System.out.println("Identification Card
                    " + rs.getString("IC"));
                System.out.println("Address
                    " + rs.getString("ADDRESS"));
            }

            /*
            INSERT
            */

            String insertSQL =
                "INSERT INTO PATIENTTBL VALUES("
                + "'H089', 'Hariman Siregar', 'KTP 0987.0089.89',
                    'Jl Gunung Salak 87
                    Bandung 43719 Indonesia')";
            System.out.println();
            System.out.println(insertSQL);
            int jumlahTerinsert = stmt.executeUpdate(insertSQL);
```

```
System.out.println("Jumlah Ter-insert :  
" + jumlahTerinsert );  
  
rs = stmt.executeQuery(selectSQL);  
  
while(rs.next())  
{  
    System.out.println();  
    System.out.println("P A T I E N T");  
    System.out.println("Patient Identification Number:  
" + rs.getString("PIN"));  
    System.out.println("Name :  
" + rs.getString("NAME"));  
    System.out.println("Identification Card :  
" + rs.getString("IC"));  
    System.out.println("Address :  
" + rs.getString("ADDRESS"));  
  
}  
  
/*  
UPDATE  
*/  
  
String updateSQL =  
    "UPDATE PATIENTTBL "  
    + "SET IC= 'SIM 978123.65621', ADDRESS=  
        'Jl Sungai Citarum X No 19  
        Jakarta 13445 Indonesia' "  
    + "WHERE PIN = 'H089'";  
System.out.println();  
System.out.println(updateSQL);  
int jumlahTerupdate = stmt.executeUpdate(updateSQL);  
  
System.out.println("Jumlah Ter-update :  
" + jumlahTerupdate );  
  
rs = stmt.executeQuery(selectSQL);  
  
while(rs.next())  
{  
    System.out.println();  
    System.out.println("P A T I E N T");  
    System.out.println("Patient Identification Number:  
" + rs.getString("PIN"));  
    System.out.println("Name :  
" + rs.getString("NAME"));  
    System.out.println("Identification Card :  
" + rs.getString("IC"));  
    System.out.println("Address :  
" + rs.getString("ADDRESS"));  
  
}  
  
/*  
DELETE  
*/
```

```
String deleteSQL =
    "DELETE FROM PATIENTTBL WHERE PIN='H089'";
System.out.println();
System.out.println(deleteSQL);
int jumlahTerdelete = stmt.executeUpdate(deleteSQL);

System.out.println("Jumlah Ter-update : " + jumlahTerupdate
);

rs = stmt.executeQuery(selectSQL);

while(rs.next())
{
    System.out.println();
    System.out.println("P A T I E N T");
    System.out.println("Patient Identification Number:
        " + rs.getString("PIN"));
    System.out.println("Name
        " + rs.getString("NAME"));
    System.out.println("Identification Card
        " + rs.getString("IC"));
    System.out.println("Address
        " + rs.getString("ADDRESS"));
}
}
catch(ClassNotFoundException cnfe)
{
    cnfe.printStackTrace();
}
catch(SQLException sqle)
{
    sqle.printStackTrace();
}
finally
{
    try
    {
        if(rs != null) rs.close();
        if(stmt != null) stmt.close();
        if(conn != null) conn.close();
    }
    catch(SQLException sqle)
    {
        sqle.printStackTrace();
    }
}
}
```

Langkah 3 : Simpan sebagai LatJDBCUpdateApp.java

Simpanlah di directory yang telah Anda persiapkan sebagai LatJDBCUpdateApp.java

Langkah 4 : Luncurkan sebuah console

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan change directory ke directory yang sudah dipersiapkan.

Langkah 5 : Menge-set variabel lingkungan PATH dan CLASSPATH

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan PATH harus memuat directory dimana perintah java dan javac. Contoh untuk menge-set variabel lingkungan PATH :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

```
$ export CLASSPATH=/home/lab/touch
```

JDBC driver juga harus tercakup dalam CLASSPATH. Contoh

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

Langkah 6 : Meng-compile

Melalui console, jalankan javac :

```
$ javac LatJDBCUpdateApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatJDBCUpdateApp.class di dalam directory yang sama.

Langkah 7 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java LatJDBCUpdateApp
```

Hasilnya adalah sebuah pesan dituliskan ke console :

P A T I E N T

```
Patient Identification Number : M201
Name : Muhammad Ilham
Identification Card : KTP 209/8989/876/200969
Address : Jl Cimanggis 3 No 77,
Jakarta Selatan 12000 Indonesia
```

P A T I E N T

```
Patient Identification Number : K099
Name : Kwik Tan Tong
Identification Card : Passport S908080D
Address : Block 998L #20-2,
Red Hill 98962 Singapore
```

```
INSERT INTO PATIENTTBL VALUES('H089', 'Hariman Siregar',
'KTP 0987.0089.89', 'Jl Gunung Salak 87 Bandung 43719 Indonesia')
Jumlah Ter-insert : 1
```

P A T I E N T

```
Patient Identification Number : M201
Name : Muhammad Ilham
Identification Card : KTP 209/8989/876/200969
Address : Jl Cimanggis 3 No 77,
Jakarta Selatan 12000 Indonesia
```

P A T I E N T

Patient Identification Number : K099
Name : Kwik Tan Tong
Identification Card : Passport S908080D
Address : Block 998L #20-2,
Red Hill 98962 Singapore

P A T I E N T

Patient Identification Number : H089
Name : Hariman Siregar
Identification Card : KTP 0987.0089.89
Address : Jl Gunung Salak 87
Bandung 43719 Indonesia

UPDATE PATIENTTBL SET IC= 'SIM 978123.65621',
ADDRESS='Jl Sungai Citarum X No 19 Jakarta 13445
Indonesia' WHERE PIN = 'H089' Jumlah Ter-update : 1

P A T I E N T

Patient Identification Number : M201
Name : Muhammad Ilham
Identification Card : KTP 209/8989/876/200969
Address : Jl Cimanggis 3 No 77,
Jakarta Selatan 12000 Indonesia

P A T I E N T

Patient Identification Number : K099
Name : Kwik Tan Tong
Identification Card : Passport S908080D
Address : Block 998L #20-2,
Red Hill 98962 Singapore

P A T I E N T

Patient Identification Number : H089
Name : Hariman Siregar
Identification Card : SIM 978123.65621
Address : Jl Sungai Citarum X No 19
Jakarta 13445 Indonesia

DELETE FROM PATIENTTBL WHERE PIN='H089'
Jumlah Ter-update : 1

P A T I E N T

Patient Identification Number : M201
Name : Muhammad Ilham
Identification Card : KTP 209/8989/876/200969
Address : Jl Cimanggis 3 No 77,
Jakarta Selatan 12000 Indonesia

P A T I E N T

Patient Identification Number : K099
Name : Kwik Tan Tong
Identification Card : Passport S908080D
Address : Block 998L #20-2,
Red Hill 98962 Singapore

29. Akses DB dengan JDBC Melalui Data Access Object

Tujuan

Bergerak lebih lanjut dengan JDBC, Anda akan mencoba mengembangkan sebuah Data Access Object (DAO). DAO dikembangkan sebagai sebuah design pattern yang dimaksudkan untuk memisahkan proses akses db dari proses bisnis lainnya.

Rancangan

Anda akan mengembangkan CatalogueDAO yang akan menggunakan JDBC untuk mengakses table CATALOGUETBL dalam database CATALOGUEDB.

Pembekalan

?

Persiapan

- Anda perlu meng-install terlebih dahulu Java Development Kit (JDK)
- Buatlah sebuah directory untuk latihan Anda, misalnya /home/lab/touch. Dan di dalam directory tsb, buatlah beberapa sub-directory :
 - net/developerforce/amazon/catalogue/app
 - net/developerforce/amazon/catalogue/bean
 - net/developerforce/amazon/catalogue/dao
- Anda membutuhkan JDBC Driver. Untuk mySQL, Anda dapat men-download mm.mysql-2.0.8-you-must-unjar-me.jar. Dengan meng-unjar file ini Anda mendapatkan mm.mysql-2.0.8/mm.mysql-2.0.8-bin.jar, yang memuat JDBC Driver untuk mySQL

Langkah

Langkah 1 : Menulis kode sumber CatalogueBean.java

Anda dapat menggunakan sembarang text editor untuk menuliskan kode sumber Anda. Di Linux Anda dapat menggunakan kwrite. Di Windows Anda dapat menggunakan Notepad.

```
package net.developerforce.amazon.catalogue.bean;

import java.util.Date;

public class CatalogueBean
{
    public CatalogueBean(String isbn)
    {
        this.isbn = isbn;
    }

    private String isbn;
    public void setIsbn(String isbn)
```

```
{
    this.isbn = isbn;
}

public String getIsbn()
{
    return this.isbn;
}

private String title;
public void setTitle(String title)
{
    this.title = title;
}

public String getTitle()
{
    return this.title;
}

private String author;
public void setAuthor(String author)
{
    this.author = author;
}

public String getAuthor()
{
    return this.author;
}

private String publisher;
public void setPublisher(String publisher)
{
    this.publisher = publisher;
}

public String getPublisher()
{
    return this.publisher;
}

private String city;
public void setCity(String city)
{
    this.city = city;
}

public String getCity()
{
    return this.city;
}

private String year;
public void setYear(String year)
{
    this.year = year;
}
```



```
public String getYear()
{
    return this.year;
}

private Date lastUpdatedOn;
public void setLastUpdatedOn(Date lastUpdatedOn)
{
    this.lastUpdatedOn = lastUpdatedOn;
}

public Date getLastUpdatedOn()
{
    return this.lastUpdatedOn;
}
}
```

Simpanlah di sub-directory directory net/developerforce/amazon/catalogue/bean yang telah Anda persiapkan sebagai CatalogueBean.java

Langkah 2 : Menulis kode sumber CatalogueDAO.java

```
package net.developerforce.amazon.catalogue.dao;

import java.sql.*;
import java.text.*;
import java.util.Date;
import net.developerforce.amazon.catalogue.bean.CatalogueBean;

public class CatalogueDAO
{
    private Connection conn;
    public CatalogueDAO(Connection conn)
        throws NullPointerException
    {
        if(conn==null)
        {
            throw new NullPointerException();
        }

        this.conn = conn;
    }

    public CatalogueBean selectCatalogueBean(String isbn)
        throws SQLException
    {
        CatalogueBean bean = null;

        String sqlSelect =
            "SELECT * FROM CATALOGUETBL "
            + "WHERE ISBN = '" + isbn + "'";
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(sqlSelect);

        if(rs.next())
        {
            bean = new CatalogueBean(isbn);
        }
    }
}
```

```
        bean.setTitle(rs.getString("TITLE"));
        bean.setAuthor(rs.getString("AUTHOR"));
        bean.setPublisher(rs.getString("PUBLISHER"));
        bean.setCity(rs.getString("CITY"));
        bean.setYear(rs.getString("YEAR"));

        String lastUpdatedOn = rs.getString("LASTUPDATEDON");
        SimpleDateFormat format = new SimpleDateFormat("yyyyMMdd");
        try
        {
            Date date = format.parse(lastUpdatedOn);
            bean.setLastUpdatedOn(date);
        }
        catch(ParseException pe)
        {
            pe.printStackTrace();
        }
    }
    return bean;
}

public int insertCatalogueBean(CatalogueBean bean)
    throws SQLException
{
    String lastUpdatedOn = null;
    Date date = bean.getLastUpdatedOn();
    if(date != null)
    {
        SimpleDateFormat format = new SimpleDateFormat("yyyyMMdd");
        lastUpdatedOn = format.format(date);
    }

    Statement stmt = conn.createStatement();
    String sqlInsert =
        "INSERT INTO CATALOGUETBL "
        + "VALUES ("
        + "'" + bean.getIsbn() + "',"
        + "'" + bean.getTitle() + "',"
        + "'" + bean.getAuthor() + "',"
        + "'" + bean.getPublisher() + "',"
        + "'" + bean.getCity() + "',"
        + "'" + bean.getYear() + "',"
        + "'" + lastUpdatedOn + "'"
        + ")"
        ;
    return stmt.executeUpdate(sqlInsert);
}

public int updateCatalogueBean(CatalogueBean bean)
    throws SQLException
{
    String lastUpdatedOn = null;
    Date date = bean.getLastUpdatedOn();
    if(date != null)
    {
        SimpleDateFormat format = new SimpleDateFormat("yyyyMMdd");
```

```
        lastUpdatedOn = format.format(date);
    }

    String sqlUpdate =
        "UPDATE CATALOGUETBL "
        + "SET "
        + "TITLE = '" + bean.getTitle() + "',"
        + "AUTHOR = '" + bean.getAuthor() + "',"
        + "PUBLISHER = '" + bean.getPublisher() + "',"
        + "CITY = '" + bean.getCity() + "',"
        + "YEAR = '" + bean.getYear() + "',"
        + "LASTUPDATEDON = '" + lastUpdatedOn + "'"
        + "WHERE "
        + "ISBN = '" + bean.getIsbn() + "'";

    Statement stmt = conn.createStatement();
    return stmt.executeUpdate(sqlUpdate);
}

public int deleteCatalogueBean(String isbn)
    throws SQLException
{
    String sqlDelete =
        "DELETE FROM CATALOGUETBL "
        + "WHERE "
        + "ISBN = '" + isbn + "'";
    Statement stmt = conn.createStatement();
    return stmt.executeUpdate(sqlDelete);
}
}
```

Simpanlah di sub-directory `net/developerforce/amazon/catalogue/dao` yang telah Anda persiapkan sebagai `CatalogueDAO.java`

Langkah 3 : Menulis kode sumber `CatalogueDAOTestApp.java`

```
package net.developerforce.amazon.catalogue.app;

import java.sql.*;
import java.util.Date;
import net.developerforce.amazon.catalogue.bean.CatalogueBean;
import net.developerforce.amazon.catalogue.dao.CatalogueDAO;

public class CatalogueDAOTestApp
{
    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/AMAZONDB";
            String user = "ekobs";
            String pwd = "j2ee";

            conn
```

```
        = DriverManager.getConnection(
            url, user, pwd);

    CatalogueDAOTestApp app = new CatalogueDAOTestApp(conn);

    app.create();
    app.view();
    app.modify();
    app.view();
    app.remove();
}
catch(ClassNotFoundException cnfe)
{
    cnfe.printStackTrace();
}
catch(SQLException sqle)
{
    sqle.printStackTrace();
}
catch(NullPointerException npe)
{
    npe.printStackTrace();
}
finally
{
    try
    {
        if(conn != null)conn.close();
    }
    catch(SQLException sqle)
    {
        sqle.printStackTrace();
    }
}
}

private CatalogueDAO dao;

public CatalogueDAOTestApp(Connection conn)
    throws NullPointerException
{
    dao = new CatalogueDAO(conn);
}

public void create()
    throws SQLException
{
    System.out.println("Creating ...");
    String isbn = "0-201-47967-2";
    CatalogueBean bean = new CatalogueBean(isbn);
    bean.setTitle("Customer-centered Growth:
        5 Proven Strategies For Building Competitive Advantage");
    bean.setAuthor("Richard Whiteley");
    bean.setPublisher("Addison Wesley Publishing Company");
    bean.setCity("Massachusetts");
    bean.setYear("1996");
    bean.setLastUpdatedOn(new Date());
}
```

```
        dao.insertCatalogueBean(bean);
    }

    public void modify()
        throws SQLException
    {
        System.out.println("Modifying ...");
        String isbn = "0-201-47967-2";
        CatalogueBean bean = dao.selectCatalogueBean(isbn);
        bean.setCity("Reading, Massachusetts");

        dao.updateCatalogueBean(bean);
    }

    public void remove()
        throws SQLException
    {
        System.out.println("Removing ...");
        String isbn = "0-201-47967-2";
        dao.deleteCatalogueBean(isbn);
    }

    public void view()
        throws SQLException, NullPointerException
    {
        System.out.println("Displaying ...");
        String isbn = "0-201-47967-2";
        CatalogueBean bean = dao.selectCatalogueBean(isbn);

        System.out.println("ISBN           : " + bean.getIsbn());
        System.out.println("Title           : " + bean.getTitle());
        System.out.println("Author          : " + bean.getAuthor());
        System.out.println("Publisher       : " + bean.getPublisher());
        System.out.println("City            : " + bean.getCity());
        System.out.println("Year            : " + bean.getYear());
        System.out.println("Last Updated On : " + bean.getLastUpdatedOn());
    }
}
```

Simpanlah di sub-directory `net/developforce/amazon/catalogue/app` yang telah Anda persiapkan sebagai `CatalogueDAOTestApp.java`

Langkah 4 : Persiapkan sebuah console untuk meng-compile dan meluncurkan aplikasi

Anda akan meng-compile dan meluncurkan aplikasi dari console. Luncurkan sebuah console, dan lakukan `change directory` ke directory yang sudah dipersiapkan.

```
$ cd /home/lab/touch/
```

Untuk dapat sukses melakukan kompilasi dan meluncurkan aplikasi, Anda harus menge-set setidaknya dua buah variabel lingkungan dalam Operating System Anda, yaitu : Variabel lingkungan `PATH` harus memuat directory dimana perintah `java` dan `javac`. Contoh untuk menge-set variabel lingkungan `PATH` :

```
$ export PATH=/home/lab/jdk1.3.1_01/bin:$PATH
```

Variabel lingkungan CLASSPATH harus memuat directory di mana file .class dari aplikasi Anda berada. Contoh untuk menge-set variabel lingkungan CLASSPATH :

```
$ export CLASSPATH=/home/lab/touch
```

JDBC driver juga harus tercakup dalam CLASSPATH. Contoh

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

Langkah 5 : Meng-compile

Melalui console, jalankan javac :

```
$ javac -sourcepath /home/lab/touch net/developerforce/amazon/catalogue/app/CatalogueDAOTestApp.java
```

Jika Anda menjalani langkah-langkah dengan benar, Anda dapat menemukan file LatJDBCQueryApp.class di dalam directory yang sama.

Langkah 6 : Meluncurkan aplikasi

Untuk meluncurkan aplikasi Anda, melalui console, jalankan java :

```
$ java net/developerforce/amazon/catalogue/app/CatalogueDAOTestApp
```

Hasilnya adalah sebuah pesan dituliskan ke console :

```
Creating ...
Displaying ...
ISBN          : 0-201-47967-2
Title         : Customer-centered Growth:
5 Proven Strategies For Building Competitive Advantage
Author        : Richard Whiteley
Publisher     : Addison Wesley Publishing Company
City          : Massachusetts
Year          : 1996
Last Updated On : Sat Aug 17 00:00:00 SGT 2002
Modifying ...
Displaying ...
ISBN          : 0-201-47967-2
Title         : Customer-centered Growth:
5 Proven Strategies For Building Competitive Advantage
Author        : Richard Whiteley
Publisher     : Addison Wesley Publishing Company
City          : Reading, Massachusetts
Year          : 1996
Last Updated On : Sat Aug 17 00:00:00 SGT 2002
Removing ...
```

Biografi dan Profil



Eko Budhi Suprasetiawan. Lahir 22 Juli 1975 di Jakarta. Menyelesaikan S1 di Institut Teknologi Bandung (ITB) pada tahun 1998. Pendiri dan pengelola situs DeveloperForce.Net. Minat dan bidang keahlian adalah pada J2EE, mySQL, dan Linux. Aktif dalam berbagai organisasi kemahasiswaan selama di ITB, termasuk didalamnya: Pusat Studi Ilmu Kemasyarakatan, Pusat Teknologi Tepat Guna Masjid Salman, Pusat Ilmu Komputer dan Sistem Informasi ITB, Jaringan Informasi Islam, dsb. Saat ini juga aktif di organisasi MIFTA

Berpengalaman kerja di Fujitsu System Indonesia, Sun Microsystems Indonesia, iGine Ptd Ltd. Eko Budhi Suprasetiawan adalah pengelola milis: jlinux@yahoogroups.com.

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:

Email: ekobs@developerforce.net