



INSTITUT TEKNOLOGI DEL

MATERI PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

Topik	:	JDBC	
Aktivitas	:	Menulis program	
Tempat Penyerahan	:	Ecourse	
Tujuan	:	Mahasiswa dapat menghubungkan aplikasi Java dengan database.	
Kebutuhan	:	Software or resource	Version required
		NetBeans IDE	Minimum 8.0
		Java Development Kit (JDK)	Minimum 8.0

Buatlah project baru di Netbeans anda dan beri nama **JDBC_NIM**. Selanjutnya ikuti langkah praktikum berikut ini.

I. Pendahuluan

JDBC Java Database Connectivity adalah API yang disediakan Java yang memungkinkan sebuah program berbasis Java untuk berinteraksi dengan media penyimpanan persisten berupa database.

Secara umum, terdapat 4 tipe driver yang dapat digunakan dua diantaranya adalah:

1. JDBC-ODBC: bertindak sebagai bridge antara native driver dan Java program (Tipe 1), dan
2. Pure JDBC: pure Java API yang secara langsung berinteraksi dengan database (Tipe 4).

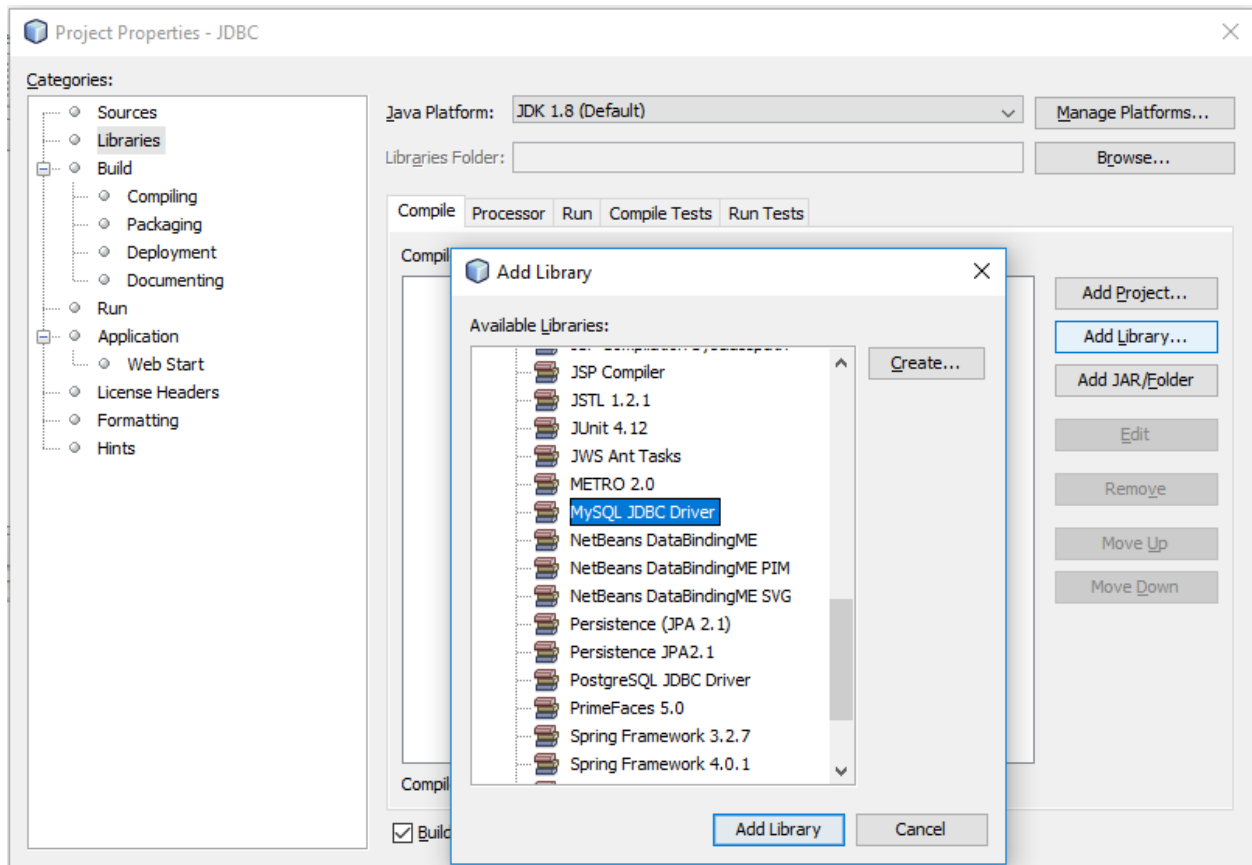
Untuk berinteraksi dengan sebuah basisdata, diperlukan setidaknya lima (5) langkah sbb:

1. Membangun koneksi (didahului dengan me-load driver yang sesuai)
2. Mendesain SQL dan menginstansiasi objek dari kelas Statement.
3. Mengeksekusi statement.
4. Membaca dan mengoperasikan result set hasil eksekusi.
5. Menutup result set, statement & koneksi.

Kali ini anda akan menggunakan JDBC untuk berinteraksi dengan DBMS MySQL (SQL dump tabel customer yang telah tersedia ke database db_jdbc yang Anda buat sendiri).

II. Add MySql driver

Sebelum anda menggunakan driver spesial untuk MySQL, terlebih dahulu tambahkan library MySQL ke dalam project anda, caranya: Klik kanan pada proyek anda > properties > Libraries > Add Library, kemudian tambahkan MySQL JDBC Driver.



Gambar 1 Tambah MySQL JDBC Driver

III. JDBC to MySQL using Statement

Tulislah kelas-kelas berikut.

```
// File name: ConnectToDB.java
public class ConnectToDB {
    private Connection connection=null;
    private Statement stmt=null;
    private ResultSet rs=null;
    public void connect(){
        try{
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            try {
                connection = DriverManager.getConnection("jdbc:mysql:"
                    + "://localhost/db_jdbc?"
                    + "user=username&password=password");
                stmt = connection.createStatement();
                rs = stmt.executeQuery("SELECT * FROM customer");
                while (rs.next()) {
```

```

        System.out.println(rs.getString(3));
    }
    System.out.println("lah");
    rs.close();
    stmt.close();
    connection.close();
} catch (SQLException ex) {
    // handle any errors
    System.out.println("SQLException: " + ex.getMessage());
    System.out.println("SQLState: " + ex.getSQLState());
    System.out.println("VendorError: " + ex.getErrorCode());
}
} catch (ClassNotFoundException cnfe) {
    System.out.println("Terjadi kesalahan: "
        + cnfe.getMessage());
}
catch (Exception ex) {
    ex.printStackTrace();
}
}

```

```

// File name: JDBC.java
public class JDBC {

    public static void main(String[] args) {
        new ConnectToDB().connect();
    }

}

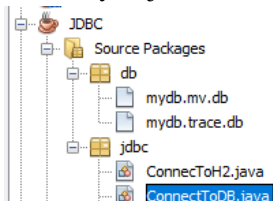
```

IV. JDBC to H2 using Statement

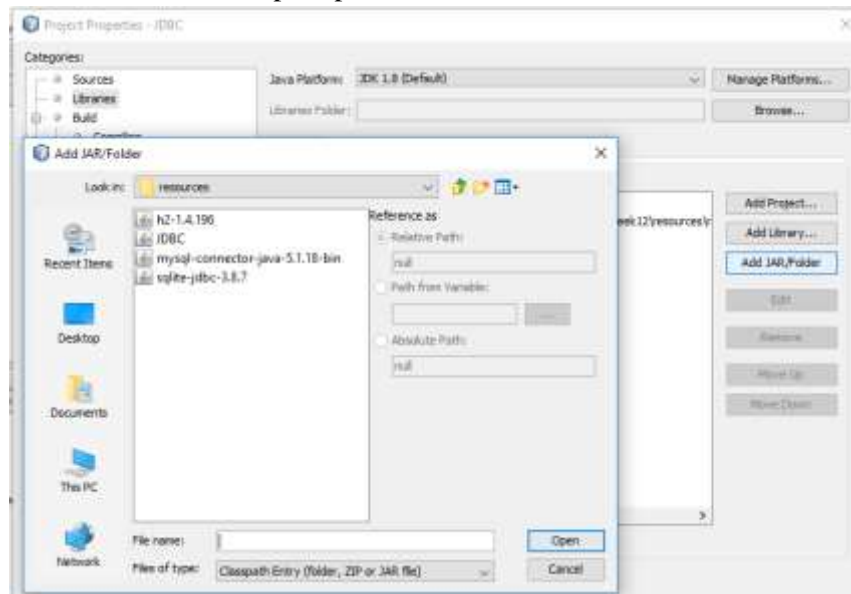
JDBC tidak hanya dapat berkomunikasi dengan database ber DBMS saja seperti MySQL, SQL Server, Oracle dsb, namun juga dengan self-contained database seperti H2 dan SQLite. Hal yang perlu diperhatikan adalah, biasanya self-contained database memiliki fitur yang sangat terbatas, misalnya ketiadaan data integrity check.

Langkah-langkahnya adalah sebagai berikut:

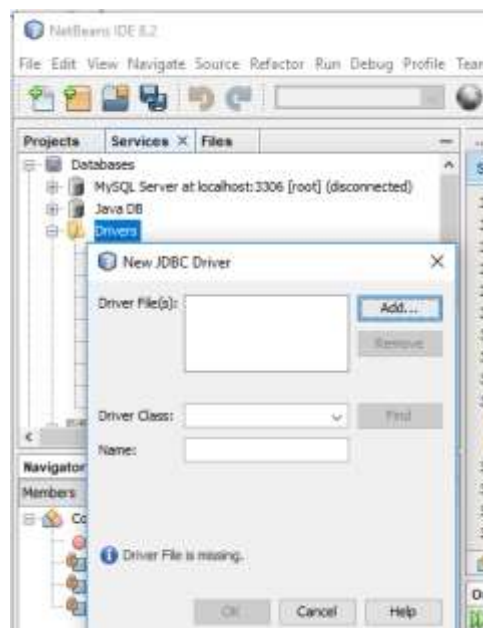
1. Tambahkan library h2 (file h2-1.4.196.jar) pada direktori proyek anda. Lihat Gambar 2.
2. Dari tab service, tambahkan driver h2 (file h2-1.4.196.jar). Lihat Gambar 3.
3. Klik kanan H2 Database Engine → Connect Using
4. Masukkan username dan password (bebas), dan pada JDBC URL: jdbc:h2:<path db>
contohnya: jdbc:h2:D:\DATA\JDBC\src\db\mydb



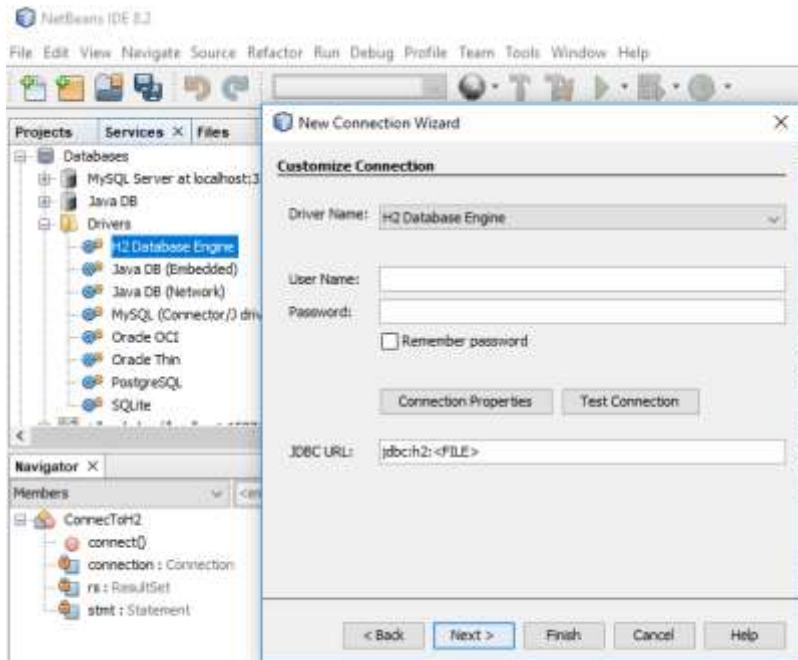
5. Create tabel customer seperti pada Gambar 5.
6. Tambahkan data ke dalam tabel seperti pada Gambar 6 dan 7.



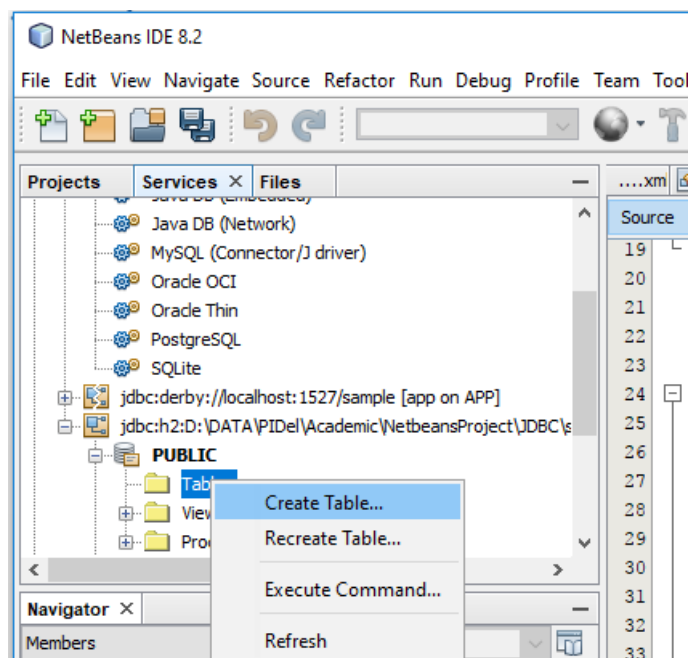
Gambar 2 Tambah library H2



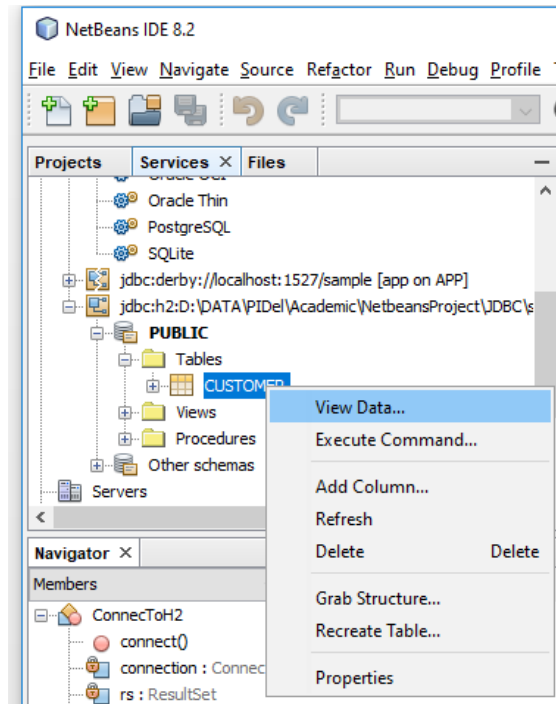
Gambar 3 Tambah driver H2



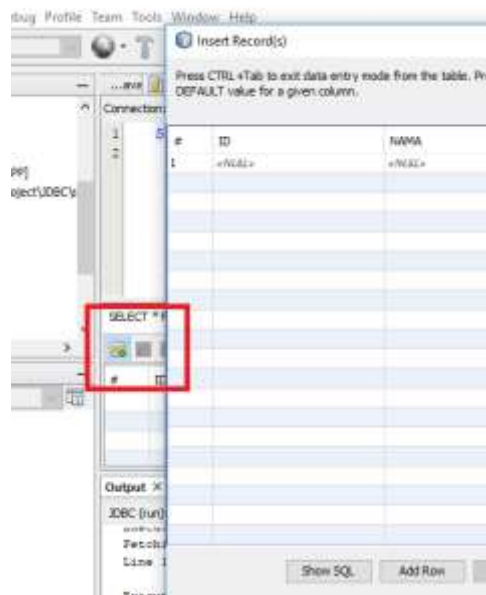
Gambar 4 Create database



Gambar 5 Create tabel



Gambar 6 Tambah data



Gambar 7 Tambah data

7. Klik kanan H2 Database Engine→Disconnect
8. Selanjutnya buatlah kelas-kelas berikut.

```
//File name ConnecToH2.java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
public class ConnecToH2{
    public void connect() {
        try {
            Class.forName("org.h2.Driver");
            try {
                connection =
                DriverManager.getConnection("jdbc:h2:./src/db/mydb","username","password");
                Statement stmt = connection.createStatement();
                ResultSet rs = stmt.executeQuery("SELECT * FROM customer");
                while (rs.next()) {
                    System.out.println(rs.getString(3));
                }
                rs.close();
                stmt.close();
                connection.close();
            } catch (SQLException ex) {
                // handle any errors
                System.out.println("SQLException: " + ex.getMessage());
                System.out.println("SQLState: " + ex.getSQLState());
                System.out.println("VendorError: " + ex.getErrorCode());
            }
            catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    }
}
```

```
// File name: JDBC.java
public class JDBC {

    public static void main(String[] args) {
        new ConnecToDB().connect();
        new ConnecToH2 ().connect();
    }

}
```

V. Tugas

Buatlah GUI untuk menampilkan dan CRUD tabel customer. Beri nama proyek anda Customer_App_NIM. Gunakan embedded database H2.