

Pemrograman Database Java

Java Database Connectivity
ODBC
Native Driver

Endy Muhardin
<endy@artivisi.com>

Materi

- Koneksi dengan ODBC
- Koneksi dengan Native Driver
- Koneksi dengan Hibernate

Koneksi Database

- Jenis – jenis koneksi :
 - ODBC
 - Native Driver
- Langkah – langkah koneksi :
 - Data Source
 - Instalasi Driver
 - Connection String
 - Buat koneksi ke Database

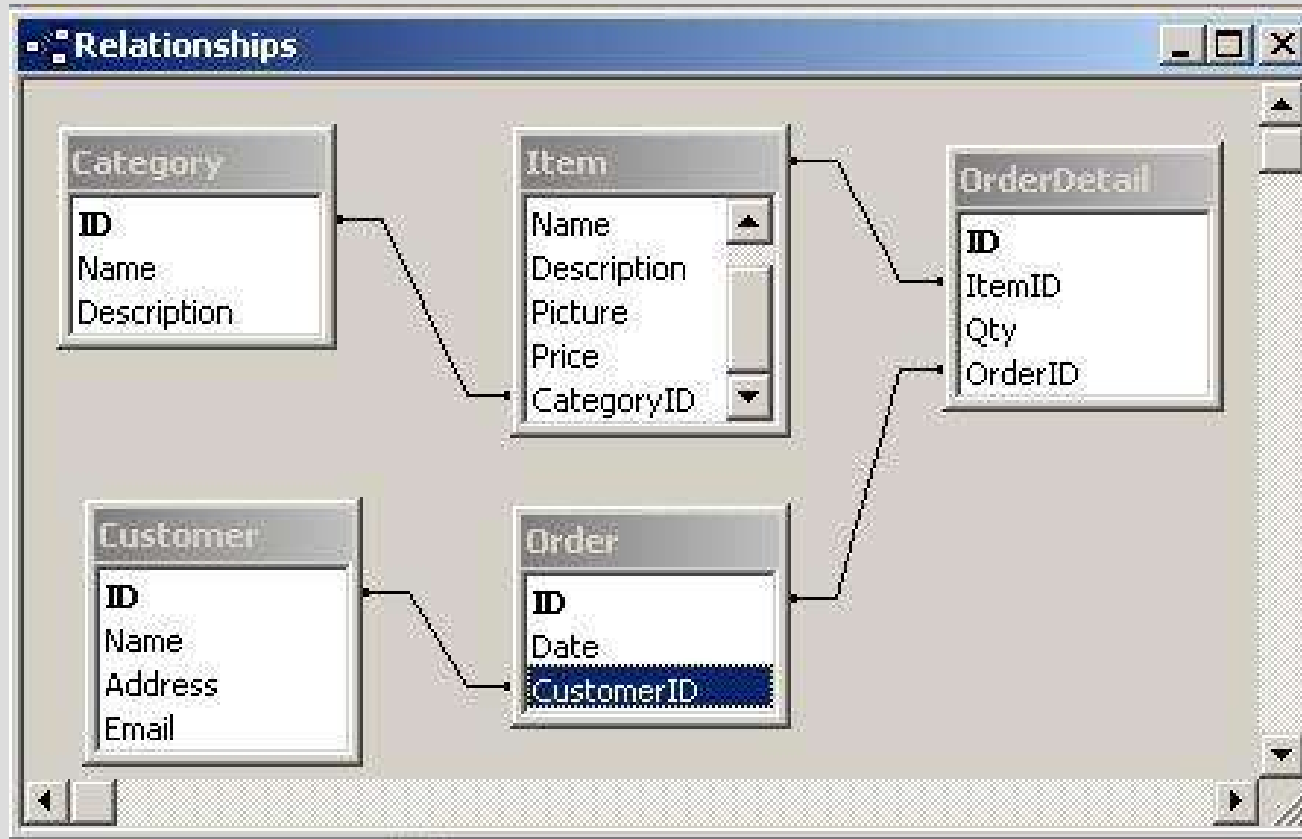
Koneksi ODBC

- Buat Database
- Konfigurasi ODBC Data Source
- Panggil dengan JDBC-ODBC Driver

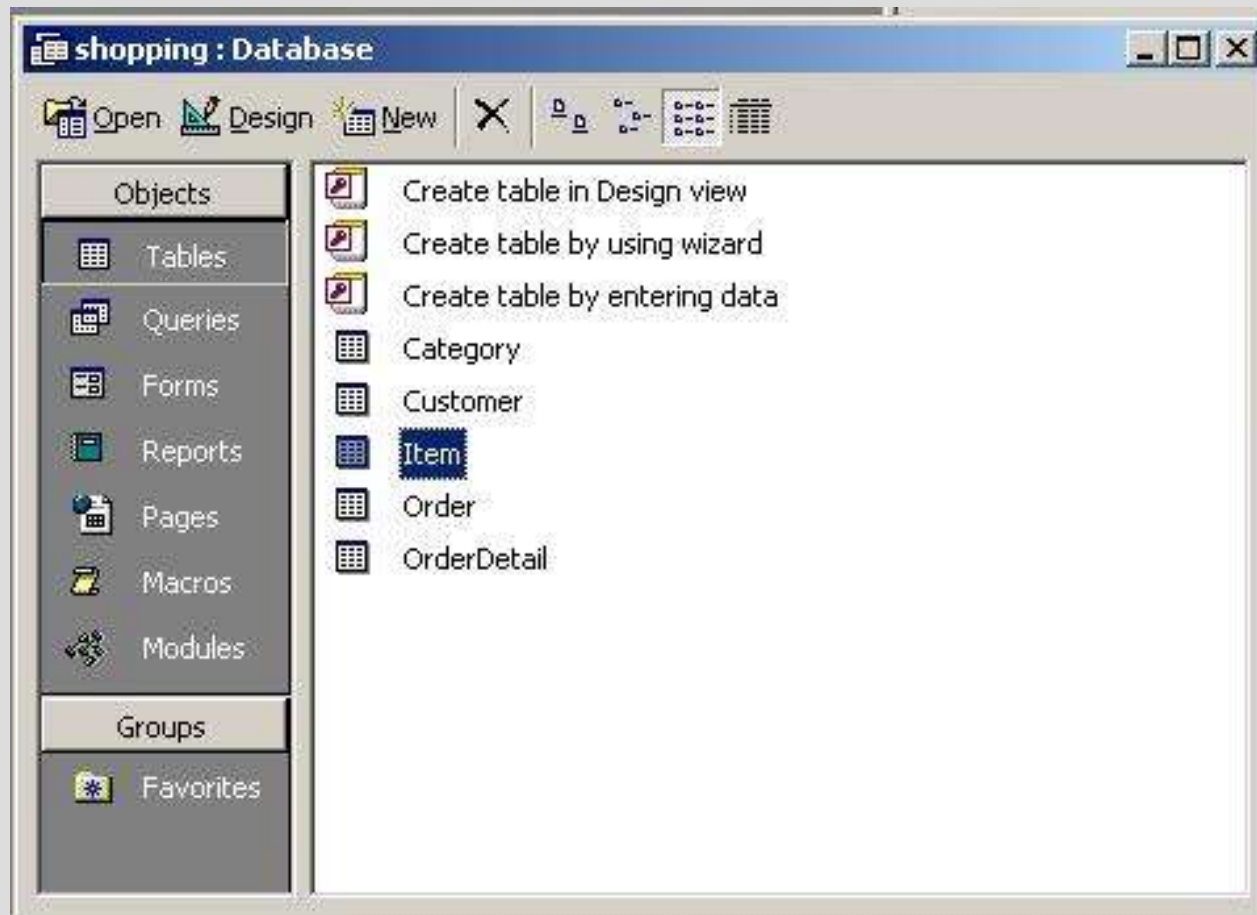
Membuat Database

- Database dapat dibuat dengan berbagai database engine
 - Microsoft Access
 - MS SQL Server
 - Oracle
 - Dsb
- Contoh kasus : shopping.mdb (MS Access)

Struktur Database



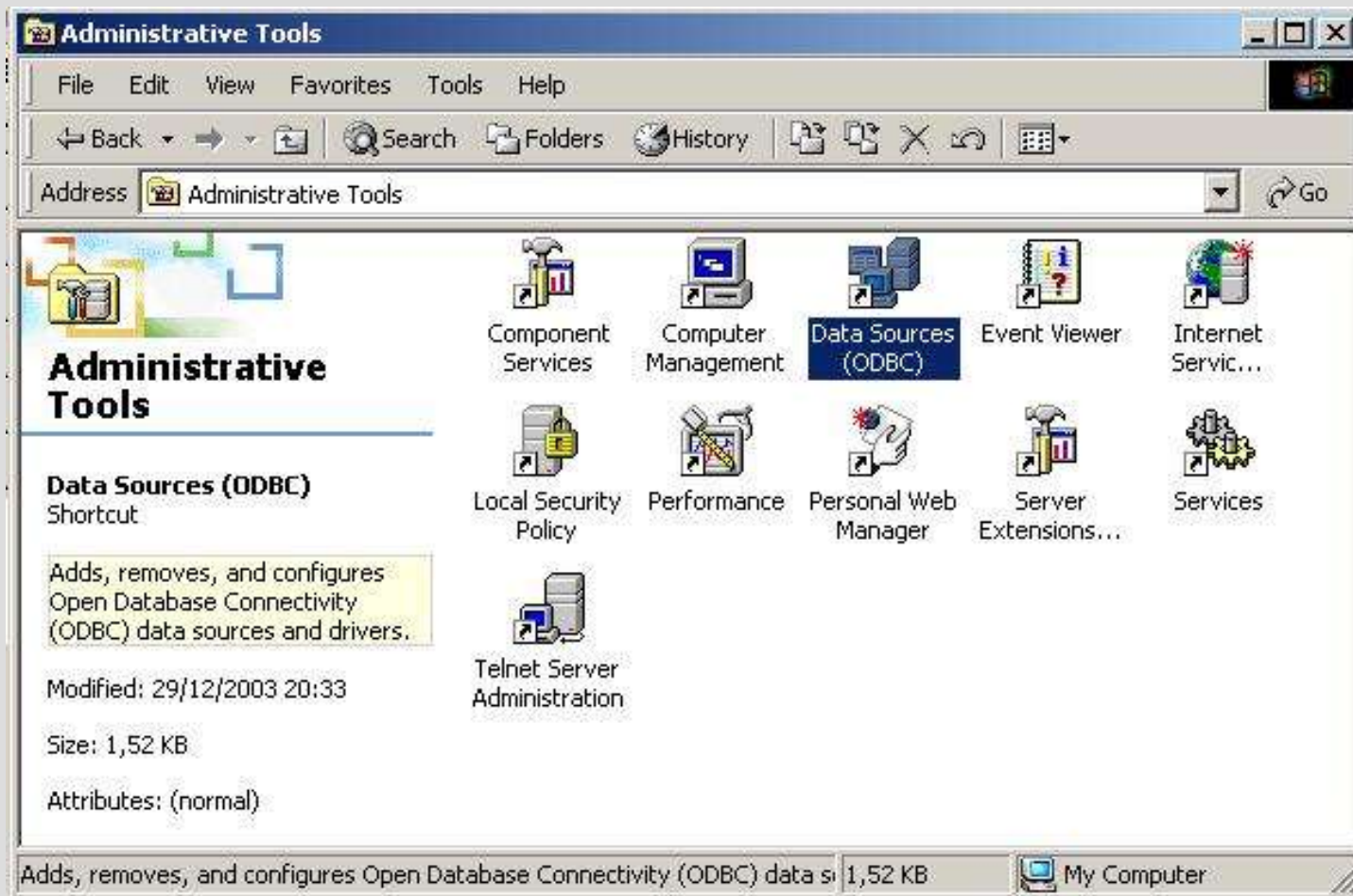
Microsoft Access



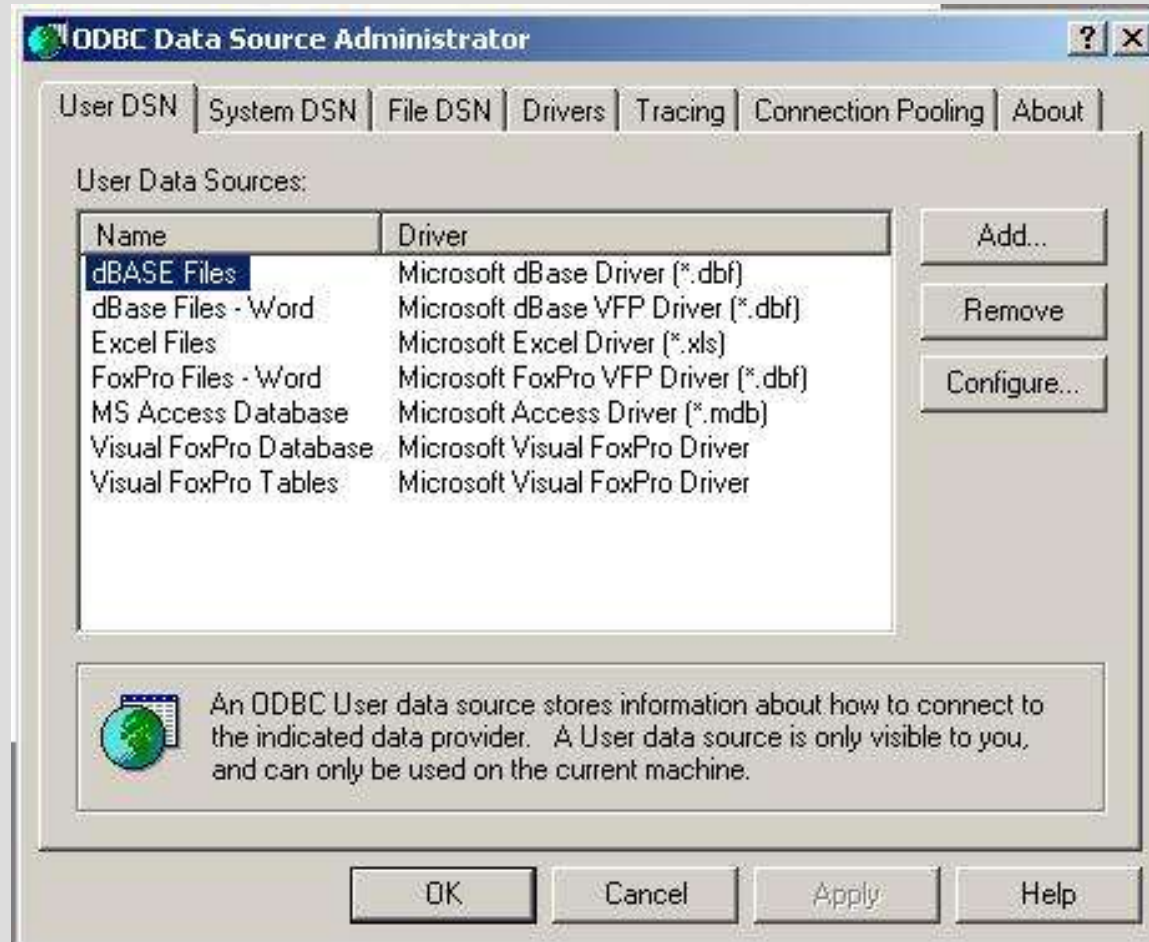
ODBC DataSource

- Buka Control Panel
- Masuk ke menu DataSource (ODBC)
- Klik tombol Add
- Pilih Driver
- Beri Nama Datasource
- Pilih Database
- Klik OK

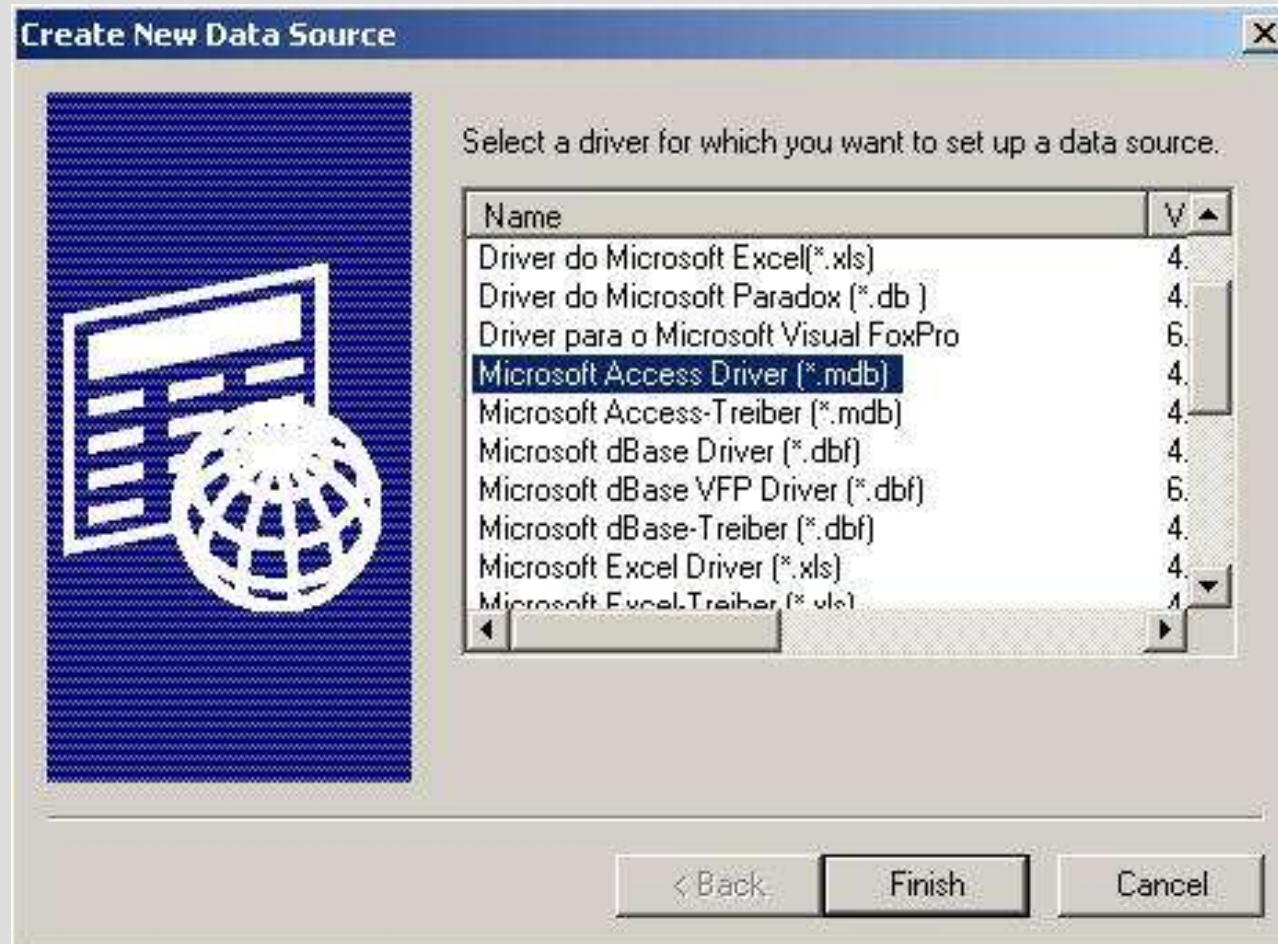
Buka Control Panel



DataSource Administrator



Menambah DataSource MS Access



DataSource Setup

ODBC Microsoft Access Setup

Data Source Name: shop

Description:

Database:

Database:

Select... Create... Repair... Compact...

System Database

None

Database:

System Database...

OK

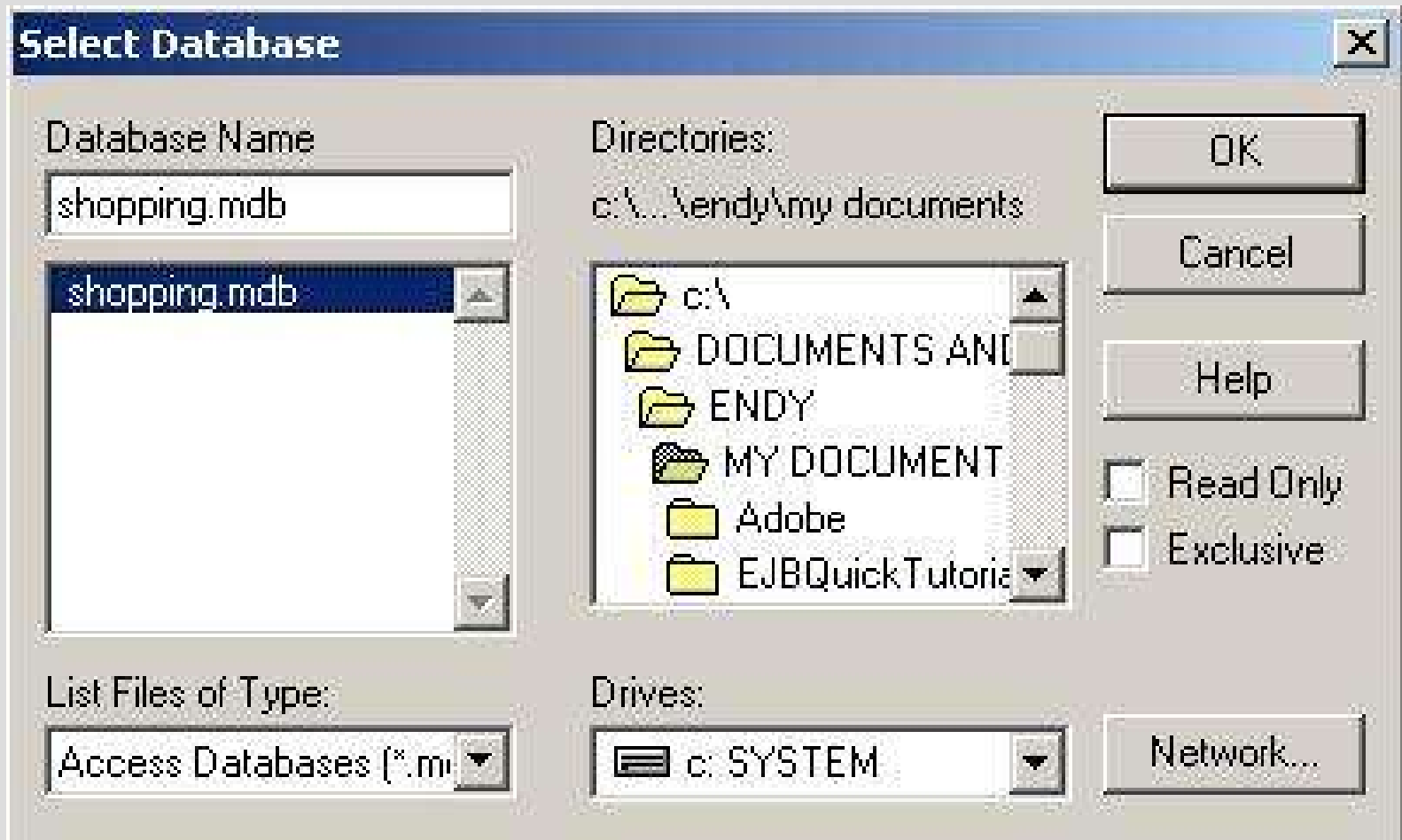
Cancel

Help

Advanced...

Options>>

Pilih Database



Kode Program ODBC

- Instalasi ODBC Driver
 - `Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");`
- Buat Connection String
 - `String cs = "jdbc:odbc:nama-datasource";`
- Buat koneksi ke database
 - `Connection cn = DriverManager.getConnection(cs);`
- Buat query
 - `String q = "SELECT * FROM Customer";`
 - `Statement stm = cn.createStatement();`
- Eksekusi
 - `ResultSet rs = stm.executeQuery(q);`

ODBCDemo.java

```
import java.sql.*;
public class ODBCDemo
{
    public static void main(String[] args)
    {
        try {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            String cs = "jdbc:odbc:shop";
            Connection cn = DriverManager.getConnection(cs);
            String qry = "SELECT * FROM Customer";
            Statement stm = cn.createStatement();
            ResultSet rs = stm.executeQuery(qry);
            while(rs.next()) {
                System.out.println("Nama : "+rs.getString("Name"));
                System.out.println("Email : "+rs.getString("Email"));
                System.out.println("Alamat : "+rs.getString("Address"));
            }
        } catch (Exception err) {err.printStackTrace();}
    }
}
```

Koneksi dengan Native Driver

- Download Native Driver
- Lihat Dokumentasi
- Instal Driver ke dalam Classpath
- Buat koneksi ke database

Native Driver MySQL

- Native Driver MySQL dapat didownload di <http://www.mysql.com>, bagian ConnectorJ
- Dalam tarball yang didownload, kita dapat menemukan:
 - Dokumentasi
 - `mysql-connector-java-3.0.8-stable-bin.jar`
- `mysql-connector-java-3.0.8-stable-bin.jar` harus diletakkan dalam CLASSPATH

Perbedaan ODBC dan Native

- Di Java, teknik koneksi database dengan ODBC dan Native Driver secara konseptual tidak berbeda.
- Untuk migrasi dari ODBC ke Native JDBC, baris kode yang perlu diganti adalah :
 - Instalasi Driver JDBC
 - Connection String

Native MySQL Demo

```
import java.sql.*;
public class MySQLDemo
{
    public static void main(String[] args)
    {
        try {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            String cs = "jdbc:mysql://localhost:3306/shop?user=test&password=test";
            Connection cn = DriverManager.getConnection(cs);
            String qry = "SELECT * FROM Customer";
            Statement stm = cn.createStatement();
            ResultSet rs = stm.executeQuery(qry);
            while(rs.next()) {
                System.out.println("Nama : "+rs.getString("Name"));
                System.out.println("Email : "+rs.getString("Email"));
                System.out.println("Alamat : "+rs.getString("Address"));
            }
        } catch (Exception err) {err.printStackTrace();}
    }
}
```

Query JDBC

- Dalam JDBC, query dibedakan menjadi dua jenis :
 - Query yang menghasilkan ResultSet (yaitu query SELECT)
 - Query yang tidak menghasilkan ResultSet (yaitu query INSERT, UPDATE, DELETE, CREATE, dsb)

Query Insert

```
import java.sql.*;
public class MySQLDemo
{
    public static void main(String[] args)
    {
        try {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            String cs = "jdbc:mysql://localhost:3306/shop?user=test&password=test";
            Connection cn = DriverManager.getConnection(cs);
            String qry = "INSERT INTO Customer ";
            qry += "VALUES (null, 'Endy', 'Surabaya', 'endy@artivisi.com)";
            Statement stm = cn.createStatement();
            int result = stm.executeUpdate(qry);
            if (result == 0) {
                System.out.println("Insert Gagal");
            } else {
                System.out.println("Insert Sukses");
            }
        } catch (Exception err) {err.printStackTrace();}
    }
}
```