ACTIVITY PERTEMUAN 3

NAMA **: Leonhard Dominikus Adiarsa Fernandez**

NPM **: 50421746**

KELAS **: 3IA14**

MATERI **:**

MATA PRAKTIKUM **: RPL2**

ModelMahasiswa.java :

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package me.leo.mahasiswa.model;

/**
 *
 * @author totos
 */
public class ModelMahasiswa {
    private int id;
    private String nama;
    private String npm;
    private int semester;
    private float ipk;

    public ModelMahasiswa(int id, String nama, String npm, int semester, float ipk) {
        this.id = id;
        this.nama = nama;
        this.npm = npm;
        this.semester = semester;
        this.ipk = ipk;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getNpm() {
        return npm;
    }

    public void setNpm(String npm) {
```

```
        this.npm = npm;
    }

    public int getSemester() {
        return semester;
    }

    public void setSemester(int semester) {
        this.semester = semester;
    }

    public float getIpk() {
        return ipk;
    }

    public void setIpk(float ipk) {
        this.ipk = ipk;
    }


}
```

MahasiswaDAO.java :

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package me.leo.mahasiswa.model;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;
/**
 *
 * @author totos
 */
public class MahasiswaDAO {
    private Connection connection;

    public MahasiswaDAO() {
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/mvc_50421746", "root", "");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public boolean checkConnection() {
        try {
            if(connection != null && !connection.isClosed()){
                return true;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }

    public void closeConnection() {
        try {
```

```
        if(connection != null) {
            connection.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public void addMahasiswa(ModelMahasiswa mahasiswa) {
        String sql = "INSERT INTO mahasiswa (npm, nama, semester, ipk) VALUES (?, ?, ?, ?)";
    try {
            PreparedStatement pstmt = connection.prepareStatement(sql);
        pstmt.setString(1, mahasiswa.getNpm());
        pstmt.setString(2, mahasiswa.getNama());
        pstmt.setInt(3, mahasiswa.getSemester());
        pstmt.setFloat(4, mahasiswa.getIpk());
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public List<ModelMahasiswa> getAllMahasiswa(){
    List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
    String sql = "SELECT * FROM mahasiswa";
    try{
        Statement stmt = connection.createStatement();
        ResultSet rs = stmt.executeQuery(sql);
        while(rs.next()){
            mahasiswaList.add(new ModelMahasiswa(
                rs.getInt("id"),
                rs.getString("npm"),
                rs.getString("nama"),
                rs.getInt("semester"),
                rs.getFloat("ipk")
            ));
        }
    } catch(SQLException e) {
        e.printStackTrace();
    }
    return mahasiswaList;
}

public void updateMahasiswa(ModelMahasiswa mahasiswa){
    String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?";
    try{
        PreparedStatement pstmt = connection.prepareStatement(sql);
        pstmt.setString(1, mahasiswa.getNpm());
        pstmt.setString(2, mahasiswa.getNama());
        pstmt.setInt(3, mahasiswa.getSemester());
        pstmt.setFloat(4, mahasiswa.getIpk());
        pstmt.setInt(5, mahasiswa.getId());
    } catch(SQLException e) {
        e.printStackTrace();
    }
}

public void deleteMahasiswa(int id){
    String sql = "DELETE FROM mahasiswa WHERE id = ?";
    try{
        PreparedStatement pstmt = connection.prepareStatement(sql);
```

```
            pstmt.setInt(1, id);
            pstmt.executeUpdate();
        } catch(SQLException e){
            e.printStackTrace();
        }
    }
}
```

MahasiswaController.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package me.leo.mahasiswa.controller;

import me.leo.mahasiswa.model.MahasiswaDAO;
import me.leo.mahasiswa.model.ModelMahasiswa;
import java.util.List;
/**
 *
 * @author totos
 */
public class MahasiswaController {
    private MahasiswaDAO mahasiswaDAO;

    public MahasiswaController(MahasiswaDAO mahasiswaDAO) {
        this.mahasiswaDAO = mahasiswaDAO;
    }

    public void displayMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
        if(mahasiswaList.isEmpty()) {
            System.out.println("Tidak ada data mahasiswa");
        } else {
            System.out.println("");
            System.out.println("============================");
            for(ModelMahasiswa m: mahasiswaList){
                System.out.println("ID          : " + m.getId());
                System.out.println("NPM         : " + m.getNpm());
                System.out.println("NAMA        : " + m.getNama());
                System.out.println("SEMESTER    : " + m.getSemester());
                System.out.println("IPK         : " + m.getIpk());
                System.out.println("============================");
            }
        }
    }

    public void displayMessage(String message) {
        System.out.println(message);
    }

    public void checkDatabaseConnection() {
        boolean isConnected = mahasiswaDAO.checkConnection();
        if(isConnected){
            displayMessage("Koneksi ke db berhasil");
        } else {
            displayMessage("Koneksi DB Gagal");
        }
    }

    public void displayAllMahaiswa() {
        List<ModelMahasiswa> mahasiswaList = mahasiswaDAO.getAllMahasiswa();
```

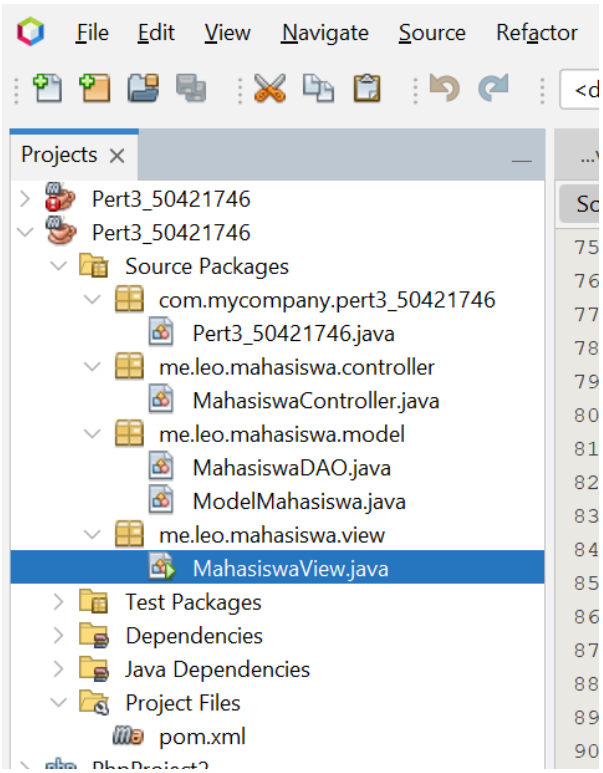
```
        displayMahasiswaList(mahasiswaList);
    }

    public void addMahasiswa(String npm, String nama, int semester, float ipk) {
        ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(0, npm, nama, semester, ipk);
        System.out.println("Controller Data:    " + npm + nama + semester + ipk);
        System.out.println(mahasiswaBaru);
        mahasiswaDAO.addMahasiswa(mahasiswaBaru);
        displayMessage("Mahasiswa berhasil ditambahkan!");
    }

    public void updateMahasiswa(int id, String npm, String nama, int semester, float ipk){
        ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id, npm, nama, semester, ipk);
        mahasiswaDAO.updateMahasiswa(mahasiswaBaru);
        displayMessage("Mahasiswaberhasil diperbarui!");
    }

    public void deleteMahasiswa(int id){
        mahasiswaDAO.deleteMahasiswa(id);
        displayMessage("Mahasiswa Berhasil Dihapus!");
    }

    public void closeConnection() {
        mahasiswaDAO.closeConnection();
    }
}
```

MahasiswaView.java

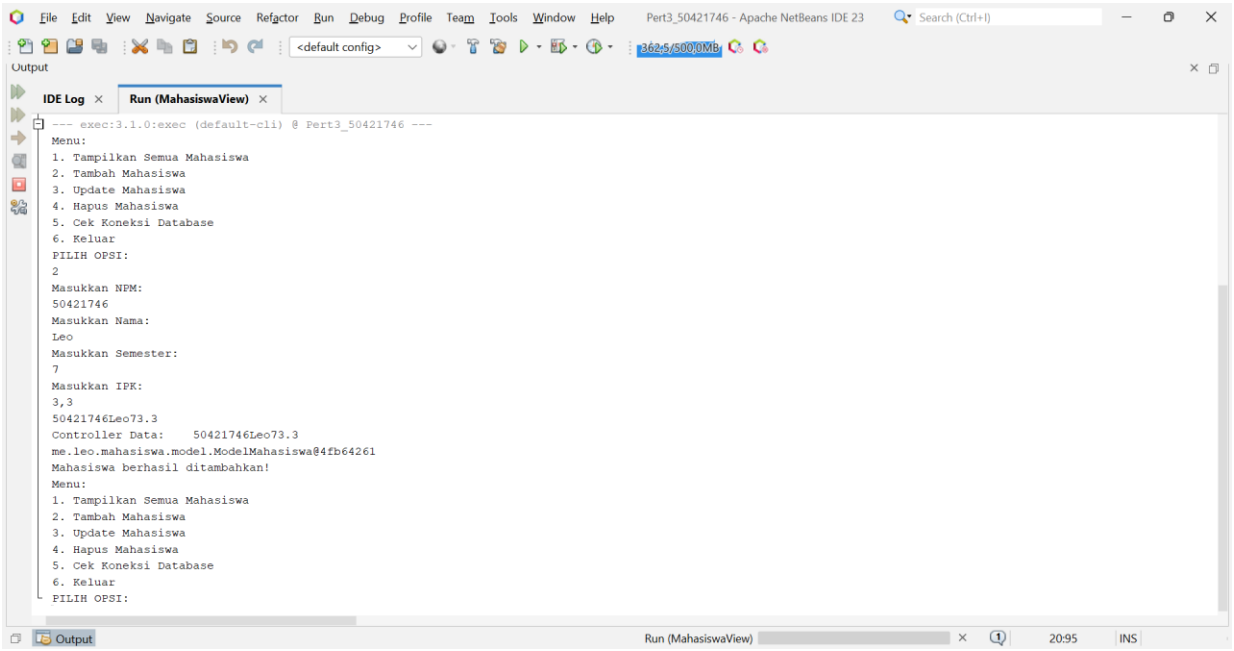
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package me.leo.mahasiswa.view;

import me.leo.mahasiswa.controller.MahasiswaController;
import me.leo.mahasiswa.model.MahasiswaDAO;
import java.util.Scanner;
/**
 *
 * @author totos
 */
public class MahasiswaView {
    public static void main(String[] args) {
        MahasiswaDAO mahasiswaDAO = new MahasiswaDAO();
        MahasiswaController mahasiswaController = new MahasiswaController(mahasiswaDAO);

        Scanner scanner = new Scanner(System.in);
        int pilihan;

        while(true) {
            System.out.println("Menu:");
            System.out.println("1. Tampilkan Semua Mahasiswa");
            System.out.println("2. Tambah Mahasiswa");
            System.out.println("3. Update Mahasiswa");
            System.out.println("4. Hapus Mahasiswa");
            System.out.println("5. Cek Koneksi Database");
            System.out.println("6. Keluar");
            System.out.println("PILIH OPSI: ");
            pilihan = scanner.nextInt();
            scanner.nextLine();
```

```
            switch (pilihan){
                case 1:
                    mahasiswaController.displayAllMahaiswa();
                    break;

                case 2:
                    // tambah mhs
                    System.out.println("Masukkan NPM: ");
                    String npm = scanner.next();
                    System.out.println("Masukkan Nama: ");
                    String nama = scanner.next();
                    System.out.println("Masukkan Semester: ");
                    int semester = scanner.nextInt();
                    System.out.println("Masukkan IPK: ");
                    float ipk = scanner.nextFloat();
                    System.out.println(npm + nama + semester + ipk);

                    mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
                    break;

                case 3:
                    System.out.print("Masukkan ID mahasiswa: ");
                    int id = scanner.nextInt();
                    scanner.nextLine();

                    System.out.println("Masukkan NPM: ");
                    String npmBaru = scanner.next();
                    System.out.println("Masukkan Nama: ");
                    String namaBaru = scanner.next();
                    System.out.println("Masukkan Semester: ");
                    int semesterBaru = scanner.nextInt();
                    System.out.println("Masukkan IPK: ");
                    float ipkBaru = scanner.nextFloat();

                    mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semesterBaru, ipkBaru);
                    break;

                case 4:
                    System.out.print("Masukkan ID Mahasiswa: ");
                    int idHapus = scanner.nextInt();
                    mahasiswaController.deleteMahasiswa(idHapus);

                case 5:
                    mahasiswaController.checkDatabaseConnection();
                    break;

                case 6:
                    // keluar
                    mahasiswaController.closeConnection();
                    System.out.println("Program Selesai.");
                    return;

                default:
                    System.out.println("Input Tidak Valid");
            }
        }
    }
}
```

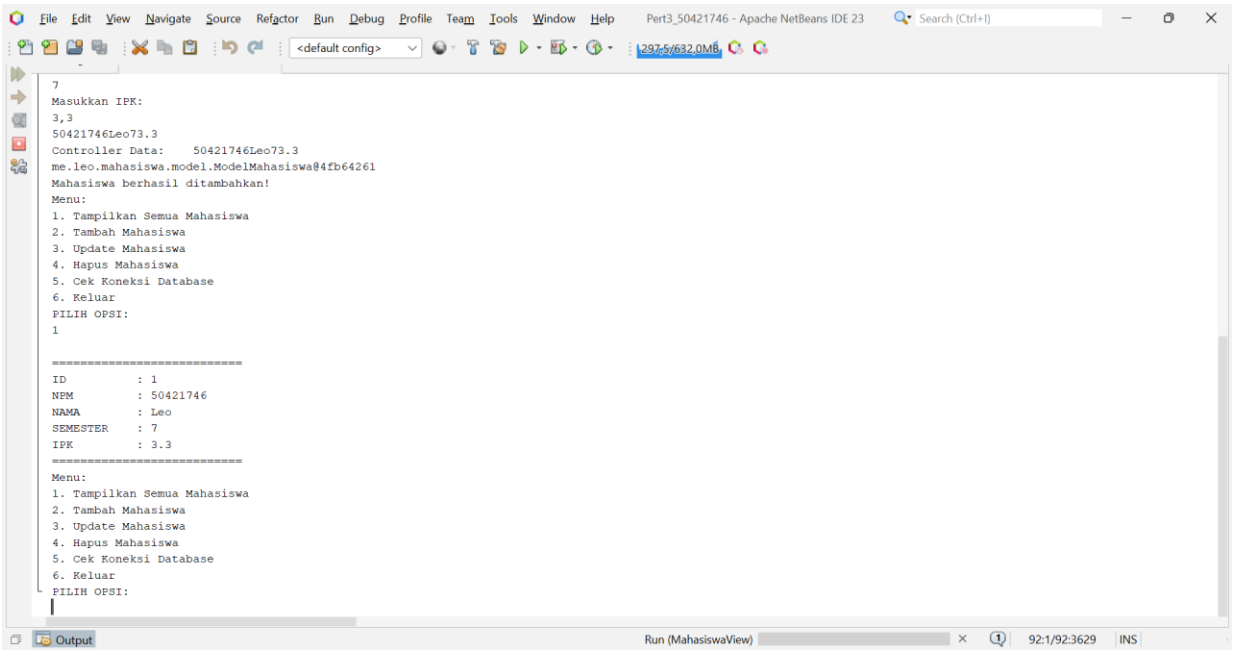
Structure:



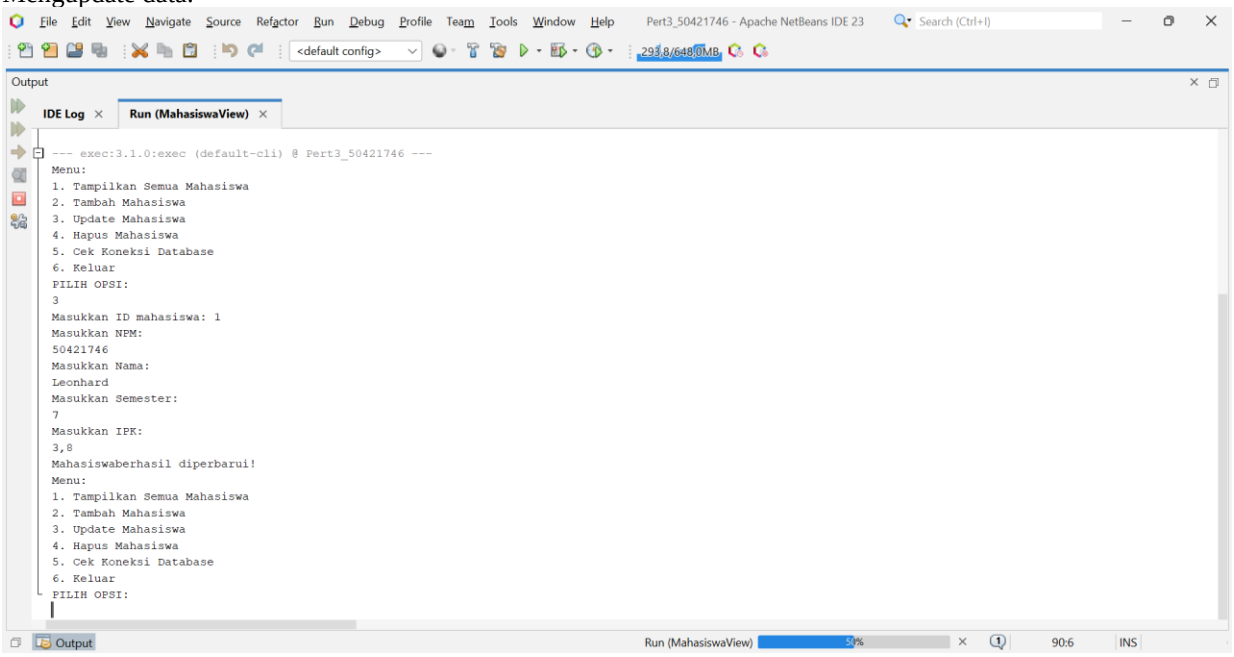
Menambahkan data:



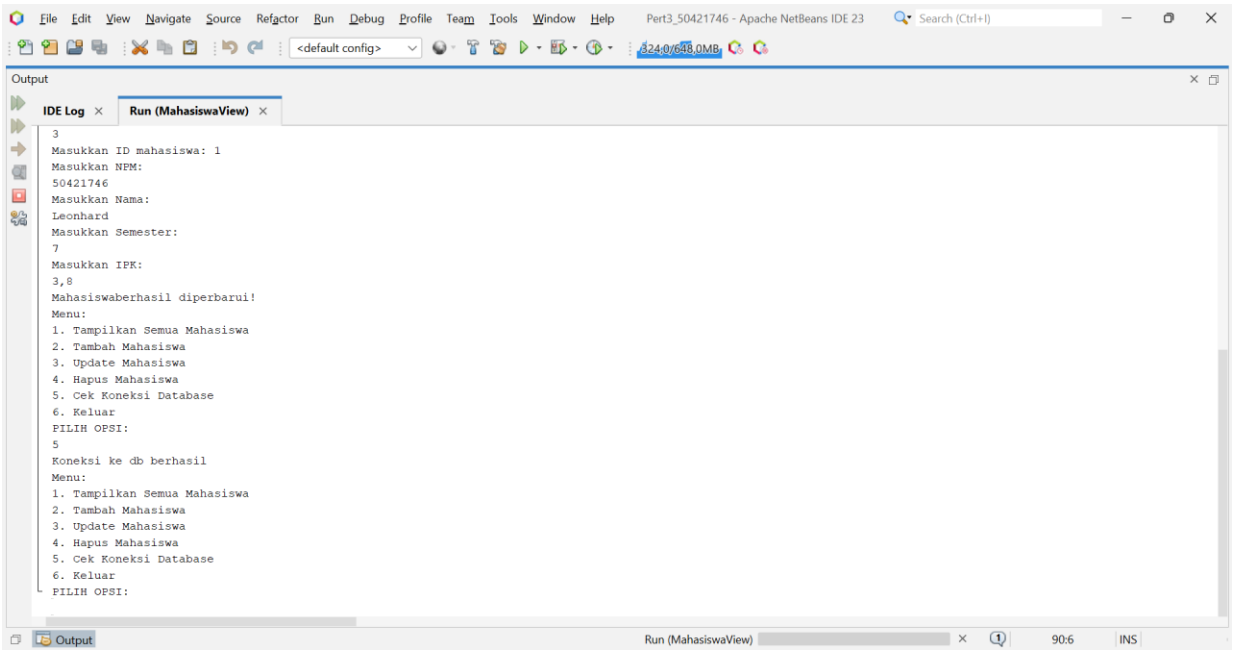
Menampilkan data:

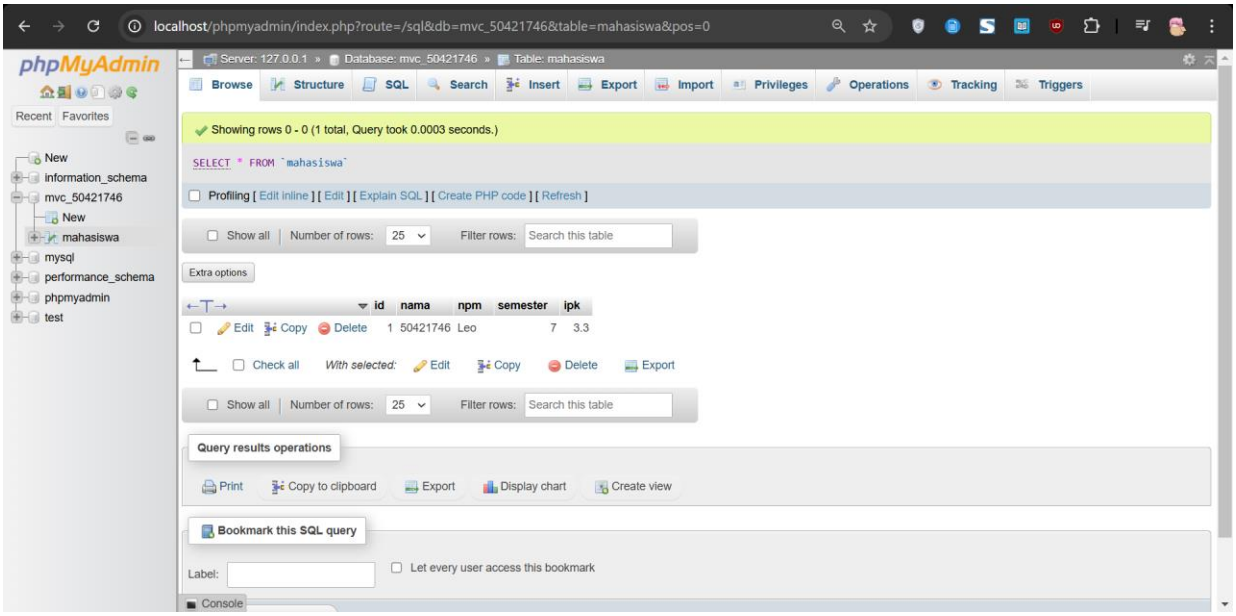


Mengupdate data:

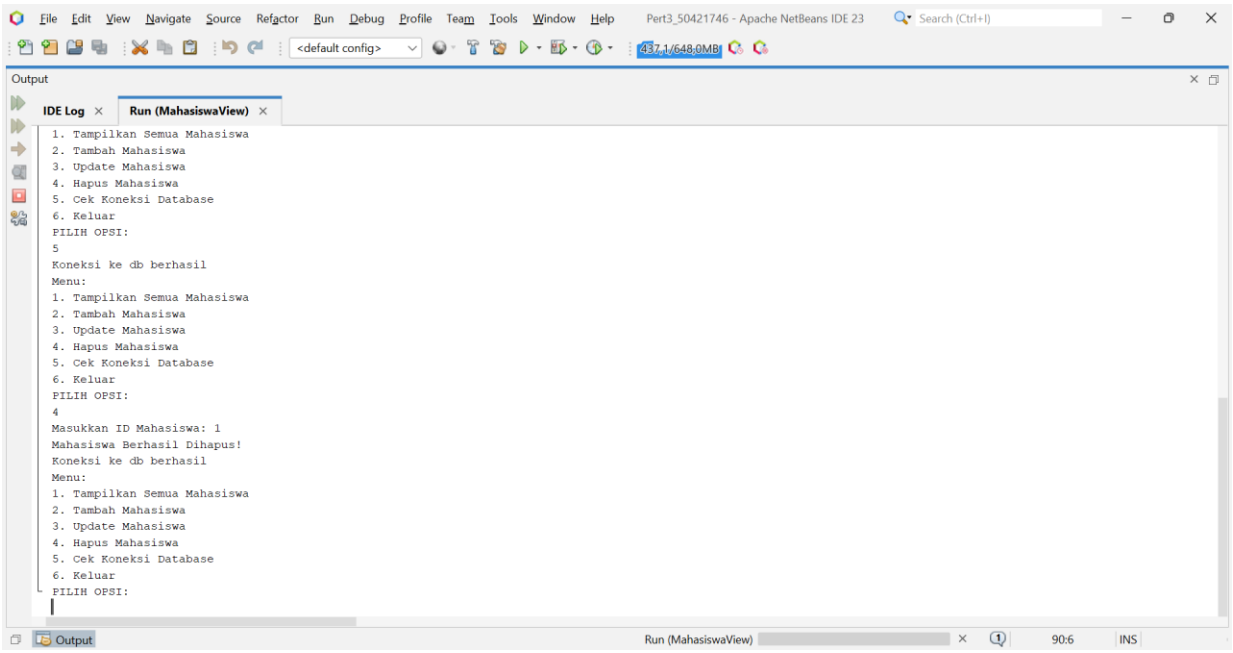


Melihat koneksi ke database:





Menghapus data :



Keluar program: