PRAKTIKUM PEMOGRAMA
BERORIENTASI OBYEK LANJUTAN
KELAS C

DISUSUN OLEH :

Nama : Johanes Yogtan Wicaksono Raharja
NIM : 215314105

TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA YOGYAKARTA
2022

1. Kerjakan potongan program berikut ini. Tujuan latihan ini adalah untuk menguji koneksi yang anda buat. Sesuaikan data koneksi dengan database yang akan anda gunakan.

Screenshot Program :

```
package KoneksiDatabse;

import java.sql.*;
public class DataHandler1 {
    // string 'localhost' anda ganti dengan IP server yang

    private String jdbcURL = "jdbc:oracle:thin:@//localhost:1521";//ssid
    //private String jdbcURL = "jdbc:oracle:thin:@//172.23.9.185:1521/orcl";//ser
    private String user = "hr";
    private String password = "sysl303";
    Connection conn;

    public DataHandler1() {
    }
     public void getDBConnection() {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            conn = DriverManager.getConnection(jdbcURL, user, password);

            System.out.println("Koneksi berhasil");

        } catch (Exception ex) {
            System.out.println("Masih belum konek");
        }
    }
    public void close() {
        try {
            conn.close();
        } catch (SQLException ex) {
            System.out.println("Tidak bisa tutup koneksi");
        }
    }
    public static void main(String[] args) throws SQLException {
        DataHandler1 datahandler = new DataHandler1();
        datahandler.getDBConnection();
        datahandler.close();

    }
}
```

Screenshot Output

```
Output
) #5 × JavaApplication1 (run) #6 × JavaApplication1 (run) #7 ×
run:
Koneksi berhasil
BUILD SUCCESSFUL (total time: 1 second)
```

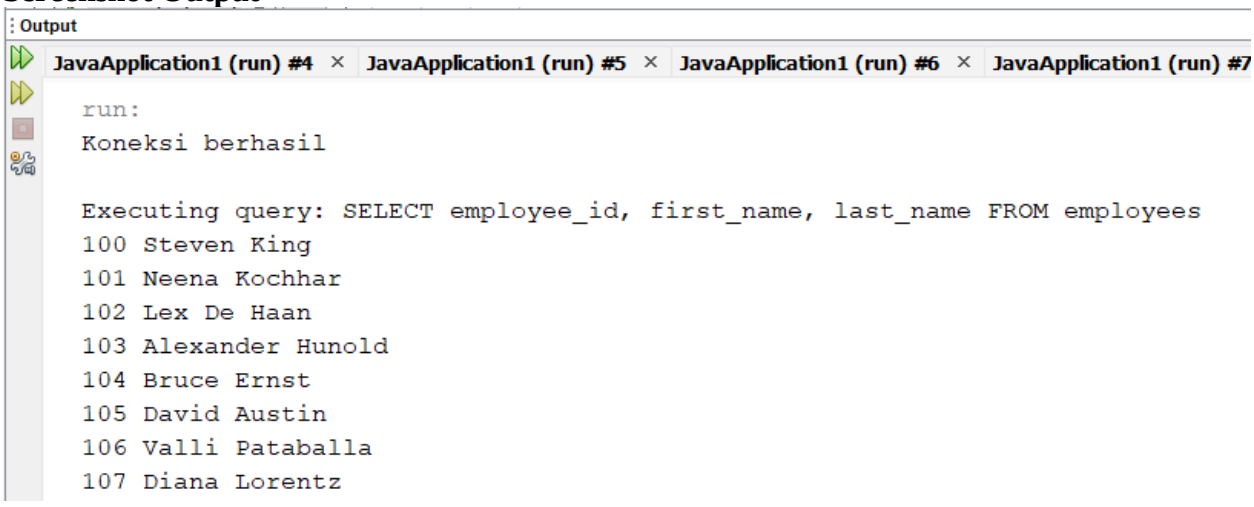
2. Kerjakan potongan program berikut ini. Tujuan latihan ini adalah untuk menampilkan isi tabel employees

Screenshot Program :

```
package KoneksiDatabse;
import java.sql.*;
import java.util.ArrayList;

public class DataEmployee2 {

    Connection conn;
    Statement stmt;
    ResultSet rset;
    String query;

    public DataEmployee2() throws SQLException {
        DataHandler1 dataHandler = new DataHandler1();
        dataHandler.getDBConnection();
        //buat koneksi
        conn = dataHandler.conn;
    }

    public ResultSet getAllEmployees() throws SQLException {
        stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                ResultSet.CONCUR_READ_ONLY);
        query = "SELECT employee_id, first_name, last_name"+
" FROM employees ";
        System.out.println("\nExecuting query: " + query);
        rset = stmt.executeQuery(query);
        return rset;
    }
```

```
    public ArrayList searchEmployee(String keyword) throws
            SQLException {
        ArrayList result = null;
        // stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
       // ResultSet.CONCUR_READ_ONLY);
String query = "SELECT * FROM employee e, department d "
                + "where e.department_id=d.department_id and "
                + "d.department_name like '" + keyword + "'";
        rset = stmt.executeQuery(query);
        result = new ArrayList();
        while (rset.next()) {
            Employee3 temp = new Employee3(rset.getInt(1), rset.getString(2),
                    rset.getString(3), rset.getInt(4));
            result.add(temp);
        }
        conn.close();
        return result;
    }
    public static void main(String[] args) throws SQLException {
        DataEmployee2 le = new DataEmployee2();
        ResultSet rset = le.getAllEmployees();
        String keyword = null;
//         le.searchEmployee(keyword);

        while (rset.next()) {
            System.out.println(rset.getInt(1) + " " + rset.getString(2)
                    + " " + rset.getString(3));
        }
        le.conn.close();
    }
}
```

Screenshot Output



(Karena terlalu banyak jadi sebagian saja, ini sebagai bukti berhasilnya program)

3. Buatlah kelas Employee.java

Screenshot Program :

```
package KoneksiDatabse;
public class Employee3 {
 private int employee_id;
 private String first_name;
 private String last_name;
 private int department_id;
 public Employee3()
 {
 }
 public Employee3(int id,String first,String last, int dep_id)
 {
 this.employee_id=id;
 this.first_name=first;
 this.last_name=last;
 this.department_id=dep_id;
 }

    public int getEmployee_id() {
        return employee_id;
    }
    public void setEmployee_id(int employee_id) {
        this.employee_id = employee_id;
    }
    public String getFirst_name() {
        return first_name;
    }
    public void setFirst_name(String first_name) {
        this.first_name = first_name;
    }
    public String getLast_name() {
        return last_name;
    }
    public void setLast_name(String last_name) {
        this.last_name = last_name;
    }
    public int getDepartment_id() {
        return department_id;
    }
    public void setDepartment_id(int department_id) {
        this.department_id = department_id;
    }
}
```

4. Tambahkan method searchEmployee di kelas DataEmployee

Screenshot Program :

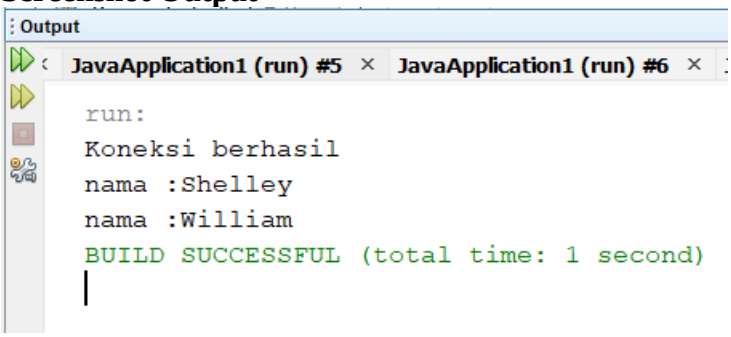
```
public ArrayList searchEmployee(String keyword) throws SQLException {
    ArrayList result = null;
    stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
    //Perintah Query untuk mencari pegawai
    String query = "SELECT * FROM Employees e, departments d "
            + "where e.department_id=d.department_id and "
            + "d.department_name like '" + keyword + "'";
    rset = stmt.executeQuery(query);
    result = new ArrayList();
    while (rset.next()) {
        Employee3 temp = new Employee3(rset.getInt(1), rset.getString(2),
                rset.getString(3), rset.getInt(11));
        result.add(temp);
    }
    conn.close();
    return result;
}
```

5. Buatlah kelas EmployeeTableModel

Screenshot Program :

```
package KoneksiDatabse;
import java.sql.SQLException;
import java.util.ArrayList;
import javax.swing.table.AbstractTableModel;

public class EmployeeTableModel extends AbstractTableModel {
    String columNames[] = {"Id", "First Name", "Last Name", "Department Id"};
    ArrayList data;

    public EmployeeTableModel(String keyword) throws SQLException {
        DataEmployee2 r = new DataEmployee2();
        data = r.searchEmployee(keyword);
    }
    public int getRowCount() {
        return data.size();
    }
    public int getColumnCount() {
        return columNames.length;
    }
    @Override
    public String getColumnName(int col) {
        return columNames[col];
    }
    public Object getValueAt(int rowIndex, int columnIndex) {
        Employee3 temp = (Employee3) data.get(rowIndex);
        switch (columnIndex) {
            case 0:
                return temp.getEmployee_id();
            case 1:
                return temp.getFirst_name();
            case 2:
                return temp.getLast_name();
            case 3:
                return temp.getDepartment_id();
        }
        return null;
    }
    @Override
    public Class getColumnClass(int c) {
        System.out.println(c);
        return getValueAt(0, c).getClass();
    }
    public static void main(String[] args) throws SQLException {
        EmployeeTableModel test = new EmployeeTableModel("Accounting");
        for (int i = 0; i < test.getRowCount(); i++) {
            System.out.println("nama :" + test.getValueAt(i, 1));
        }
    }
}
```

Screenshot Output

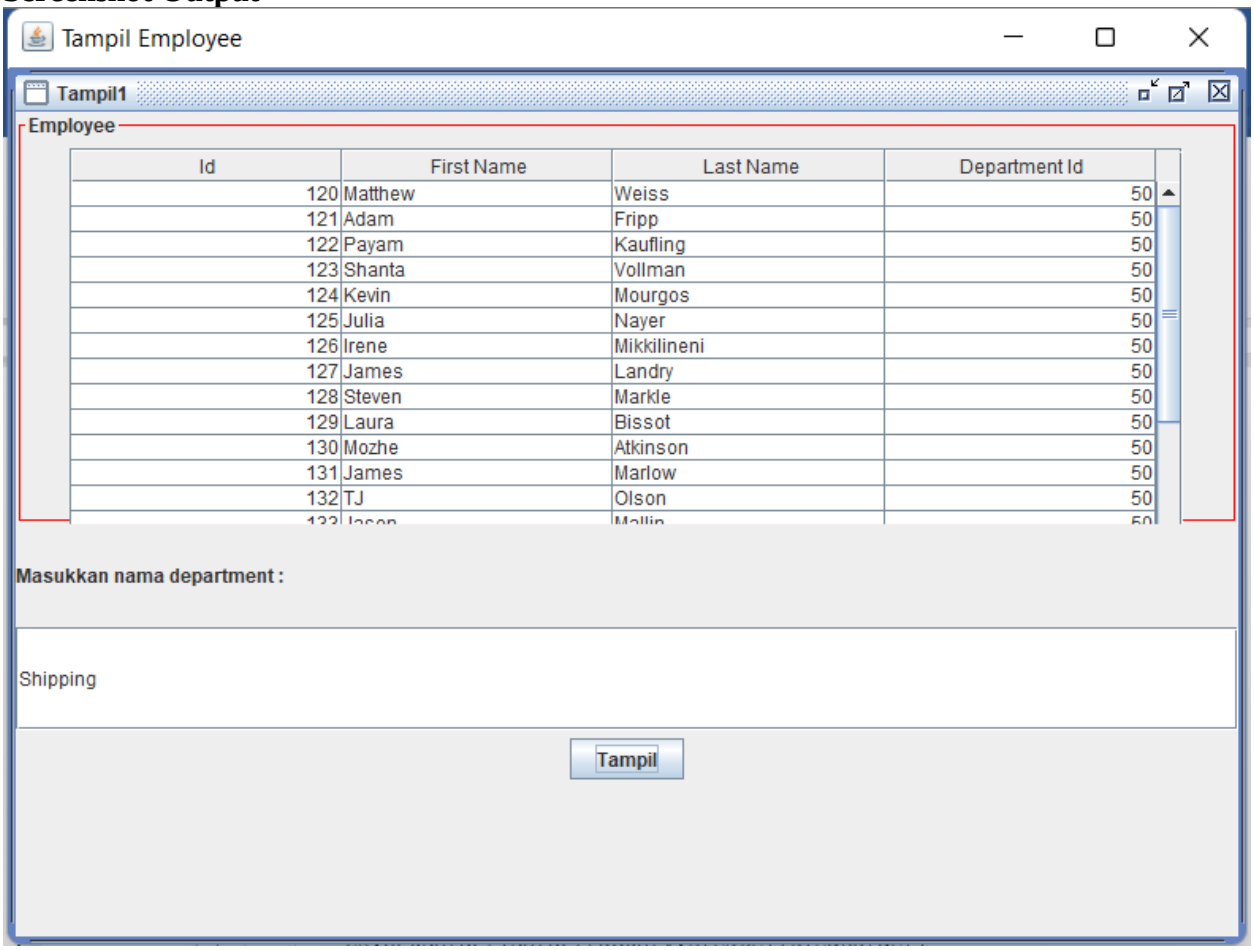


6. Program berikut ini merupakan sebuah JInternalFrame. Silahkan anda membuat JFrame yang salah satu menunya memanggil JInternalFrame tersebut. Tujuan latihan ini adalah untuk menampilkan tabel hasil pencarian

Screenshot Program :

```
package KoneksiDatabse;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;

public class JIFEmp extends javax.swing.JInternalFrame implements ActionListener

    static int openFrameCount = 0;
    static final int xOffset = 30, yOffset = 30;
    JPanel pa, pb, p1, p2;
    JLabel label1;
    JButton but1;
    JTextField text1;
    JTable jtabel1;

    public JIFEmp(String title) {
        super(title + (++openFrameCount),
                true, //resizable
                true, //closable
                true, //maximizable
                true);//iconifiable
        setSize(800, 600);
        addPanel();
        //Set the window's location.
        setLocation(xOffset * openFrameCount,
                yOffset * openFrameCount);
    }

    public void addPanel() {
        pa = new JPanel();
        pa.setBorder(BorderFactory.createTitledBorder(
                BorderFactory.createLineBorder(Color.red), "Employee"));
        jtabel1 = new JTable();
        jtabel1.setPreferredScrollableViewportSize(new Dimension(700, 350));
        JScrollPane scrollPane = new JScrollPane(jtabel1);
```

Screenshot Output



7. Tambahkan method insertEmployee(int id,String first,String last, int dep_id), updateEmployee(int employee_id, int department_id), deleteEmployee(String first_name) pada kelas DataEmployee dan ubah method main di kelas tersebut untuk menjalankan tiga perintah tersebut.

Screenshot Program method insertEmployee(int id,String first,String last, int dep_id), updateEmployee(int employee_id, int department_id), deleteEmployee(String first_name)

```
static void insertEmployee(int id, String first, String last, int dep_id)
        throws SQLException {
    String query = "INSERT INTO praktikumEmployee (employee_id, first_name,"
            + " last_name, department_id) Values (" + id + ", '"
            + first + "', '" + last + "', " + dep_id + ")";
    int hasil = stmt.executeUpdate(query);

}

static void deleteEmployee(String first) throws SQLException {
    String query = "Delete from praktikumEmployee where first_name is"
            + " '"+first+"'";
    int hasil = stmt.executeUpdate(query);

}

static void updateEmployee(int id, int dep_id ) throws SQLException {
    String query = "Update praktikumEmployee set employee_id = "+id+" "
            + "where department_id = "+dep_id;
    int hasil = stmt.executeUpdate(query);
}

static void insertDepartment(int id, String first, String last) throws
        SQLException {
    String query = "INSERT INTO praktikumDepartment (idDepartment, name, "
            + "city) Values (" + id + ", '" + first + "', '" + last + "')";
    int hasil = stmt.executeUpdate(query);

}
```

Screenshot Program Generate Data Employee

```
public ResultSet getAllEmployees() throws SQLException {
    stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet
            .CONCUR_READ_ONLY);
    query = "SELECT employee_id, first_name, last_name, department_id"
            + " FROM praktikumEmployee ";
    System.out.println("\nExecuting query: " + query);
    rset = stmt.executeQuery(query);
    return rset;
}
```

```
    DataEmployee2 le = new DataEmployee2();
    String queryStr = "CREATE TABLE praktikumEmployee"
            + "(employee_id INT NOT NULL PRIMARY KEY"
            + ",first_name VARCHAR(45) NULL"
            + ",last_name VARCHAR(45) NULL"
            + ",department_id INT NULL)";
    System.out.println("\nExecuting query: " + queryStr);
    stmt.executeUpdate(queryStr);

    insertEmployee(25348, "Matthew", "Smith",3);
    insertEmployee(10102, "Ann", "Jones",3);
    insertEmployee(18316, "John", "Barrimore",1);
    insertEmployee(29346, "James", "James",2);
    insertEmployee(9031, "Elsa", "Bertoni",2);
    insertEmployee(2581, "Elke", "Hansel",2);
    insertEmployee(28559, "Sybill", "Moser",1);
```

Screenshot Program Generate Data Department

```
public ResultSet getAllDepartments() throws SQLException {
    stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
    query = "SELECT idDepartment, name, city"
            + " FROM praktikumDepartment ";
    System.out.println("\nExecuting query: " + query);
    rset = stmt.executeQuery(query);
    return rset;
}
```

```
    String kueryStr = "CREATE TABLE praktikumDepartment"
            + "(idDepartment INT NOT NULL PRIMARY KEY"
            + ",name VARCHAR(45) NULL"
            + ",city VARCHAR(45) NULL)";
    System.out.println("\nExecuting query: " + kueryStr);
    stmt.executeUpdate(kueryStr);
    insertDepartment(1, "Research", "Dallas");
    insertDepartment(2, "Accounting", "Seattle");
    insertDepartment(3, "Marketing", "Dallas");
    rset = le.getAllEmployees();
```

Screenshot Output Data Employee dan Data Department