

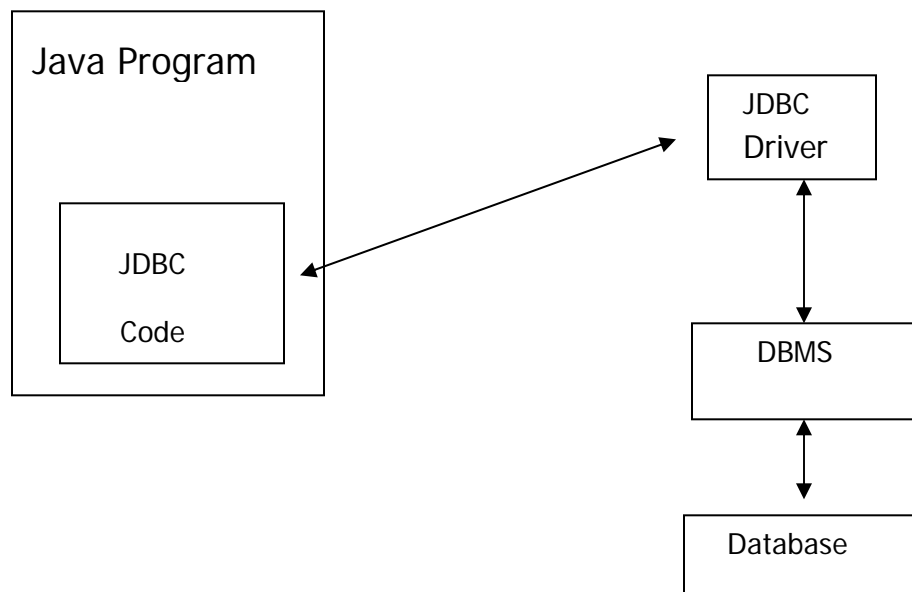
## Bagian 2

### Pengenalan JDBC (1)

#### 1. Pengenalan JDBC

Java Database Connectivity (JDBC) merupakan Application Programming Interface (API), dalam paket `java.sql.*`, yang memungkinkan untuk mengakses segala database menggunakan Java. Dari JDBC ini kemudian muncul konsep-konsep yang berdiri di atas JDBC, seperti SQL-Mapping (misal menggunakan iBatis), Data Access Object (DAO, misal menggunakan SpringDAO), dan Object Relational Mapping (ORM, misal menggunakan Oracle Toplink dan Hibernate).

Posisi JDBC dalam program Java dapat dilihat pada gambar berikut :



#### 2. Akses database dengan Java dan JDBC

Pada dasarnya, untuk mengakses database dengan Java+JDBC, diperlukan langkah-langkah berikut :

- a. Load Database Driver
- b. Buat koneksi
- c. Lakukan pemrosesan query
- d. Tutup koneksi

Berdasarkan langkah di atas maka kita memerlukan Database Driver dari masing-masing vendor database (misal untuk MySQL bisa diunduh dari <http://mysql.com> ) supaya JDBC

bisa mengakses database dari vendor tersebut. Untuk me-load sebuah driver digunakan dynamic class loading, contoh :

```
Class.forName("com.mysql.jdbc.Driver");
```

Syntax di atas digunakan untuk me-load database driver milik MySQL ke dalam system property "jdbc.drivers". Setelah driver di-load, class `javax.sql.DriverManager` dibutuhkan untuk me-load system property "jdbc.drivers". Saat koneksi dibutuhkan, method `getConnection()` dari Driver manager digunakan untuk membuat koneksi JDBC.

```
Connection con = DriverManager.getConnection(
    "jdbc:mysql://host:port/dbname",
    "dbuser",
    "dbpass" );
```

Setelah koneksi berhasil terbentuk, maka bisa dilakukan interaksi dengan database melalui perintah-perintah SQL, dengan terlebih dahulu membuat objek `Statement` sebagai berikut :

```
. . . . .
Statement stmt = con.createStatement();
ResultSet rset = stmt.executeQuery(SQLQuery);
. . . . .
```

`SQLQuery` merupakan suatu `String` yang berisi query ke database. Hasil query tersebut disimpan di objek `ResultSet`. Untuk menampilkan hasil dari query tersebut, digunakan variabel `rset` yang kemudian dimasukkan dalam suatu loop sebagai berikut :

```
. . . . .
while(rset.next()){
    System.out.print(rset.getString(1)+"--");
    System.out.print(rset.getString(2)+"--");
    System.out.print(rset.getString(3));
    System.out.println();
}
. . . . .
```

perintah `while` di atas akan mengambil hasil kolom pertama, kedua dan ketiga. Bila koneksi telah dilakukan dan aplikasi tidak lagi membutuhkan koneksi ke database, session yang terbentuk antara aplikasi dengan database sebaiknya ditutup dengan kode sebagai berikut :

```

try{
    con.close();
}catch(SQLException ex){}

```

### 3. Tugas

1. Load driver ke dalam netbeans.

Klik kanan libraries → pilih Add JAR/Folder → arahkan ke mysql-connector-java.jar

2. Buatlah program berikut (sesuaikan nama database, nama user dan password dengan milik anda...):

```

import java.sql.*;

public class KategoriAccess {

public static void main(String[] args) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
    } catch (ClassNotFoundException cnf) {
        System.out.println(cnf);
    }

    Connection con = null;
    try {
        // buat koneksi. 3306 adalah port default mysql
        con = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/nama_db_anda",
            "user_anda", "password_anda");

        // interaksi dengan database
        Statement stmt = con.createStatement();
        ResultSet rset = stmt.executeQuery("SELECT * FROM
kategori_barang");
        // iterasi untuk mengambil elemen database, kolom pertama
        //diberi index 1

        while (rset.next()) {
            System.out.print(rset.getString(1) + "--");
            System.out.print(rset.getString(2));
            System.out.println();
        }
    } catch (SQLException se) {
        System.out.println(se);
    } finally {
        // tutup koneksi
        try {
            con.close();
        } catch (SQLException ex) {
        }
    }
}
}

```

Jalankan, dan pahami program tersebut.

3. Buatlah kelas `BarangAccess` yang akan menampilkan seluruh data barang yang ada pada tabel barang !