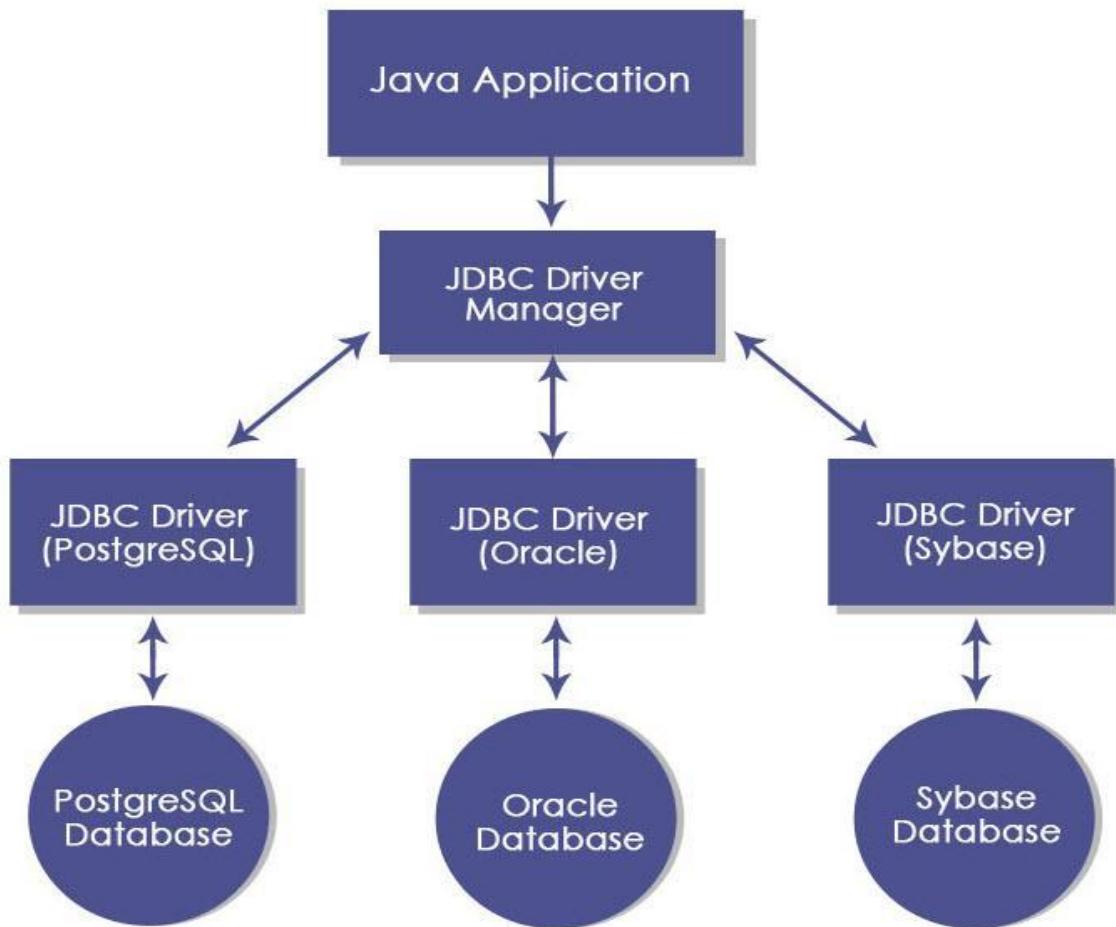


Pengertian JDBC

Jdbc atau java database connectivity adalah API yang menyediakan konektiviti(melakukan komunikasi) antara aplikasi java dengan database relasional. Dengan adanya jdbc, aplikasi java dapat menjalankan perintah sql select, insert, update dan delete.

Sebelum kita masuk kedalam praktek membuat program java yang dapat berinteraksi dengan database, alangkah baiknya kita memahami arsitektur dari JDBC. Perhatikan diagram dibawah ini.



Berdasarkan diagram diatas, berikut penjelasannya

1. Java application : Aplikasi java yang menyimpan hasil pengolahan datanya didalam database. Aplikasi tersebut bisa berupa aplikasi web, aplikasi dekstop maupun webservices.
2. Driver Manager : Kelas yang meload jdbc driver.
3. Driver : Software vendor yang mengimplementasi JDBC API(driver database mysql dengan driver database yang lain nya akan berbeda).
4. Database : Media penyimpanan data (Mysql, Oracle, postgresSQL Sysbase dll).

Pemrograman Database Java dan Database Mysql

1. Syarat yang harus dipenuhi sebelum mengikuti tutorial

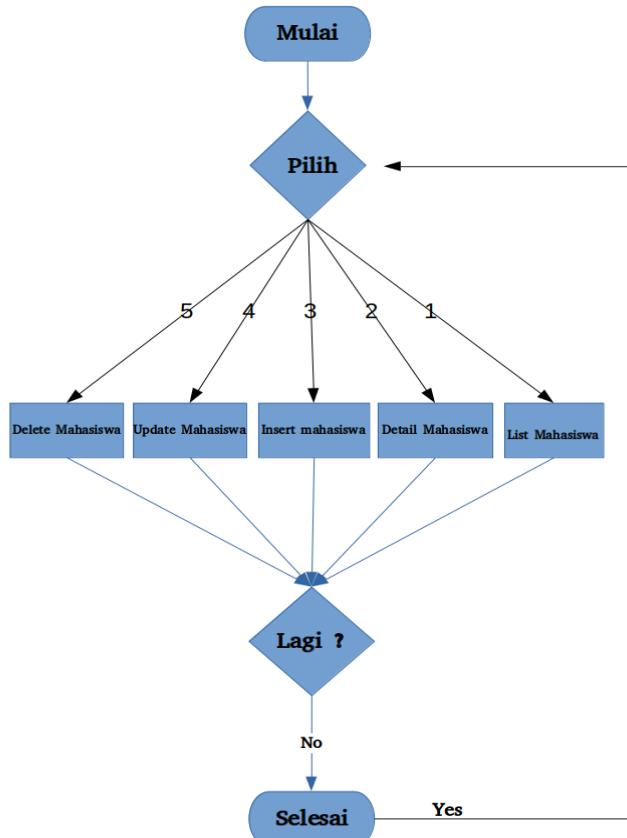
Dalam mengikuti praktek pemrograman database yang menggunakan jdbc, pembaca diharapkan sudah punya **IntelliJ IDEA** yang terinstall di laptop. IntelliJ IDEA yang saya gunakan **IntelliJ IDEA versi community**, database **Mysql**, dan build automation tool menggunakan **maven**(agar projek java yang memerlukan dependency, kita tidak melakukan download dan menambahkan jar kedalam java build path secara manual tetapi secara otomatis menggunakan maven). **NOTE : UNTUK SOURCE CODE APLIKASI BISA DI DOWNLOAD DIGITHUB**

2. Tujuan dan Contoh aplikasi yang akan dibuat

Praktek pemrograman database bertujuan **untuk menyimpan data yang telah diolah, atau menampilkan data yang telah disimpan kemudian ditampilkan ke user sebagai informasi yang berguna**. Contoh aplikasi yang akan kita buat adalah aplikasi yang bisa melakukan beberapa hal, antara lain:

1. Melihat data mahasiswa
2. Merubah data mahasiswa
3. Menghapus data mahasiswa
4. Mencari data mahasiswa berdasarkan nim(nomor induk mahasiswa)

Untuk lebih paham terhadap aplikasi yang akan kita buat, simak flow dibawah ini



3. Langkah-langkah pembuatan aplikasi

Step-step yang harus dilakukan untuk mengikuti pemrograman database ini

1. Pembaca diharuskan sudah install database mysql di laptop/pc masing-masing
2. Pembaca login ke mysql
3. Setelah login ke mysql kemudian buat database **dbkampus**
4. Kemudian buat user dengan nama **userkampus** dan pass=**password**
5. Kasih akses /grant **userkampus** terhadap **dbkampus**
6. Buat tabel dengan menjalankan perintah berikut ini

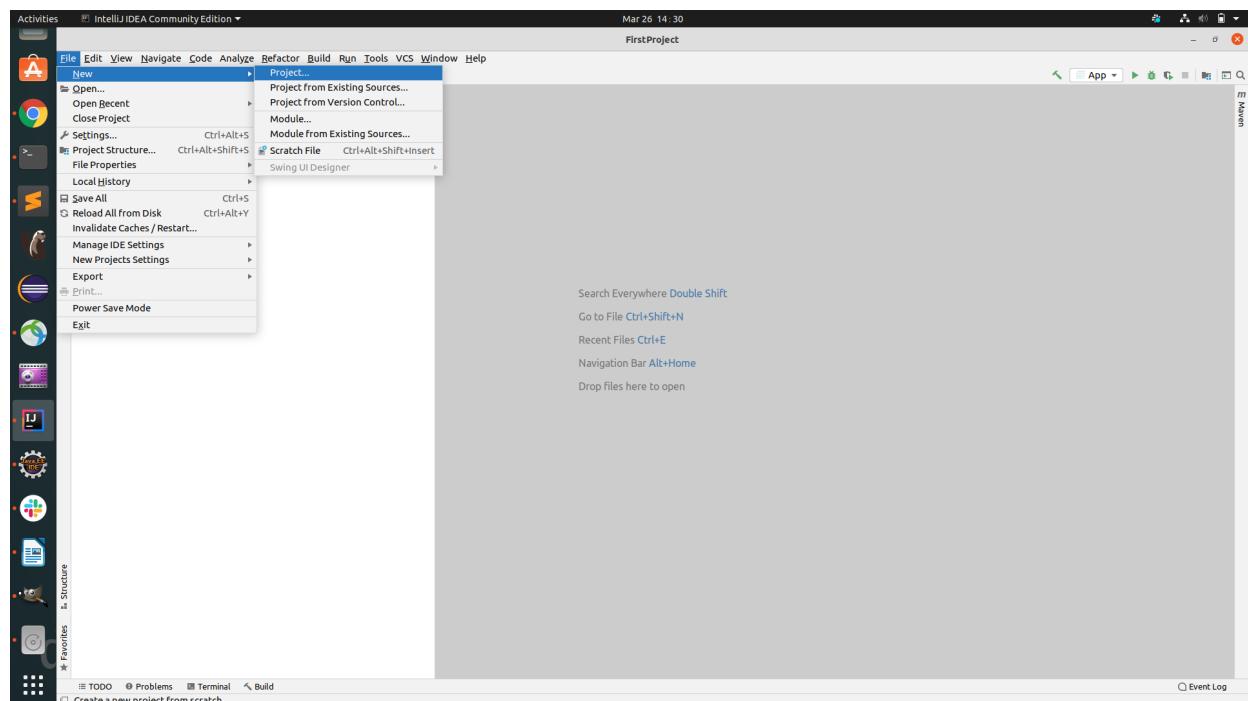
```
CREATE TABLE dbkampus.TB_MAHASISWA (
    NIM VARCHAR(20) NOT NULL,
    NAMA VARCHAR(100) NULL,
    ALAMAT VARCHAR(100) NULL,
    FAKULTAS VARCHAR(100) NULL
)
ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

Step selanjutnya, kita akan fokus di IntelliJ IDEA

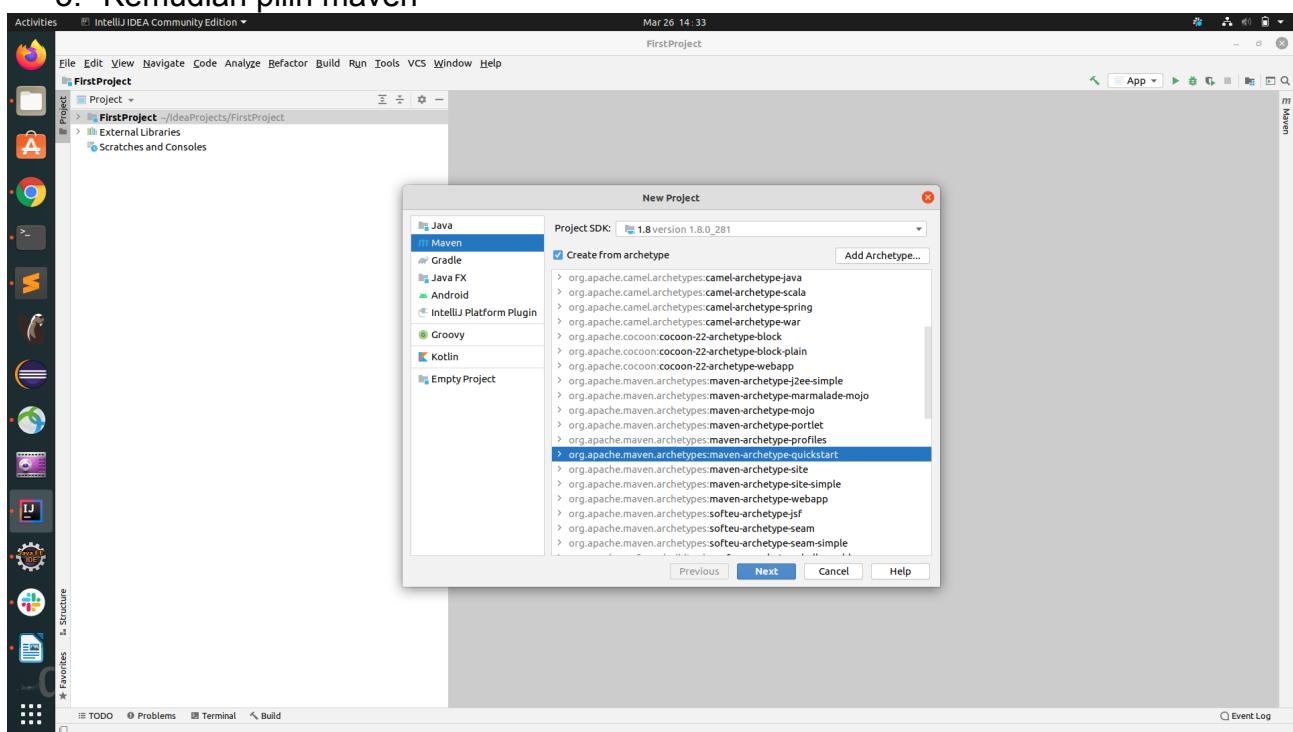
1. Buka IntelliJ IDEA



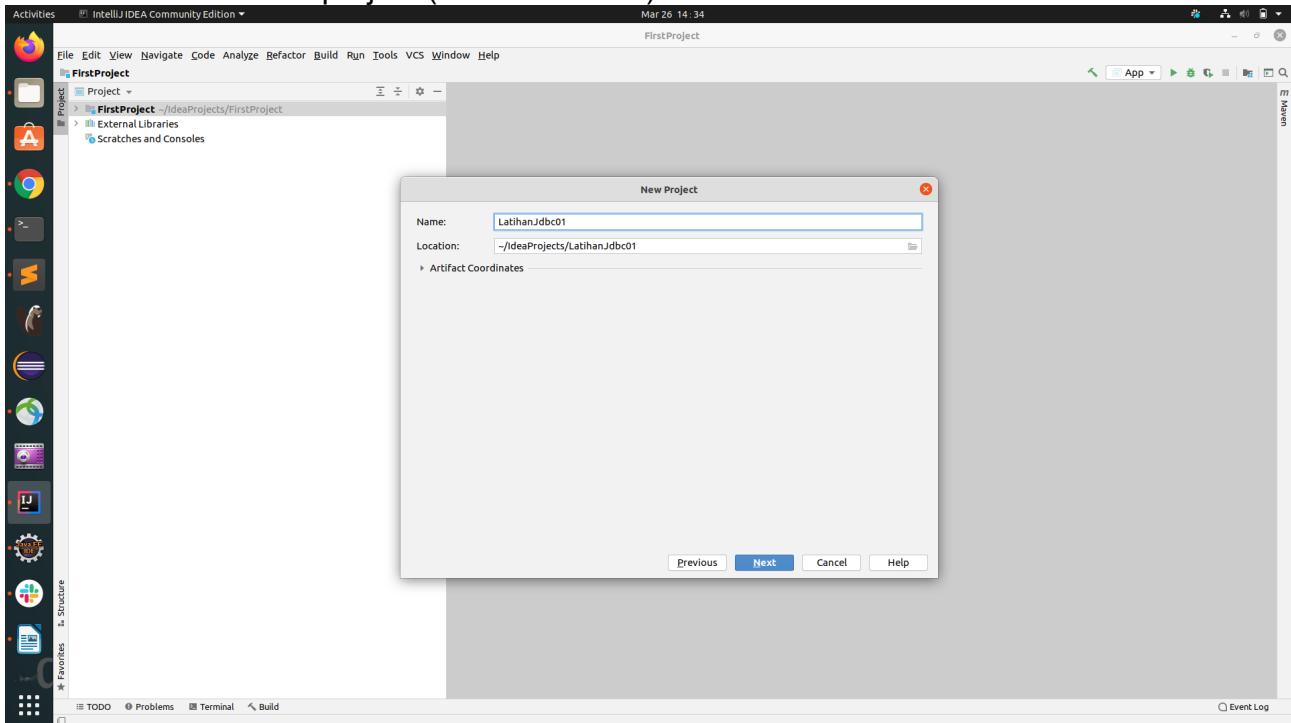
2. Buat Project maven



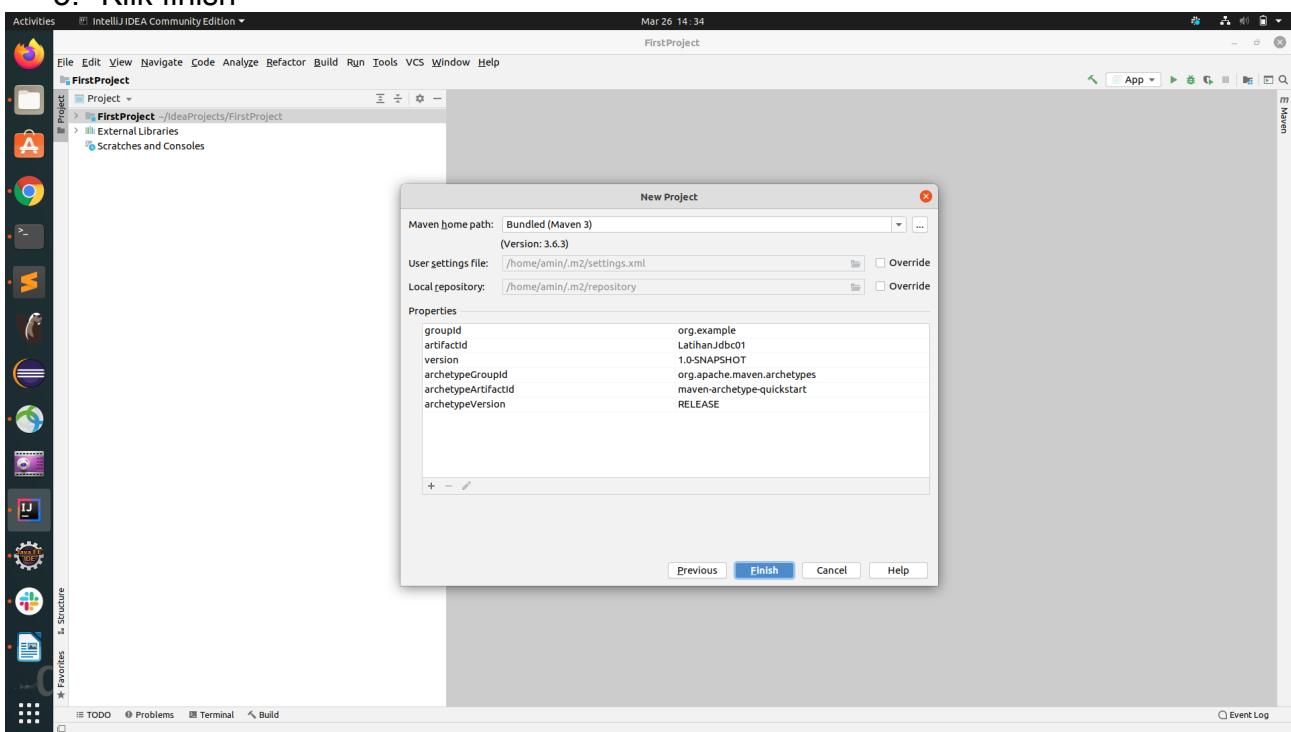
3. Kemudian pilih maven



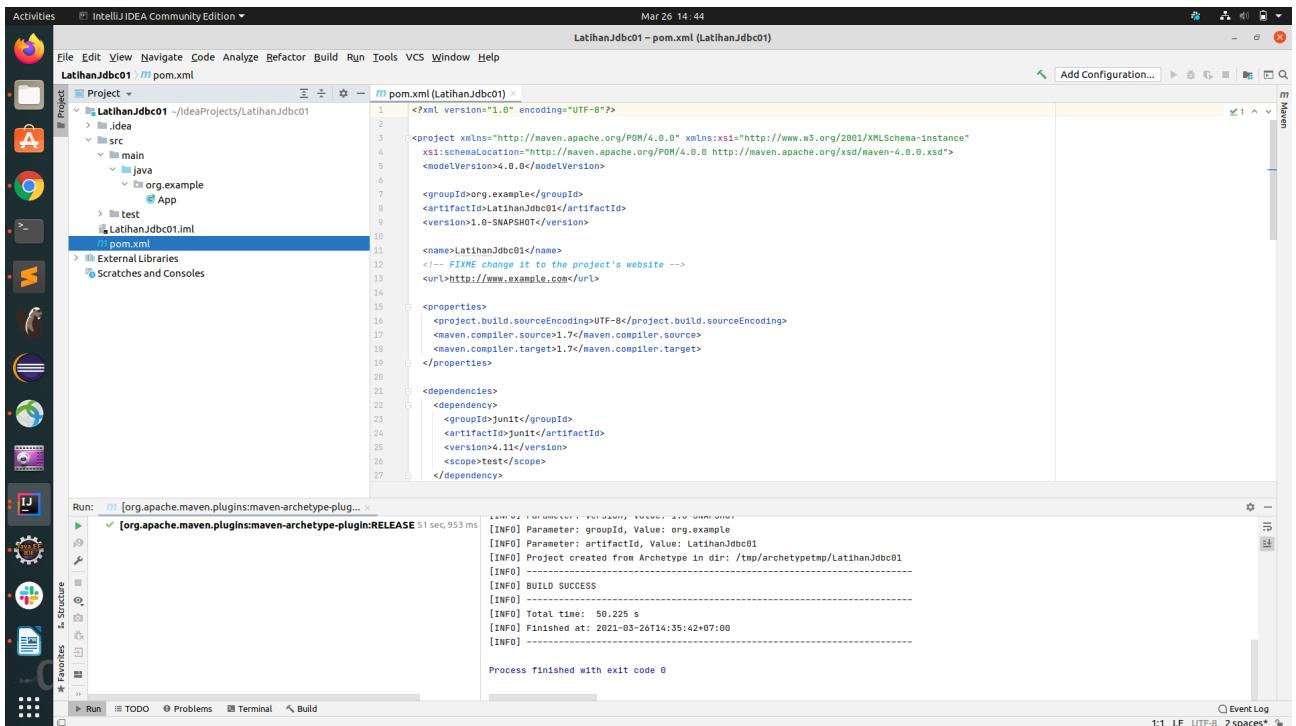
4. Ketikkan nama project (LatihanJdbc01) lalu next



5. Klik finish



6. Setelah Finish maka struktur folder dari project maven seperti pada gambar



The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The left sidebar shows the project structure with a tree view. The root project is "LatihanJdbc01". It contains a ".idea" folder, a "src" folder with "main" and "test" subfolders, and an "org.example" package containing an "App" class. There is also a "LatihanJdbc01.iml" file.
- pom.xml Content:** The main editor window displays the XML code for the pom.xml file. Key parts of the code include:
 - Project declaration with groupId "org.example", artifactId "LatihanJdbc01", and version "1.0-SNAPSHOT".
 - Properties section setting sourceEncoding to UTF-8.
 - Dependencies section adding JUnit dependency.
- Run Tab:** The bottom tab bar shows the run configuration for "org.apache.maven.plugins:maven-archetype-plugin:RELEASE". The output pane shows the build process completed successfully.
- Bottom Status Bar:** The status bar indicates the current encoding is UTF-8 and the code style is set to 2 spaces.

7. Langkah selanjutnya adalah menambahkan dependency yang kita butuhkan dan update maven. dependency yang kita butuhkan disini adalah mysql connector(**note : versi mysql connector harus sama dengan versi mysql**)

Activities IntelliJ IDEA Community Edition Mar 26 14:55

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help LatihanJdbc01 pom.xml

LatihanJdbc01 ~/ideaProjects/LatihanJdbc01

Project src main java org.example App test LatihanJdbc01.iml pom.xml New Cut Copy Copy Path... Set Paste Find Usages Alt+Shift+F7 Analyze Refactor Validate Add to Favorites Reformat Code Ctrl+Alt+L Optimize Imports Ctrl+Alt+O Delete... Delete Mark as Plain Text Build Module 'LatihanJdbc01'

Run: Open in Right Split Shift+Enter Open in Local History Reload from Disk Compare With... Mark Directory as Generate XSD Schema from XML File... Create Gist... Maven Add as Ant Build File

Reload project Generate Sources and Update Folders Ignore Projects - Unlink Maven Projects Create 'settings.xml' Create 'profiles.xml'

Download Sources Download Documentation Download Sources and Documentation Show Effective POM

Process finished with exit code 0

26:18 LF UTF-8 2 spaces* Event Log

```
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.23</version>
</dependency>
```

```
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.23</version>
</dependency>
```

8. Buat folder resources

Activities IntelliJ IDEA Community Edition Mar 26 15:00

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help LatihanJdbc01 src main pom.xml (LatihanJdbc01)

LatihanJdbc01 ~/ideaProjects/LatihanJdbc01

Project src main New File Scratch File Ctrl+Alt+Shift+Insert Directory Ctrl+Shift+R <!-- <project.build.sourceEncoding> --> HTML File n.compiler.source n.compiler.target Kotlin Script Kotlin Worksheet EditorConfig File Resource Bundle

External Lib <-- 1.8 > /j Maven: ju Maven: or Scratches as Analyze Refactor Add to Favorites Reformat Code Ctrl+Alt+L Optimize Imports Ctrl+Alt+O Delete... Delete Build Module 'LatihanJdbc01'

Open In Local History Reload from Disk Compare With... Mark Directory as Remove BOM Analyze

Run: [org.ap] Convert Java File to Kotlin File Ctrl+Alt+Shift+K 151 sec, 953 ms

[INFO] Parameter: groupId, Value: org.example
[INFO] Parameter: artifactId, Value: LatihanJdbc01
[INFO] Project created from Archetype in dir: /tmp/archetypetmp/LatihanJdbc01
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 50.225 s
[INFO] Finished at: 2021-03-26T14:35:42+07:00
[INFO] -----
Process finished with exit code 0

20:1 LF UTF-8 2 spaces* Event Log

Create new directory or package

Pilih resources

Activities IntelliJ IDEA Community Edition Mar 26 15:03

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help LatihanJdbc01 pom.xml (LatihanJdbc01)

LatihanJdbc01 ~/ideaProjects/LatihanJdbc01

Project src main pom.xml (LatihanJdbc01)

<name>LatihanJdbc01</name>
<!-- FIXME change it to the project's website -->
<url>http://www.example.com</url>

<properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<maven.compiler.source>1.7</maven.compiler.source>
<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.23</version>

</dependency>

<dependency>

<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.11</version>
<scope>test</scope>

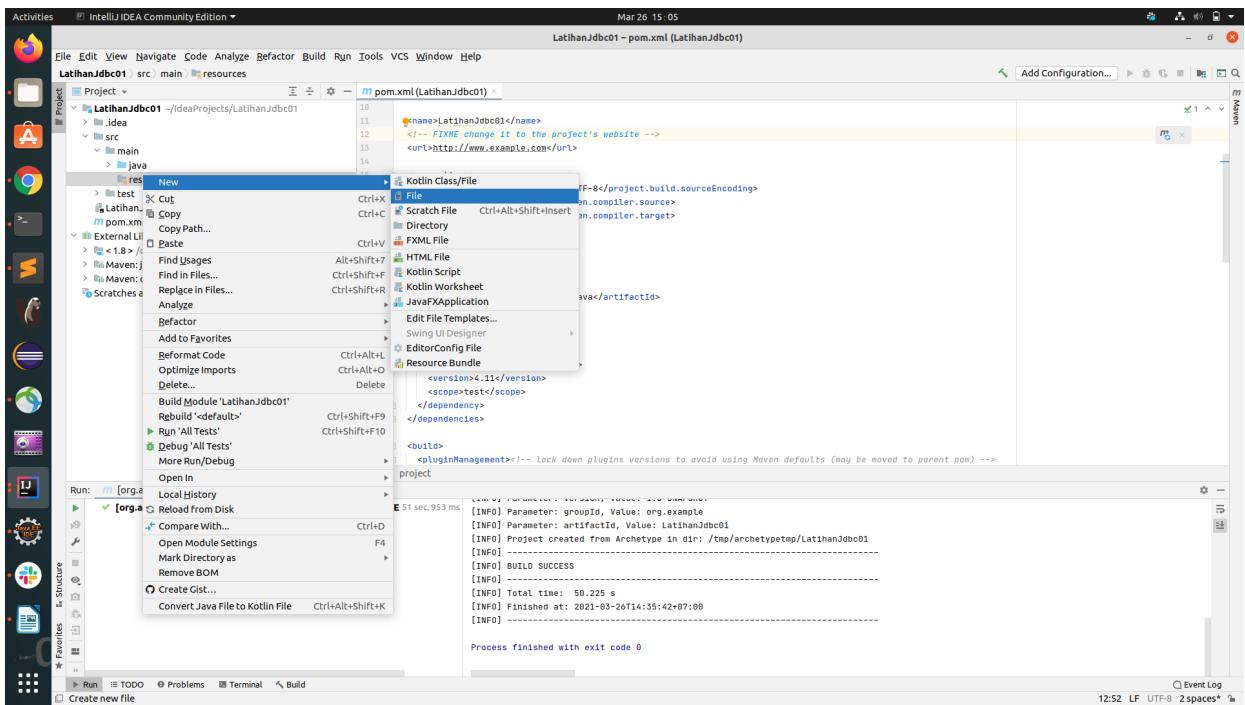
</dependency>

</dependencies>

<build>

<pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

9. Buat file db.properties, file ini digunakan untuk konfigurasi koneksi ke database mysql



10. Kemudian ketikan db.properties

```
<name>LatihanJdbc01</name>
<!-- FIXMe change it to the project's website -->
<url>http://www.example.com</url>

<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
</properties>

<dependencies>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.23</version>
    </dependency>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.11</version>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->
        <!--
            groupId: org.apache.maven.plugins
            artifactId: maven-archetype-plugin
            version: 3.0.1
            parameters:
                groupId: org.example
                artifactId: LatihanJdbc01
        -->
        <!--
            groupId: org.apache.maven.plugins
            artifactId: maven-archetype-plugin
            version: 3.0.1
            parameters:
                groupId: org.example
                artifactId: LatihanJdbc01
        -->
    </pluginManagement>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-archetype-plugin</artifactId>
            <version>RELEASE</version>
            <configuration>
                <groupId>org.example</groupId>
                <artifactId>LatihanJdbc01</artifactId>
            </configuration>
        </plugin>
    </plugins>
</build>

```

11. Kemudian copy code konfigurasi berikut ke dalam file db.properties yang sudah dibuat tadi

```
#mysql DB properties
DB_DRIVER_CLASS=com.mysql.cj.jdbc.Driver
DB_URL=jdbc:mysql://localhost:3306/dbkampus
DB_USERNAME=userkampus
DB_PASSWORD=password
```

Sehingga tampilan nya akan terlihat seperti dibawah ini,

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The project is named "LatihanJdbc01". It contains a ".idea" folder, a "src" directory with "main" and "java" sub-directories, and a "resources" directory which contains a "db.properties" file.
- Code Editor:** The "db.properties" file is open, showing the following content:

```
mysql DB properties
DB_DRIVER_CLASS=com.mysql.cj.jdbc.Driver
DB_URL=jdbc:mysql://localhost:3306/dbkampus
DB_USERNAME=userkampus
DB_PASSWORD=password
```
- Run Tab:** A Maven build is shown with the output:

```
[INFO] Parameter: version, Value: 2.0-SNAPSHOT
[INFO] Parameter: groupId, Value: org.example
[INFO] Parameter: artifactId, Value: LatihanJdbc01
[INFO] Project created from Archetype in dir: /tmp/archetypetmp/LatihanJdbc01
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 50.225 s
[INFO] Finished at: 2021-03-26T14:35:42+07:00
[INFO] -----
```

Process finished with exit code 0
- Bottom Status Bar:** Shows the time as 5:21, LF, UTF-8, and 4 spaces.

DB_DRIVER_CLASS = Class driver untuk koneksi ke database mysql, setiap database memiliki Class driver yang berbeda-beda

DB_URL = url dari database kita dan nama database kita(bila database kita di install di localhost maka ip nya menggunakan localhost)

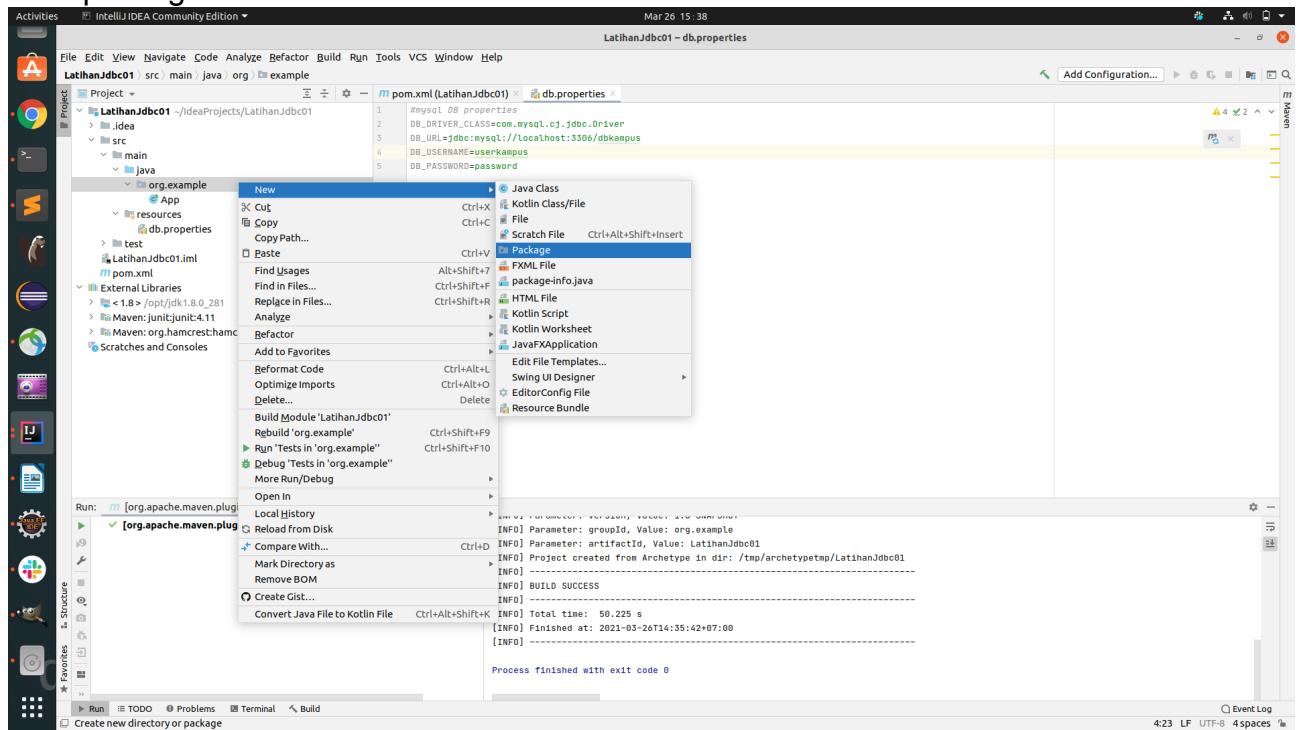
DB_USERNAME = user name dari database

DB_PASSWORD = password dari database

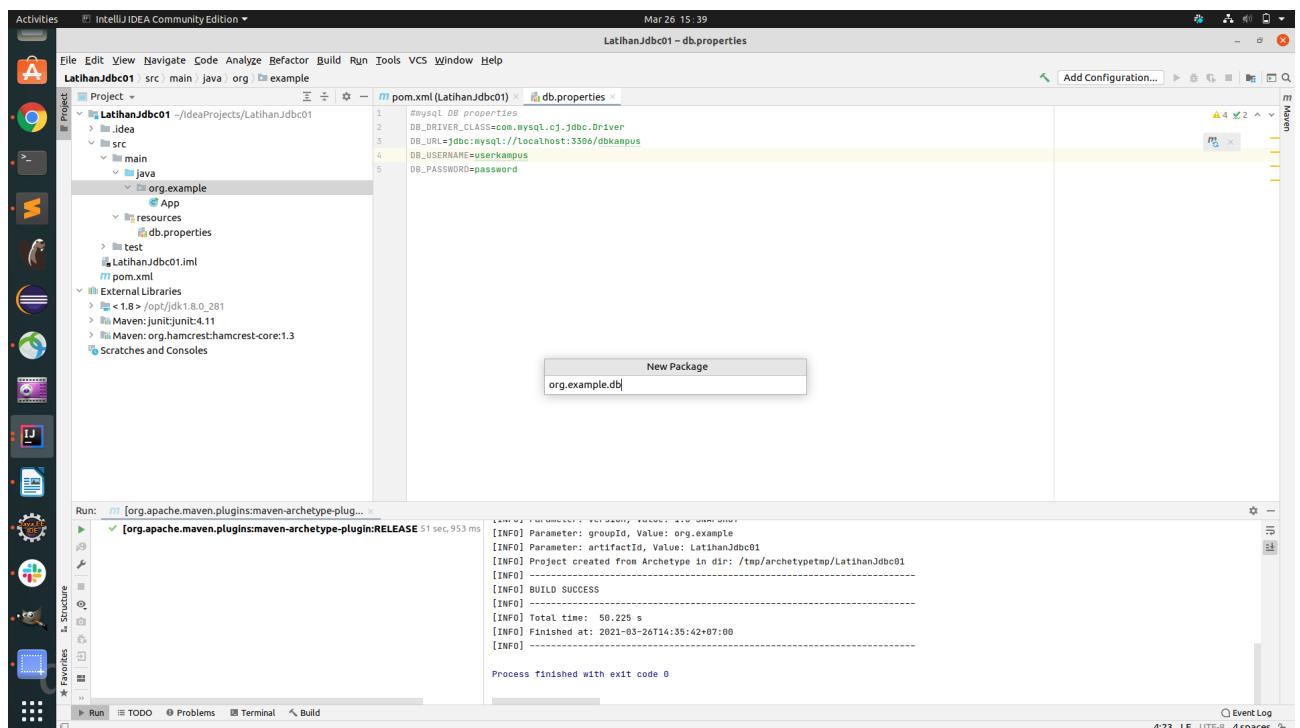
12. buat package db

Package dibuat untuk mengelompokkan kelas berdasarkan kegunaannya, package db disini untuk mengelompokkan kelas yang mengatur koneksi ke database.

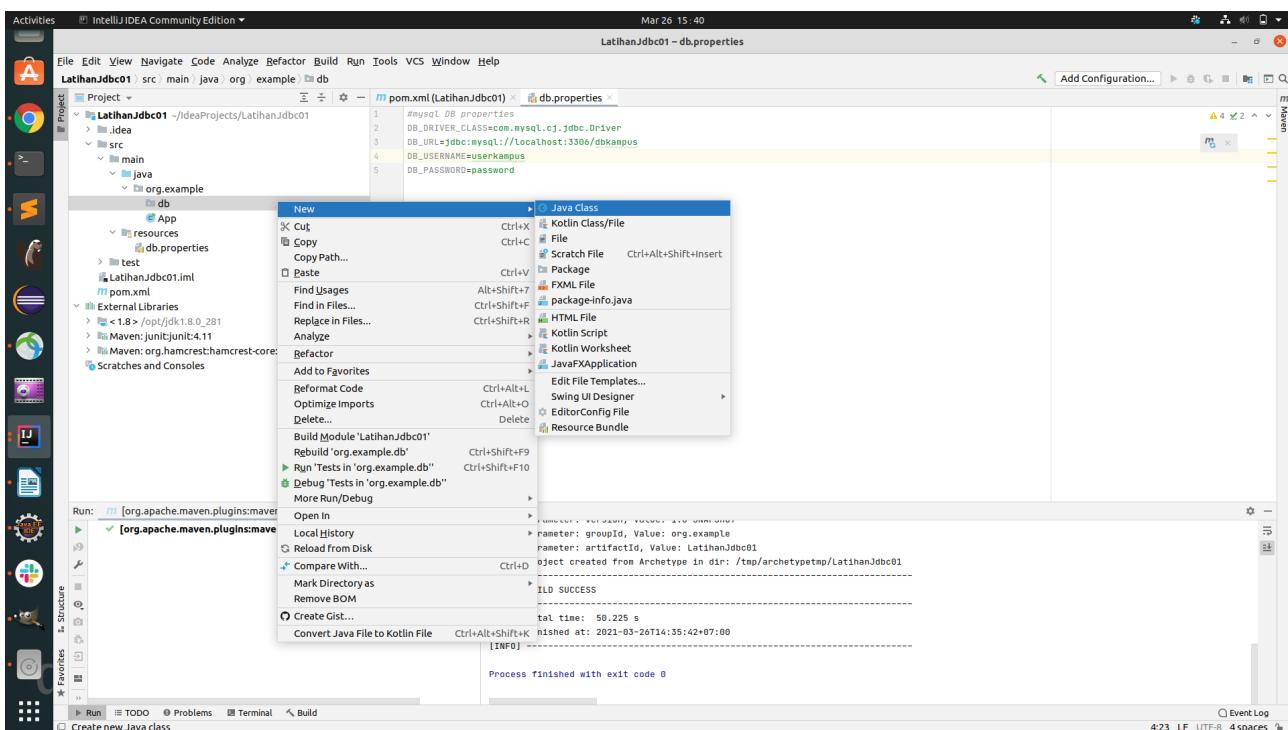
Buat package db



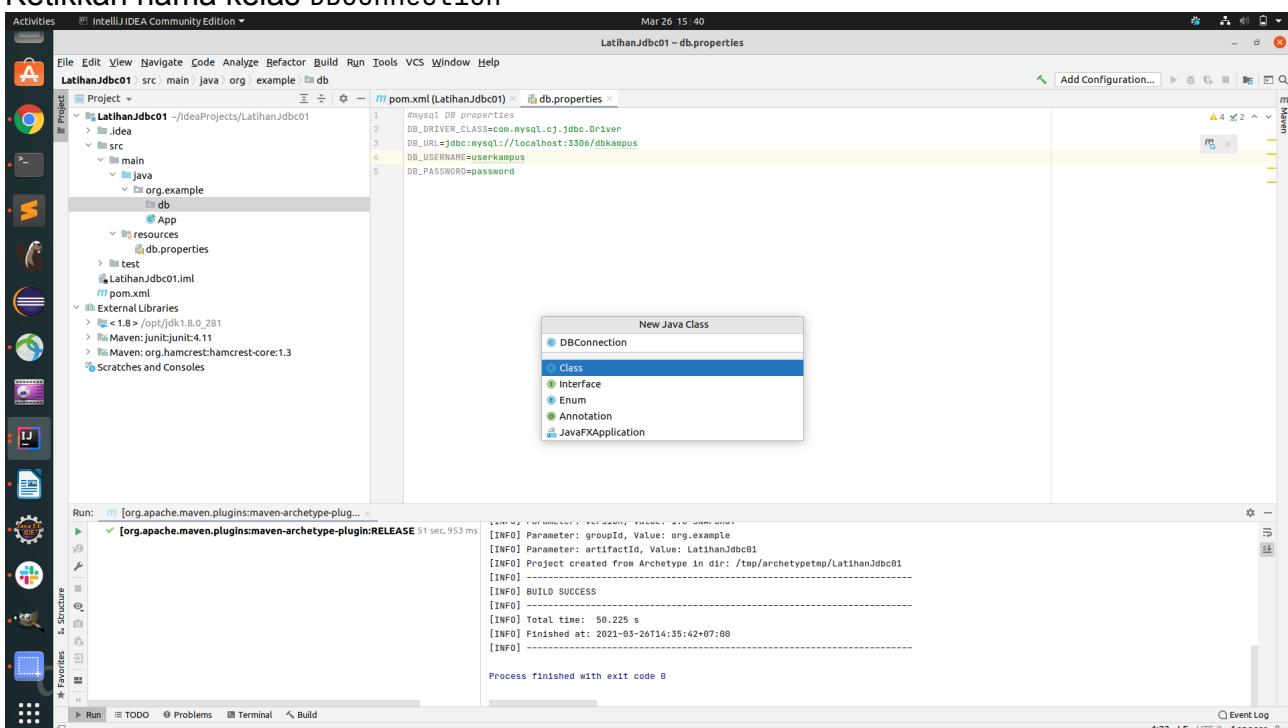
ketikkan db



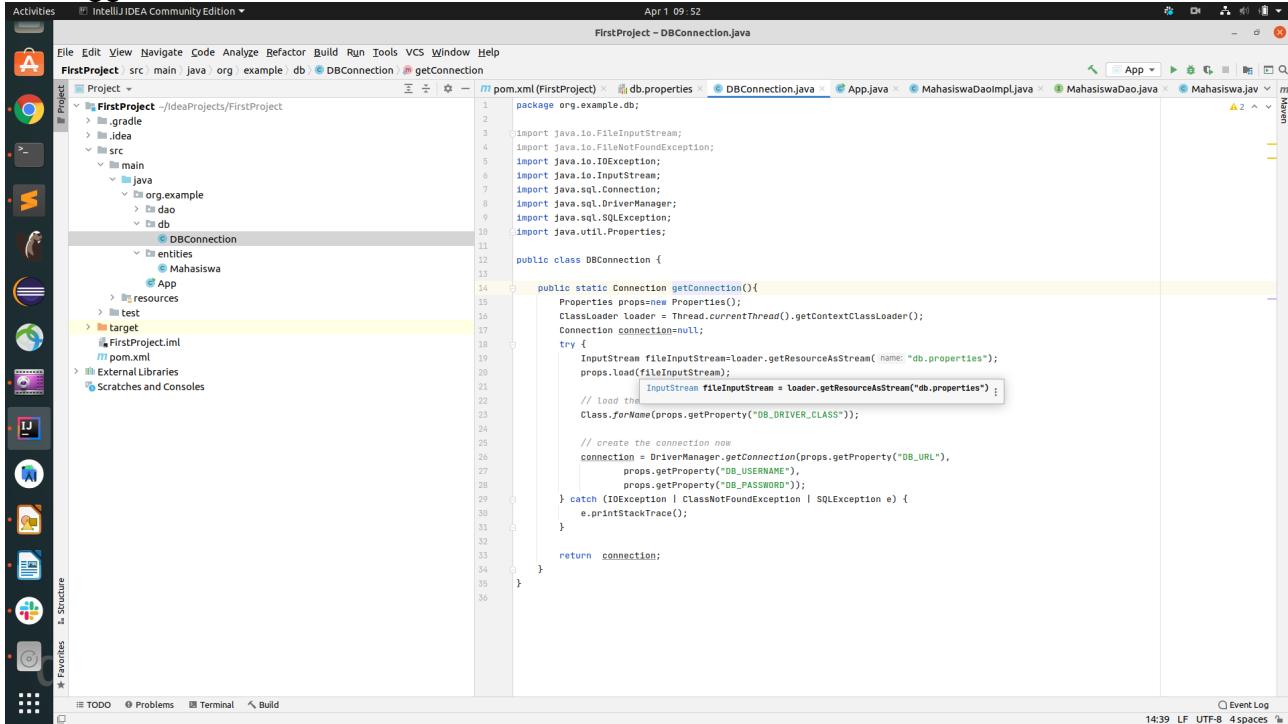
13. Kemudian buat kelas yang digunakan untuk koneksi ke database,



Ketikkan nama kelas DBConnection



Sehingga kelas DBConnection tampil seperti dibawah ini

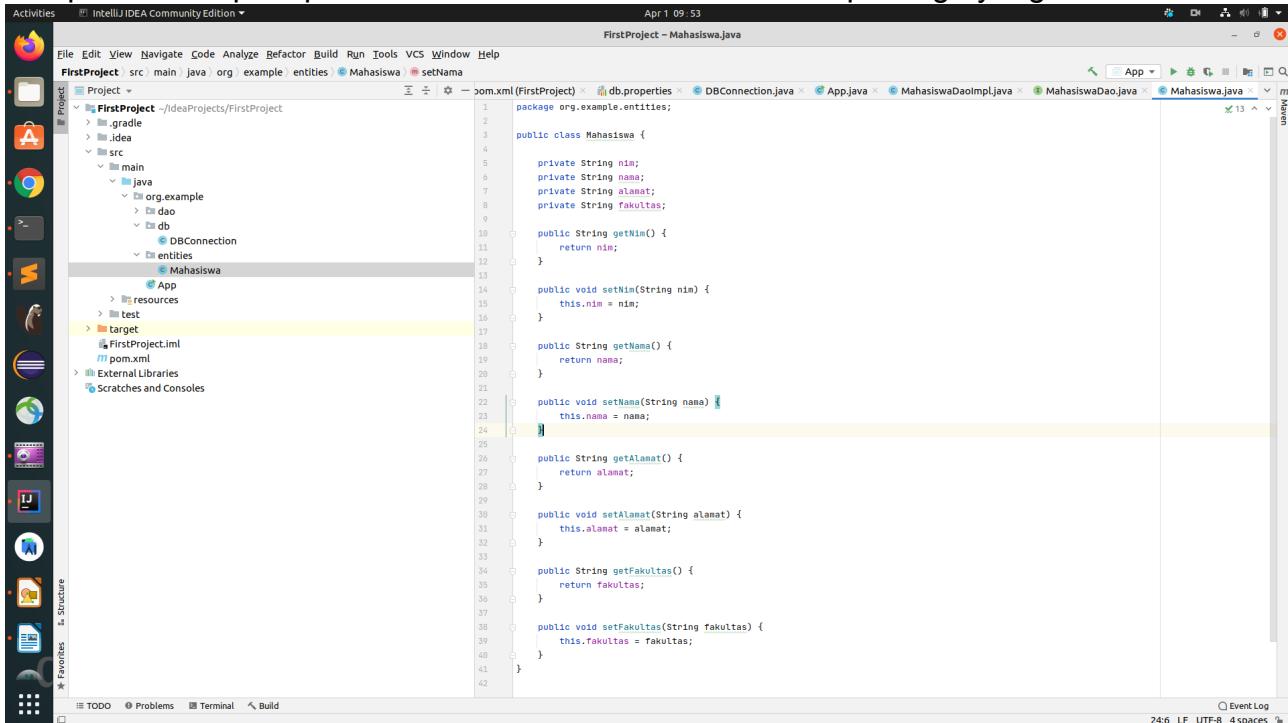


```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FirstProject src main java org example db DBConnection getConnection
FirstProject - DBConnection.java
Activities IntelliJ IDEA Community Edition ▾ Apr 1 09:52
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FirstProject - DBConnection.java
Project ▾ FirstProject ~/ideaProjects/FirstProject
  > .gradle
  > .idea
  > pom.xml
  > src
    > main
      > java
        > org.example
          > db
            > DBConnection
              > entities
                > Mahasiswa
              > App
            > resources
          > test
        > target
        > FirstProject.iml
      > pom.xml
    > External Libraries
  > Scratches and Consoles
  Favorites & Structure
  * Favorites & Structure
  & TODO & Problems & Terminal & Build
  pom.xml (FirstProject) db.properties DBConnection.java App.java MahasiswaDaoImpl.java MahasiswaDao.java Mahasiswa.java
  1 package org.example.db;
  2
  3 import java.io.FileInputStream;
  4 import java.io.FileNotFoundException;
  5 import java.io.IOException;
  6 import java.io.InputStream;
  7 import java.sql.Connection;
  8 import java.sql.DriverManager;
  9 import java.sql.SQLException;
 10 import java.util.Properties;
 11
 12 public class DBConnection {
 13
 14     public static Connection getConnection(){
 15         Properties props=new Properties();
 16         ClassLoader loader = Thread.currentThread().getContextClassLoader();
 17         Connection connection=null;
 18         try {
 19             InputStream fileInputStream=loader.getResourceAsStream( name: "db.properties");
 20             props.load(fileInputStream);
 21             // load the properties
 22             InputStream fileInputStream = loader.getResourceAsStream("db.properties");
 23             Class.forName(props.getProperty("DB_DRIVER_CLASS"));
 24
 25             // create the connection now
 26             connection = DriverManager.getConnection(props.getProperty("DB_URL"),
 27                 props.getProperty("DB_USERNAME"),
 28                 props.getProperty("DB_PASSWORD"));
 29         } catch (IOException | ClassNotFoundException | SQLException e) {
 30             e.printStackTrace();
 31         }
 32
 33         return connection;
 34     }
 35
 36 }
 37
 38
 39
 40
 41
 42
```

Event Log
14:39 LF UTF-8 4 spaces

14. buat package entities dan kelas Mahasiswa

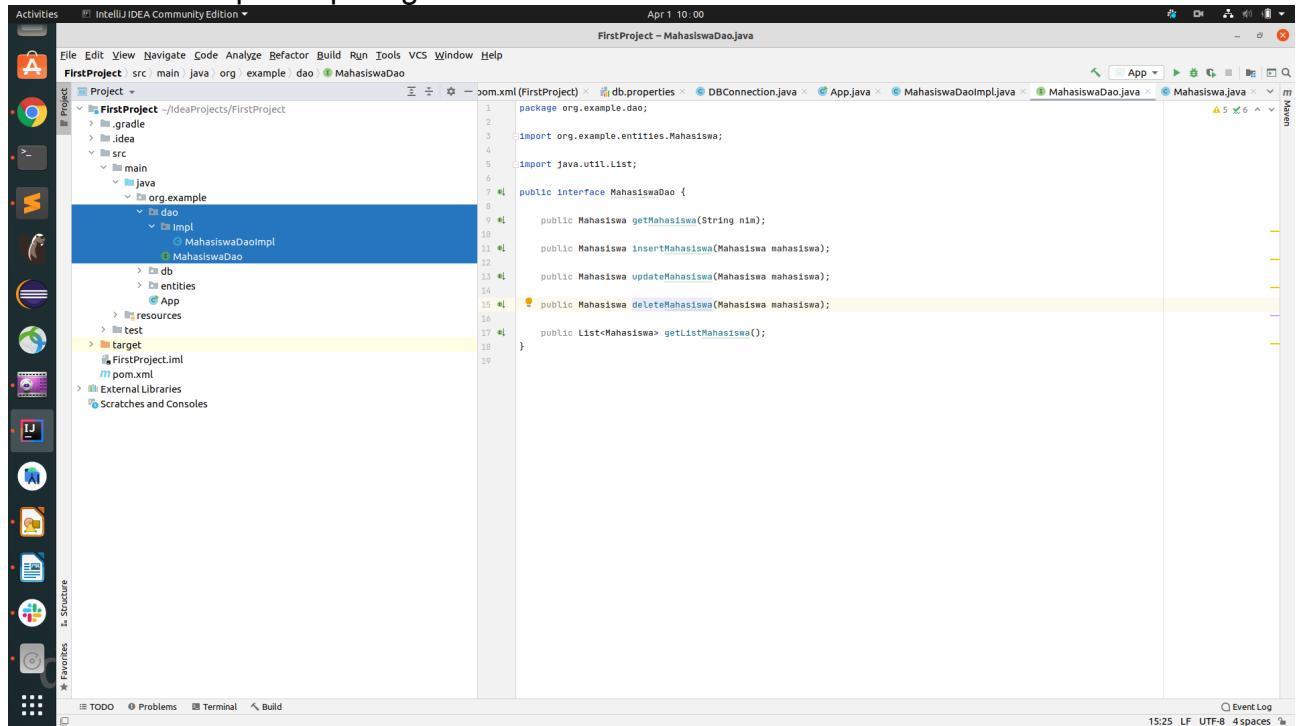
package ini dibuat untuk mengelompokkan kelas yang berfungsi, untuk mapping kolom dari suatu tabel dengan kelas entity. **Property yang ada di kelas entity harus sama dengan jumlah kolom yang ada didalam tabel.** Cara pembuatan package dan kelas hampir sama seperti pada contoh diatas. Berikut kelas dan package yang sudah dibuat



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FirstProject - Mahasiswa.java
Activities IntelliJ IDEA Community Edition ▾ Apr 1 09:53
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FirstProject - Mahasiswa.java
Project ▾ FirstProject ~/ideaProjects/FirstProject
  > .gradle
  > .idea
  > pom.xml
  > src
    > main
      > java
        > org.example
          > db
            > DBConnection
            > entities
              > Mahasiswa
              > App
            > resources
          > test
        > target
        > FirstProject.iml
      > pom.xml
    > External Libraries
  > Scratches and Consoles
  Favorites & Structure
  * Favorites & Structure
  & TODO & Problems & Terminal & Build
  pom.xml (FirstProject) db.properties DBConnection.java App.java MahasiswaDaoImpl.java MahasiswaDao.java Mahasiswa.java
  1 package org.example.entities;
  2
  3 public class Mahasiswa {
  4
  5     private String nim;
  6     private String name;
  7     private String alamat;
  8     private String fakultas;
  9
 10     public String getNim() {
 11         return nim;
 12     }
 13
 14     public void setNim(String nim) {
 15         this.nim = nim;
 16     }
 17
 18     public String getName() {
 19         return name;
 20     }
 21
 22     public void setName(String name) {
 23         this.name = name;
 24     }
 25
 26     public String getAlamat() {
 27         return alamat;
 28     }
 29
 30     public void setAlamat(String alamat) {
 31         this.alamat = alamat;
 32     }
 33
 34     public String getFakultas() {
 35         return fakultas;
 36     }
 37
 38     public void setFakultas(String fakultas) {
 39         this.fakultas = fakultas;
 40     }
 41 }
 42
```

Event Log
24:6 LF UTF-8 4 spaces

15. buat package **dao** dan **daolmpl**. Buat juga interface **MahasiswaDao** di package **dao** yang telah dibuat dan buat **Kelas MahasiswaDaolmpl** di package **daolmpl**.
Package dao dibuat untuk mengelompokkan kelas yang berfungsi untuk operasi insert, update, delete dan select ke database mysql. Untuk cara pembuatan package dan kelas nya sama seperti cara di atas . Berikut package , interface dan kelas yang dibuat akan tampak seperti gambar dibawah ini



The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The project is named "FirstProject". The "src" directory contains "main" and "org.example" packages. "org.example" has "dao" and "entities" sub-packages. "dao" contains "MahasiswaDaolmpl" and "MahasiswaDao". "entities" contains "Mahasiswa".
- MahasiswaDao.java Content:**

```

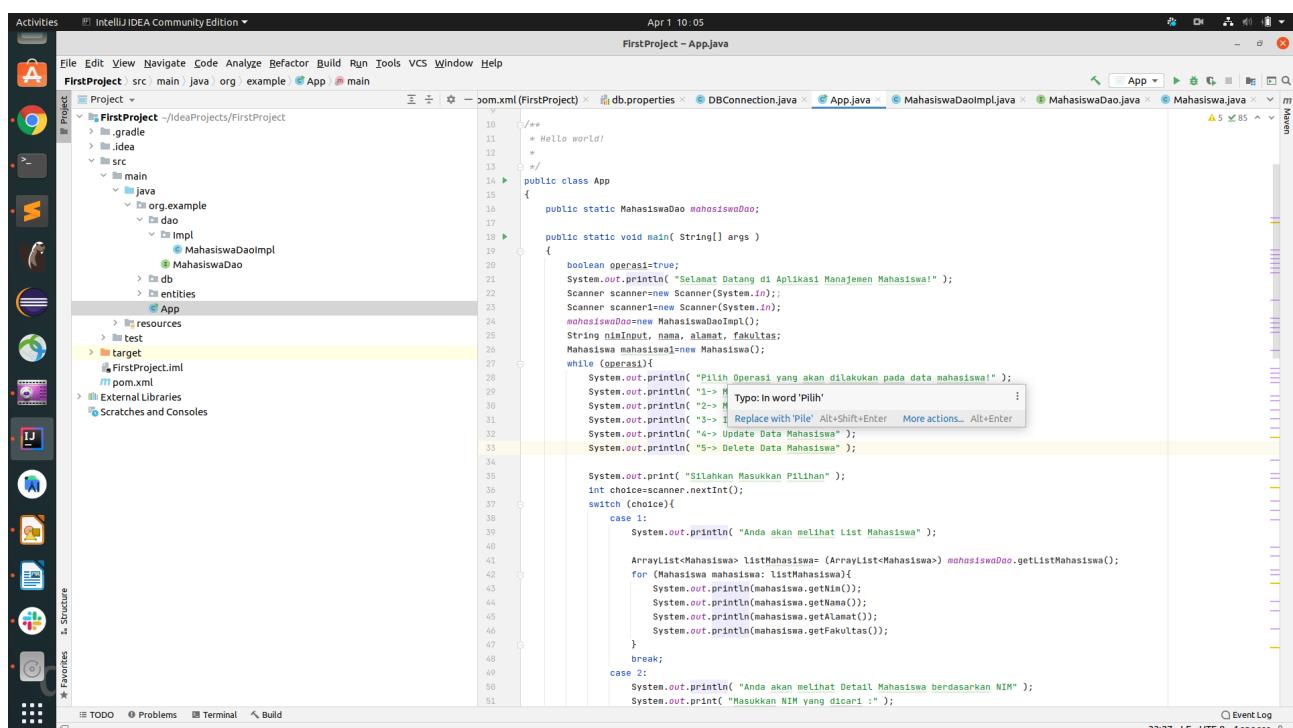
package org.example.dao;

import org.example.entities.Mahasiswa;
import java.util.List;

public interface MahasiswaDao {
    public Mahasiswa getMahasiswa(String nim);
    public Mahasiswa insertMahasiswa(Mahasiswa mahasiswa);
    public Mahasiswa updateMahasiswa(Mahasiswa mahasiswa);
    public Mahasiswa deleteMahasiswa(Mahasiswa mahasiswa);
    public List<Mahasiswa> getListMahasiswa();
}

```
- Code Editor:** The code editor shows the above Java code for the interface.
- Toolbars and Status Bar:** The top bar shows "File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help". The status bar at the bottom right shows "Apr 1 10:00", "15:25 LF UTF-8 4 spaces", and "Event Log".

16. Tulis sourcecode di class App(main class) sesuai flow, berikut tampilannya



The screenshot shows the IntelliJ IDEA interface with the following details:

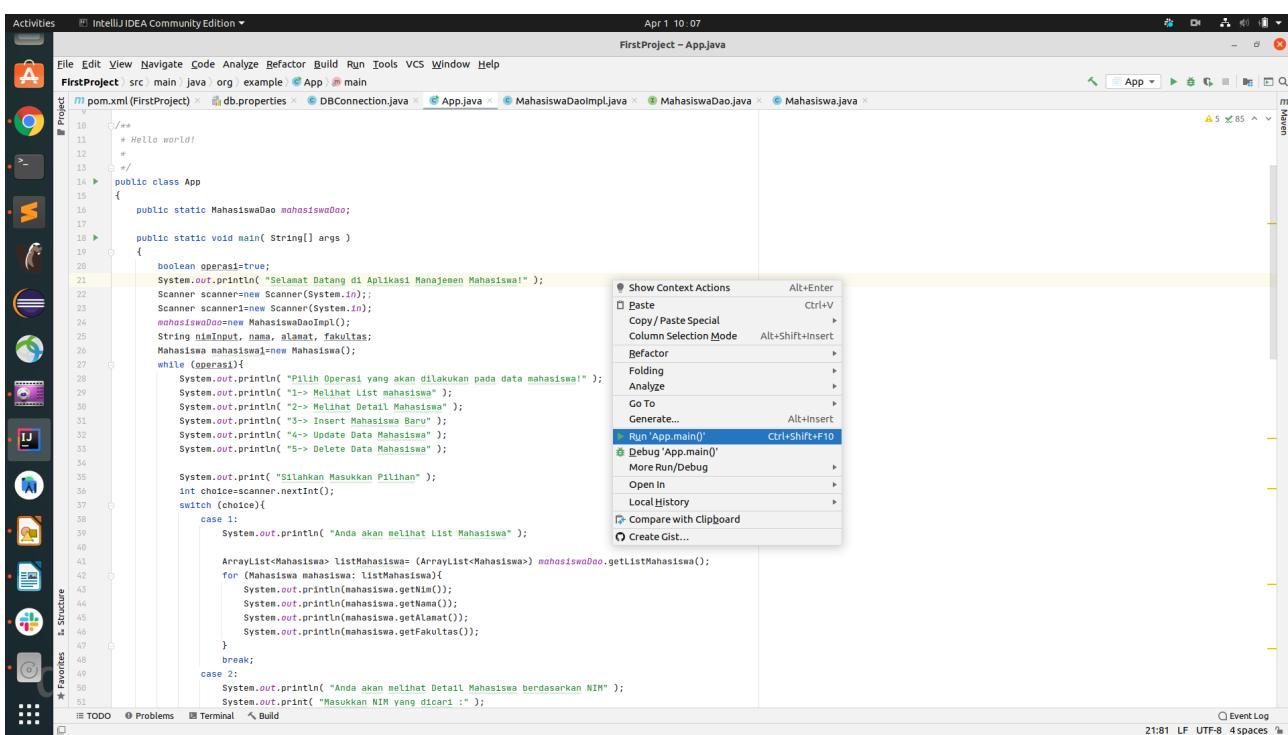
- Project Structure:** The project is named "FirstProject". The "src" directory contains "main" and "org.example" packages. "org.example" has "dao" and "entities" sub-packages. "dao" contains "MahasiswaDaolmpl" and "MahasiswaDao". "entities" contains "Mahasiswa".
- App.java Content:**

```

/**
 * Hello world!
 */
public class App {
    public static MahasiswaDao mahasiswaDao;
    public static void main( String[] args )
    {
        boolean operasi=true;
        System.out.println( "Selamat Datang di Aplikasi Manajemen Mahasiswa!" );
        Scanner scanner=new Scanner(System.in);
        Scanner scanner1=new Scanner(System.in);
        mahasiswaDao=new MahasiswaDaolmpl();
        String nimInput, nama, alamat, fakultas;
        Mahasiswa mahasiswa1=new Mahasiswa();
        while (operasi){
            System.out.println( "Pilih Operasi yang akan dilakukan pada data mahasiswa" );
            System.out.print( "1-> M Type:in word'Pilih' ");
            System.out.print( "2-> M Type:in word'Update Mahasiswa' ");
            System.out.print( "3-> I Replace with 'Pilih' Alt+Shift+Enter More actions... Alt+Enter" );
            System.out.print( "4-> Update Data Mahasiswa" );
            System.out.print( "5-> Delete Data Mahasiswa" );
            System.out.print( "Silahkan Masukkan Pilihan" );
            int choice=scanner.nextInt();
            switch (choice){
                case 1:
                    System.out.println( "Anda akan melihat List Mahasiswa" );
                    ArrayList<Mahasiswa> listMahasiswa=(ArrayList<Mahasiswa>) mahasiswaDao.getListMahasiswa();
                    for (Mahasiswa mahasiswa: listMahasiswa{
                        System.out.println(mahasiswa.getNim());
                        System.out.println(mahasiswa.getNama());
                        System.out.println(mahasiswa.getAlamat());
                        System.out.println(mahasiswa.getFakultas());
                    }
                break;
                case 2:
                    System.out.println( "Anda akan melihat Detail Mahasiswa berdasarkan NIM" );
                    System.out.print( "Masukkan NIM yang dicari :" );

```
- Code Editor:** The code editor shows the Java code for the main class "App".
- Toolbars and Status Bar:** The top bar shows "File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help". The status bar at the bottom right shows "Apr 1 10:05", "33:27 LF UTF-8 4 spaces", and "Event Log".

17. Running Aplikasi seperti pada gambar berikut :



The screenshot shows the IntelliJ IDEA interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** FirstProject, src, main, java, org, example, App.java, DBConnection.java, MahasiswaDaoImpl.java, MahasiswaDao.java.
- Code Editor:** The code for `App.java` is visible, containing Java code for a command-line application that interacts with a database using JDBC.
- Context Menu:** A context menu is open over the code editor, showing options like "Run 'App.main()", "Debug 'App.main()", and "More Run/Debug".
- Toolbars and Panels:** Standard IntelliJ toolbars and panels are visible along the top and right side of the interface.
- Status Bar:** Shows the date (Apr 1 10:07), time (21:81), and file encoding (UTF-8).

Untuk full source code beserta penjelasannya bisa di download di github.
Demikian tutorial dari saya semoga bermanfaat, Happy learning and Happy Sharing !!!

Referensi

<https://www.journaldev.com/2471/jdbc-example-mysql-oracle>

<https://imdjkoch.wordpress.com/2010/07/26/database-programming-using-jdbc-an-introduction/>

<https://www.educba.com/jdbc-architecture/>