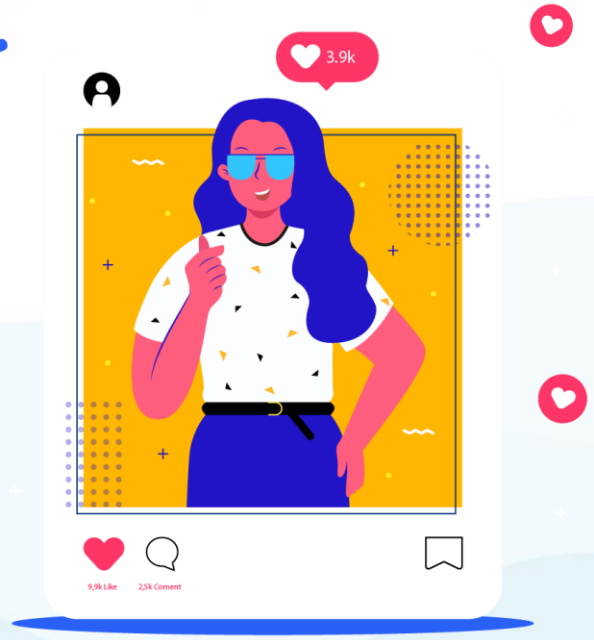
PROGRAM STUDI
SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

MATA KULIAH
PEMROGRAMAN BERORIENTASI OBYEK



Database

-penyusun-

Team penyusun matkul PBO

2021

Capaian Pembelajaran

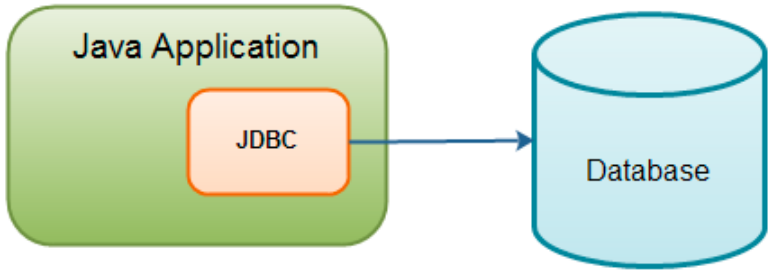
Mahasiswa memiliki kemampuan menjelaskan dan mempraktekkan Perintah DDL dan DML Database untuk diimplementasikan menggunakan bahasa pemrograman Java.

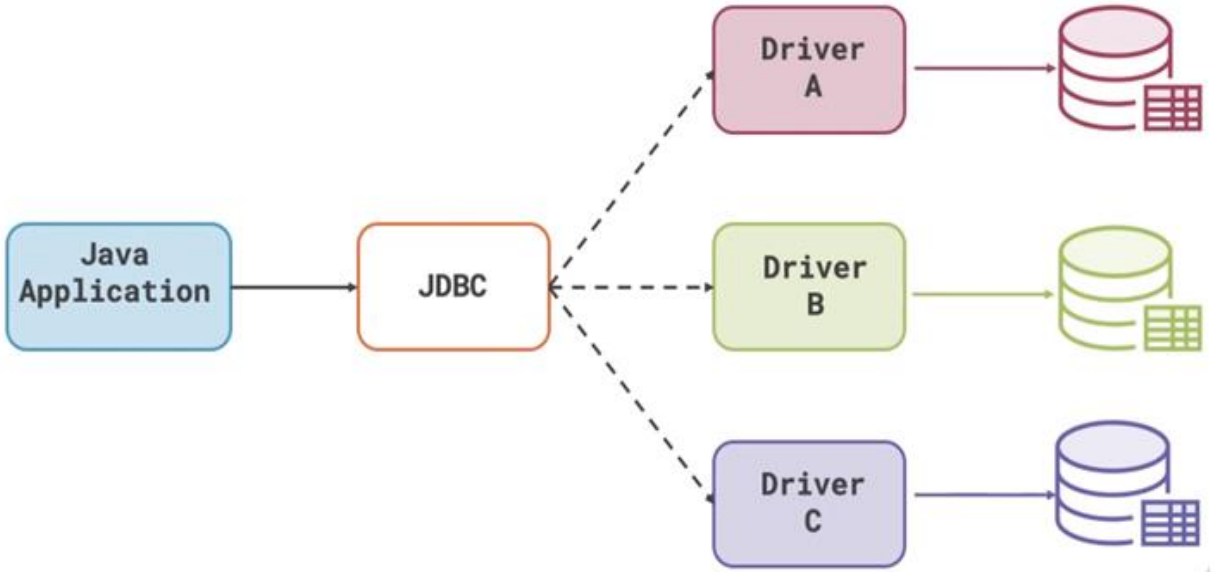
Kemampuan Akhir yang Diharapkan

- Mahasiswa memiliki kemampuan menjelaskan dan mempraktekkan database untuk mengelola data/tabel.
- Mahasiswa memiliki kemampuan menjelaskan dan mempraktekkan penggunaan database pada pembangunan sistem aplikasi.

Database

Agar data yang diinputkan dapat disimpan secara *'permanen'*, maka dapat dilakukan dengan disimpan pada suatu database.
Ada banyak sekali pilihan database yang bisa digunakan, diantaranya: MySQL, SQLite, PostgreSQL, Ms. SQL Server, Oracle, dan sebagainya.



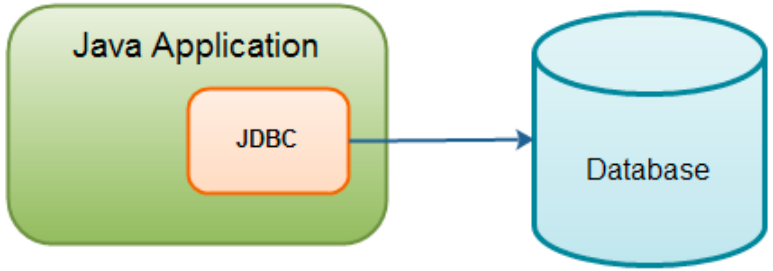


Database

Java menyediakan standard API (application Programming Interface) untuk pengembangan program aplikasi basis data (database) yang disebut dengan JDBC API.

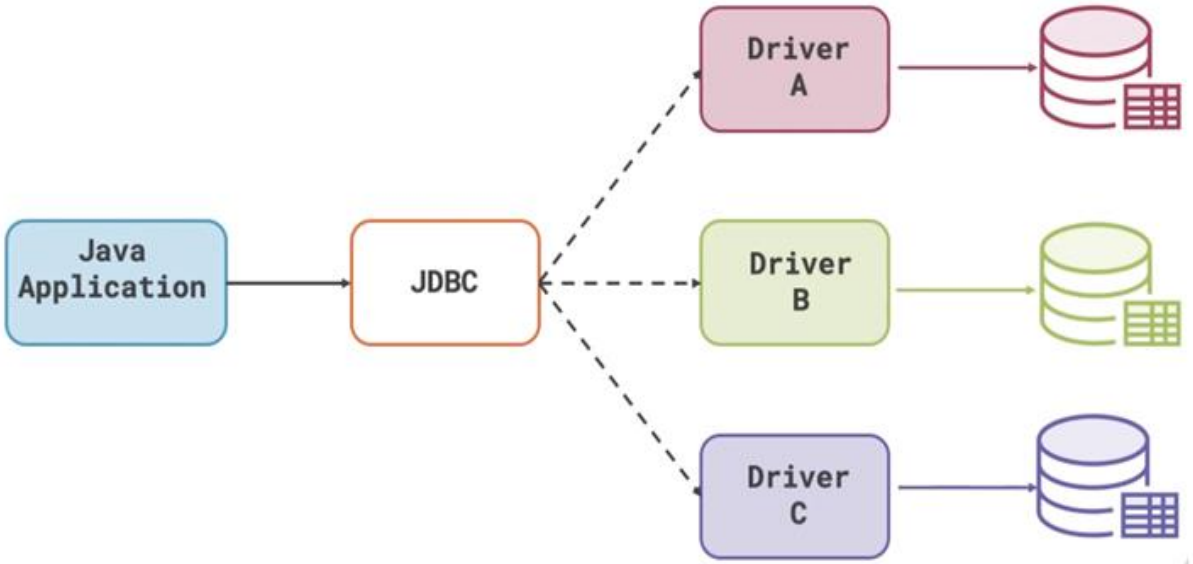
JDBC adalah API Java untuk memanipulasi basis data.
Dengan JDBC API, para pengembang aplikasi dan applet Java diberi kemudahan untuk mengakses berbagai tipe basis data dari berbagai penyedia basis data (database vendors) seperti MySQL Server, SQL Server, Oracle, Sybase dll.

JDBC merupakan perantara antara Java dengan basis data. JDBC adalah sebuah spesifikasi yang menyediakan sekumpulan interface yang membolehkan akses portabel ke semua basis data.



Database

Sebuah program Java yang mengakses data dari basis data harus menggunakan JDBC driver yang khusus untuk basis data tersebut. Ketika sebuah perusahaan beralih dari satu basis data ke basis data lain, maka program Java harus disesuaikan untuk menggunakan JDBC driver yang khusus untuk basis data baru tersebut.
Demikian pula agar program Java dapat berinteraksi dengan basis data MySQL, harus digunakan JDBC driver dari MySQL.



Database

Sebuah program Java yang mengakses data dari basis data harus menggunakan JDBC driver yang khusus untuk basis data tersebut. Ketika sebuah perusahaan beralih dari satu basis data ke basis data lain, maka program Java harus disesuaikan untuk menggunakan JDBC driver yang khusus untuk basis data baru tersebut.
Demikian pula agar program Java dapat berinteraksi dengan basis data MySQL, harus digunakan JDBC driver dari MySQL.

Database

Kita membutuhkan JDBC *(Java Data Base Connectivity)* untuk menghubungkan Java dan MySQL.
JDBC bertugas menyediakan koneksi ke database, sehingga kita bisa mengakses dan mengelola datanya dari program Java.
Ada beberapa istilah yang harus dipahami dalam JDBC:

- **DriverManager** : adalah sebuah *class* yang mengelola dirver;
- **Driver** **:** adalah interface yang menangani komunikasi dengan database.
- **Connection** : adalah *interface* yang menyediakan method untuk menghubungi database;
- **Statement** : adalah *inteface* untuk mengeksekusi *query*;
- **ResultSet** : adalah *interface* untuk menampung data hasil *query*.

Database

Koneksi

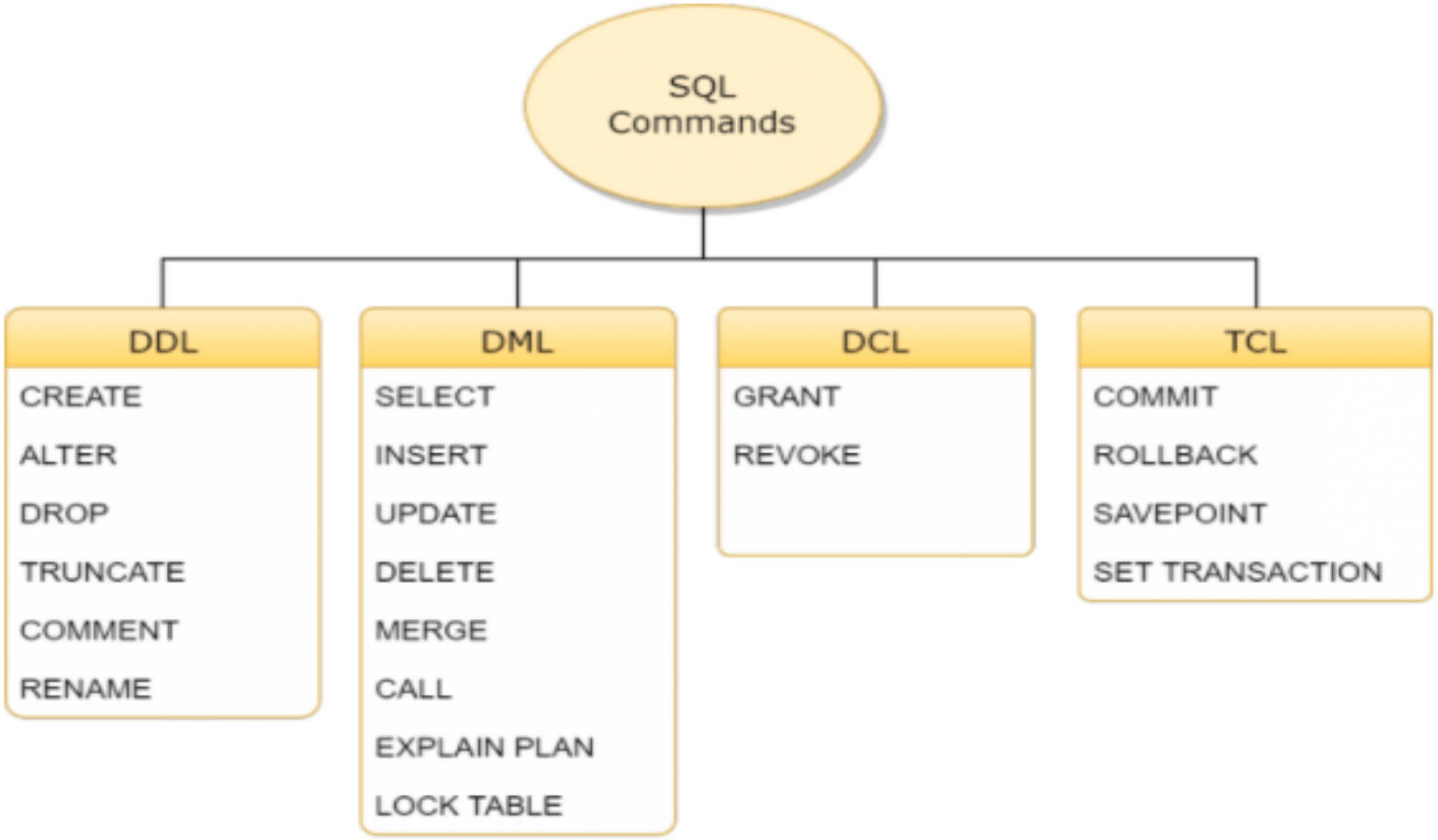
```
public class KoneksiDBMS
{
    String driverdbms = "com.mysql.jdbc.Driver";
    String database   = "jdbc:mysql://localhost/dbwulan5049";
    String user       = "wulan5049";
    String password   = "5049";

    public Connection BukaCn() throws SQLException
    {
        Connection condbms = null;
        try {
            Class.forName(driverdbms);
            condbms = DriverManager.getConnection(database,user,password);
            return condbms;
        }
        catch (Exception ex)
        {
            System.out.println("Koneksi database tidak berhasil !");
            return null;
        }
    }
}
```

Database

Pengertian SQL

SQL atau *Standard Query Language* adalah bahasa pemrograman yang digunakan dalam mengakses, mengubah, dan memanipulasi data yang berbasis relasional.



Database

DDL

DDL (*Data Definition Language*), yang berhubungan dengan skema dan deskripsi basis data, tentang bagaimana data harus berada dalam basis data.

- CREATE - untuk membuat database dan objeknya seperti (tabel, indeks, tampilan, prosedur penyimpanan, fungsi, dan trigger)
- ALTER - mengubah struktur database yang ada
- DROP - hapus objek dari database
- TRUNCATE - hapus semua catatan dari tabel, termasuk semua ruang yang dialokasikan untuk catatan dihapus.
- COMMENT - tambahkan komentar ke kamus data
- RENAME - mengganti nama objek

Database

DDL (*Data Definition Language*)
CREATE (membuat database)

Sintak penulisan :

```
CREATE DATABASE databasename;
```

Contoh :

```
CREATE DATABASE dbpenjualan;

CREATE DATABASE dbwulan5049;
```

Database

DDL (*Data Definition Language*)

Pernyataan DROP DATABASE digunakan untuk menghapus database SQL yang ada.

Sintak penulisan :

```
DROP DATABASE databasename;
```

Contoh :

```
DROP DATABASE dbpenjualan;

DROP DATABASE dbwulan5049;
```

Database

DDL (*Data Definition Language*)

CREATE (membuat tabel)

Sintak penulisan :

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

Contoh :

```
CREATE TABLE pegawai5049 (
    nip varchar(10),
    nama char(40),
    bagian varchar(20)
);
```

Database

DDL (*Data Definition Language*)

Pernyataan DROP table digunakan untuk menghapus tabel.

Sintak penulisan :

```
DROP TABLE table_name;
```

Contoh :

```
DROP TABLE pegawai5049;
```

Database

DML

DML (Manipulasi Data Language) yang berhubungan dengan manipulasi data dan termasuk pernyataan SQL yang paling umum seperti SELECT, INSERT, UPDATE, DELETE, dll, dan digunakan untuk menyimpan, memodifikasi, mengambil, menghapus dan memperbarui data dalam database.

- SELECT - mengambil data dari database
- INSERT - memasukkan data ke dalam tabel
- UPDATE - memperbarui data yang ada dalam tabel
- DELETE - Hapus semua record dari tabel database
- MERGE - operasi UPSERT (masukkan atau perbarui)
- CALL - memanggil subprogram PL/SQL atau Java
- LOCK TABLE - Kontrol konkurensi

Database

DML (Data Manipulation Language)

INSERT - memasukkan data ke dalam tabel

Sintak penulisan :

```
INSERT INTO nama_table (field1, field2, field3, ...)
       VALUES (nilai1, nilai2, nilai3, ...);
```

Contoh :

```
insert into pegawai5049(nip, nama, bagian) values
       ('"+fNip.getText() + "','"+fNama.getText()+"','"+fBagian.getText()+"');
```

Database

INSERT - memasukkan data ke dalam tabel dari program pegawai

```
void Add()
{
    try
    {
        KoneksiDBMS CnPenjualan = new KoneksiDBMS();
        Connection con = CnPenjualan.BukaCn();
        Statement stat = con.createStatement();

        String strsql = "insert into pegawai5049(nip, nama, bagian) " +
            " values ('"+fNip.getText() + "','"+fNama.getText()+"' " +
            ",'"+fBagian.getText()+"')";
        int stsproses = stat.executeUpdate(strsql);

        if (stsproses == 1)
            JOptionPane.showMessageDialog(this, "Sukses Di Tambahkan!!!");
        con.close();
    }
    catch(SQLException e) {
        JOptionPane.showMessageDialog(this, "Penambahan Gagal!!!");
    }
}
```

Database

DML (Data Manipulation Language)

UPDATE - memperbarui data yang ada dalam tabel

Sintak penulisan :

```
UPDATE nama_table
    SET field1 = nilai1, field2 = nilai2, ...
    WHERE kondisi;
```

Contoh :

```
update pegawai5049
    set nama = '"+fNama.getText()+"', bagian = '"+fBagian.getText()+"'
    where nip = '"+fNip.getText()+"';
```

Database

UPDATE - memperbarui data yang ada dalam tabel program pegawai

```
void Koreksi() {
    try {
        KoneksiDBMS CnPenjualan = new KoneksiDBMS();
        Connection con = CnPenjualan.BukaCn();
        Statement stat = con.createStatement();

        String strsql = "update pegawai5049 set nama='"+fNama.getText()+"' "+
                ",bagian='"+ fBagian.getText()+"' " +
                " where nip='"+fNip.getText()+"' ";
        int stsproses = stat.executeUpdate(strsql);

        con.close();
    }
    catch(SQLException e) {
        JOptionPane.showMessageDialog(this, "Koreksi Gagal !");
    }
}
```

Database

DML (Data Manipulation Language)
DELETE – Hapus record dari tabel
Sintak penulisan :

```
DELETE FROM nama_table  WHERE kondisi;
```

Contoh :

```
delete from pegawai5049 where nip = '"+fNip.getText()+"';
```

Database

DELETE – Hapus record data tabel program pegawai

```
void Hapus() {
    try
    {
        KoneksiDBMS CnPenjualan = new KoneksiDBMS();
        Connection con = CnPenjualan.BukaCn();
        Statement stat = con.createStatement();

        String strsql = "delete from pegawai5049 " +
            "where nip = '"+fNip.getText()+"' ";
        int stsproses = stat.executeUpdate(strsql);

        con.close();
    }
    catch(SQLException e) {
        JOptionPane.showMessageDialog(this, "Penghapusan Gagal!!!");
    }
}
```

Database

DML (Data Manipulation Language)

SELECT – mengambil data dari tabel database

Sintak penulisan :

```
SELECT *
    FROM table_name
    WHERE condition;
```

```
SELECT column1, column2, column3, ...
    FROM table_name
    WHERE condition;
```

Contoh :

```
select * from pegawai5049;
```

```
SELECT *
    FROM pegawai5049
    WHERE nip ='" + fNip.getText () + "';
```

Database

SELECT – mengambil data dari tabel database program pegawai

```
void tampiltabel() {
    try {
        KoneksiDBMS CnPenjualan = new KoneksiDBMS();
        Connection con = CnPenjualan.BukaCn();
        Statement stat = con.createStatement();

        String strsql = "select * from pegawai5049";
        rs = stat.executeQuery(strsql);
        ResultSetMetaData meta = rs.getMetaData();
        rs.beforeFirst();
        while(rs.next()) {
            String Nip = rs.getString("nip");
            String Nama = rs.getString("nama");
            String Bagian = rs.getString("bagian");
            String[] data = {Nip, Nama, Bagian};
            tabModel.addRow(data);
        }
        stat.close(); rs.close();
        con.close();
    } catch (SQLException se) {}
}
```

RANGKUMAN

Database merupakah tempat dimana data dapat dikelola sehubungan dengan Input, koreksi, hapus, serta penampilan data.

Dengan menggunakan database, data dapat dikelola sesuai dengan kebutuhan untuk menyajikan informasi.

SUMBER PUSTAKA

- https://www.w3schools.in/mysql/ddl-dml-dcl/
- https://ducmanhphan.github.io/2020-01-09-How-to-use-JDBC-to-connect-database-in-Java-project/
- https://www.studytonight.com/java/connecting-to-mysql.php
- https://ngodingdata.com/memahami-perintah-dml-mysql/
- Sumber gambar:
  www.freepik.com

THANKS

ANY QUESTIONS?