



PEMROGRAMAN FRAMEWORK JAVA

Pertemuan 7

Operasi CRUD

- Operasi CRUD merupakan operasi dasar yang dilakukan menggunakan Database
- Terdiri dari
 - Create: yang berarti membuat (tabel, data)
 - Read: yang berarti membaca data
 - Update: yang berarti mengubah data/tabel
 - Delete: yang berarti menghapus data/tabel



CREATE



READ



UPDATE



DELETE

C

R

U

D

Query Dalam Framework

- Dalam melakukan operasi CRUD, terdapat beberapa cara yang dapat dilakukan:
 - Menggunakan PreparedStatement dari library Java.SQL
 - Menggunakan Query dari JdbcTemplate
- Kedua metode ini dapat digunakan di situasi tertentu, dan memiliki kelebihan serta kekurangan masing-masing

PreparedStatement

- Merupakan statement SQL yang sudah di pra kompilasi dan disimpan dalam bentuk Objek. Memiliki kegunaan untuk REUSE atau penggunaan berulang-ulang
- **PreparedStatement pstmt = con.prepareStatement("UPDATE EMPLOYEES SET SALARY = ? WHERE ID = ?");**
- Yang di mana tanda tanya nanti akan digantikan menggunakan metode berikutnya.

JdbcTemplate

- Merupakan core atau inti dari library query SQL
- Berbeda dengan PreparedStatement, JDBC dapat menggunakan String Query secara langsung dibandingkan dengan PreparedStatement
- Namun JdbcTemplate wajib mengimplementasikan Callback PreparedStatementCreator
- execute(String sql)

PS VS JDBCTemplate

- Dalam PreparedStatement menggunakan placeholder, Oracle hanya mem-parsing SQL sekali, dan menghasilkan hasil sekali. Kemudian cache hasil parse, bersama dengan hasil SQL.
- Dalam JDBCTemplate, Server akan melakukan *parsing* data dengan kekuatan penuh apabila pernyataan SQL terlihat berbeda oleh *parser*

Hibernate dan SQL Server

- Berbeda dengan MySQL/MariaDB/Database Internal, menggunakan database dengan SQL Server memerlukan dependensi yang berbeda
- Dikarenakan SQL Server memerlukan driver yang berbeda, di dalam POM.XML wajib menyertakan dependensi dari Microsoft.
- Agar bisa menggunakan Hibernate sebagai ORM Database, maka di dalam application.properties juga wajib menyertakan definisi untuk DataSource, dan JPA.


```
spring.datasource.driverClassName=com.microsoft.sqlserver.jdbc.SQLServerDriver
spring.datasource.url=jdbc:sqlserver://localhost;databaseName=employees
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql = true

## Hibernate Properties
# The SQL dialect makes Hibernate generate better SQL for the chosen database
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.SQLServer2012Dialect

# Hibernate ddl auto (create, create-drop, validate, update)
spring.jpa.hibernate.ddl-auto = update
```

Spring Data JPA

- Dalam pengaksesan database menggunakan SQL Server maupun DB lainnya dan juga fasilitas ORM dari Hibernate tidak akan lepas dari yang namanya penggunaan Spring Data JPA
- Modul yang disediakan oleh Spring ini berguna untuk meningkatkan kemudahan pengaksesan data dengan aplikasi berbasis Spring
- Spring Data JPA bertujuan untuk meningkatkan implementasi lapisan akses data dengan cara mengurangi usaha yang sebenarnya diperlukan.
- JPA Entity merefleksikan data yang akan diakses

```
@Entity
@Table(name = "employees")
public class Employee {

    private long id;
    private String firstName;
    private String lastName;
    private String emailId;

    public Employee() {

    }

    public Employee(String firstName, String lastName, String emailId) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.emailId = emailId;
    }
}
```

Pengemasan Aplikasi Web

- Berbeda dengan aplikasi web berbasis native (HTML, PHP, JS, dll). Aplikasi yang dibangun dengan framework java memiliki format khusus yang portable dan mudah untuk diimplementasikan di berbagai jenis server
- Aplikasi web Native cukup dengan membawa folder berisi file file html tradisional yang nantinya memerlukan Web Server untuk melakukan pembacaan dokumen HTML tersebut.

Cont'd

- Berbeda dengan menggunakan Framework Java, aplikasi web yang dibuat tidak memerlukan Web Server sama sekali karena sudah disediakan *Embedded Web Server menggunakan Tomcat*.
- Namun dalam pengemasannya terdapat dua jenis file yang wajib diketahui karena memiliki cara penggunaan yang berbeda
- Java ARchive (JAR) dan Web Application Resource (WAR)

JAR File

- Format File ini sangat umum untuk aplikasi apapun berbasis Java, baik Desktop, Mobile bahkan Web dapat menggunakan Format ini.
- File ini berisikan Library, Sumber Daya, dan File Aksesoris lainnya
- File ini dapat dieksekusi dengan menggunakan Java JRE



WAR File

- File ini biasanya digunakan untuk membungkus aplikasi web yang nantinya dapat di deploy atau dipasang ke berbagai Servlet atau container JSP
- WAR biasanya berisikan file HTML, JSP, XML dll

```
1  META-INF/  
2      MANIFEST.MF  
3  WEB-INF/  
4      web.xml  
5      jsp/  
6          helloWorld.jsp  
7      classes/  
8          static/  
9          templates/  
10         application.properties  
11  lib/  
12      // *.jar files as libs
```

Perbedaan

- Perbedaan pertama dan paling jelas adalah ekstensi file. JAR memiliki ekstensi .jar, sedangkan file WAR memiliki ekstensi .war.
- Perbedaan utama kedua adalah tujuan mereka dan cara mereka berfungsi. File JAR memungkinkan kita untuk mengemas beberapa file untuk menggunakannya sebagai pustaka, plugin, atau aplikasi apa pun. Di sisi lain, file WAR hanya digunakan untuk aplikasi web.

Cont'd

- Struktur arsipnya juga berbeda. Kami dapat membuat JAR dengan struktur yang diinginkan. Sebaliknya, WAR memiliki struktur yang telah ditentukan dengan direktori WEB-INF dan META-INF.