

# Akses Database melalui Torque

**Eko Budhi Suprasetiawan**

ekobs@developerforce.net

## ***Lisensi Dokumen:***

*Copyright © 2003 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.*

*(\*) Kerjasama IlmuKomputer.Com dan DeveloperForce.Net*

## Motivasi

Torque bisa meng-generate Java Bean sesuai definisi dalam file XML. Dan juga meng-generate JDBC code. Bahkan Torque bisa digunakan untuk meng-create database dan table.

## Persiapan

Persiapkan direktori kerja, dalam contoh ini /home/lab/myjava

## Dapatkan Torque library

di-extract ke /home/lab/torque-3.1

## serta Torque generator

di-extract ke /home/lab/torque-gen-3.1

## Deskripsi

Sebuah Java Bean akan disimpan/dibaca dari table EMPLOYEE\_TBL di database. myproject-schema.xml digunakan untuk memetakan antara property dari EmployeeBean dan column di EMPLOYEE\_TBL. database.xml digunakan sebagai database configuration.

## Langkah

1. Copy dan edit build-torque.xml
2. Tulis build.properties
3. Tulis myproject-schema.xml
4. Jalankan Ant

5. Tulis Torque.properties
6. Tulis EmployeeApp
7. Set CLASSPATH
8. Compile
9. Launch

### **build-torque.xml**

Agar file-file yg di-generate Torque diletakkan di /home/lab/myjava dan bukan di direktori instalasi Torque.

Copy ke /home/lab/myjava

```
$ cp /home/lab/torque-gen-3.1/build-torque.xml /home/lab/myjava
```

lalu edit dari :

```
<property name="lib.dir" value="lib"/>
```

menjadi :

```
<property name="lib.dir" value="/home/lab/torque-gen-3.1/lib"/>
```

## Generator Configuration

Simpan di /home/lab/myjava

# build.properties

```
torque.home      = .  
torque.templatePath = /home/lab/torque-gen-3.1/templates  
torque.output.dir = .  
torque.schema.dir = ./schema  
  
torque.project  = myproject  
torque.database = mysql  
torque.targetPackage = employee
```

## Mapping

Simpan di sub-directory schema di bawah /home/lab/myjava

# myproject-schema.xml

```
<database name="mydatabase" defaultIdMethod="none">  
  <table name="EMPLOYEE_TBL">  
    <column name="ID" required="true" type="VARCHAR"/>  
    <column name="NAME" type="VARCHAR"/>  
    <column name="DEPARTMENT" type="VARCHAR"/>  
    <column name="JOB_TITLE" type="VARCHAR"/>  
    <column name="HIRE_DATE" type="DATE"/>  
    <column name="PERMANENT_EMPLOYEE" type="BOOLEANINT"/>  
    <column name="SALARY" type="DOUBLE"/>  
  </table>  
</database>
```

## Ant

### Jalankan Ant !

```
$ ant -f build-torque.xml  
Buildfile: build-torque.xml
```

```
main:
```

```
check-use-classpath:
```

```
check-run-only-on-schema-change:
```

```
sql-check:
```

```
sql:
```

```
    [echo] +-----+  
    [echo] |  
    [echo] |   Generating SQL for YOUR Torque project!   |  
    [echo] |  
    [echo] +-----+
```

```
sql-classpath:
```

```
sql-template:
```

```
    [echo] loading templates from /home/lab/torque-gen-3.1/templates  
[torque-sql] Using contextProperties file:  
/home/lab/myjava/build.properties  
[torque-sql] Generating to file  
/home/lab/myjava/sql/report.myproject.sql.generation  
[torque-sql] log4j:ERROR Could not find value for key  
log4j.appender.org.apache.torque.engine  
[torque-sql] log4j:ERROR Could not instantiate appender named  
"org.apache.torque.engine".  
[torque-sql] 2003-11-12 21:11:12,903 [main] INFO  
org.apache.torque.engine.database.transform.XmlToAppData - Parsing file:  
'myproject-schema.xml'
```

```
check-use-classpath:
```

```
check-run-only-on-schema-change:
```

```
om-check:
```

```
om:
```

```
    [echo] +-----+  
    [echo] |  
    [echo] |   Generating Peer-based Object Model for   |  
    [echo] |   YOUR Torque project!                   |  
    [echo] |  
    [echo] +-----+
```

```
om-classpath:
```

```
om-template:
```

```
[torque-data-model] Using contextProperties file:  
/home/lab/myjava/build.properties  
[torque-data-model] Generating to file  
/home/lab/myjava/java/report.myproject.om.generation  
[torque-data-model] log4j:ERROR Could not find value for key  
log4j.appender.org.apache.torque.engine  
[torque-data-model] log4j:ERROR Could not instantiate appender named  
"org.apache.torque.engine".  
[torque-data-model] 2003-11-12 21:11:17,155 [main] INFO  
org.apache.torque.engine.database.transform.XmlToAppData - Parsing  
file: 'myproject-schema.xml'
```

```
BUILD SUCCESSFUL  
Total time: 13 seconds
```

## Client App

Simpan di sub-directory app di bawah /home/lab/myjava/java

# app/EmployeeApp.java

```
package app;

import java.text.SimpleDateFormat;
import java.util.*;

import org.apache.torque.*;
import org.apache.torque.util.Criteria;

import employee.*;

public class EmployeeApp
{
    public static void main(String[] args)
    {
        try
        {
            Torque.init("Torque.properties");

            EmployeeTbl bean = new EmployeeTbl();
            bean.setId("032");
            bean.setName("Muhammad Sigit Arifianto");
            bean.setDepartment("Business Development");
            bean.setJobTitle("Account Manager");
            bean.setHireDate(new Date());
            bean.setPermanentEmployee(true);
            bean.setSalary(5000);

            System.out.println("E M P L O Y E E --- BEFORE STORE");
            System.out.println("Id : " + bean.getId());
            System.out.println("Name : " + bean.getName());
            System.out.println
                ("Department : " + bean.getDepartment());
            SimpleDateFormat format
                = new SimpleDateFormat("dd/MMMM/yyyy");
            System.out.println
                ("Hire Date : "
                 + format.format(bean.getHireDate()));
            System.out.println
                ("Permanent Emp.: "
                 + bean.getPermanentEmployee());
            System.out.println("Salary : " + bean.getSalary());
            System.out.println();

            bean.save();

            Criteria criteria = new Criteria();
            criteria.add(EmployeeTblPeer.ID, "032");
```

```
        List beans = EmployeeTblPeer.doSelect(criteria);
        bean = (EmployeeTbl) beans.get(0);

        System.out.println("E M P L O Y E E --- AFTER LOAD");
        System.out.println("Id : " + bean.getId());
        System.out.println("Name : " + bean.getName());
        System.out.println
            ("Department : " + bean.getDepartment());
        System.out.println
            ("Hire Date : "
             + format.format(bean.getHireDate()));
        System.out.println
            ("Permanent Emp.: "
             + bean.getPermanentEmployee());
        System.out.println("Salary : " + bean.getSalary());
        System.out.println();
    }
    catch(TorqueException e)
    {
        e.printStackTrace();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}

}
```

## Run-time Configuration

Simpan di /home/lab/myjava

# Torque.properties

```
log4j.rootCategory = DEBUG, default
log4j.appender.default = org.apache.log4j.FileAppender
log4j.appender.default.file = ./torque.log
log4j.appender.default.layout = org.apache.log4j.SimpleLayout
```

```
torque.database.default = mydatabase
torque.database.mydatabase.driver = org.gjt.mm.mysql.Driver
torque.database.mydatabase.url = jdbc:mysql://127.0.0.1/MYAPP_DB
torque.database.mydatabase.username = ekobs
torque.database.mydatabase.password = j2ee
```

```
torque.database.mydatabase.adapter=mysql
```

```
torque.dsfactory.mydatabase.factory=\
  org.apache.torque.dsfactory.TorqueDataSourceFactory
torque.dsfactory.mydatabase.pool.defaultMaxActive=10
torque.dsfactory.mydatabase.pool.testOnBorrow=true
torque.dsfactory.mydatabase.pool.validationQuery=SELECT 1
torque.dsfactory.mydatabase.connection.driver = org.gjt.mm.mysql.Driver
torque.dsfactory.mydatabase.connection.url =
jdbc:mysql://127.0.0.1/MYAPP_DB
torque.dsfactory.mydatabase.connection.user = ekobs
torque.dsfactory.mydatabase.connection.password = j2ee
```

## CLASSPATH

Set CLASSPATH ...

```
$ export CLASSPATH=java
export CLASSPATH=$CLASSPATH:/home/lab/torque-3.1/lib/torque-3.1.jar
$ export CLASSPATH=$CLASSPATH:/home/lab/torque-3.1/lib/village-2.0-dev-
20030825.jar
$ export CLASSPATH=$CLASSPATH:/home/lab/torque-3.1/lib/commons-lang-
1.0.1.jar
$ export CLASSPATH=$CLASSPATH:/home/lab/torque-3.1/lib/commons-logging-
1.0.3.jar

$ export CLASSPATH=$CLASSPATH:/home/lab/torque-3.1/lib/stratum-1.0-b3.jar
$ export CLASSPATH=$CLASSPATH:/home/lab/torque-3.1/lib/commons-
configuration-1.0-dev-3.20030607.194155.jar
$ export CLASSPATH=$CLASSPATH:/home/lab/torque-3.1/lib/commons-beanutils-
1.6.1.jar
$ export CLASSPATH=$CLASSPATH:/home/lab/torque-3.1/lib/commons-dbcp-
20030825.184428.jar
```

```
$ export CLASSPATH=$CLASSPATH:/home/lab/torque-3.1/lib/commons-pool-  
20030825.183949.jar  
$ export CLASSPATH=$CLASSPATH:/home/lab/torque-3.1/lib/commons-  
collections-2.1.jar  
$ export CLASSPATH=$CLASSPATH:/home/lab/mm.mysql-2.0.8/mm.mysql-2.0.8-  
bin.jar
```

## **Compile**

Compile dengan javac ...

```
$ javac -sourcepath java/ -d java/ java/app/EmployeeApp.java
```

## **Launch**

Launch dengan java ...

```
$ java app.EmployeeApp
```