ACTIVITY PERTEMUAN 3

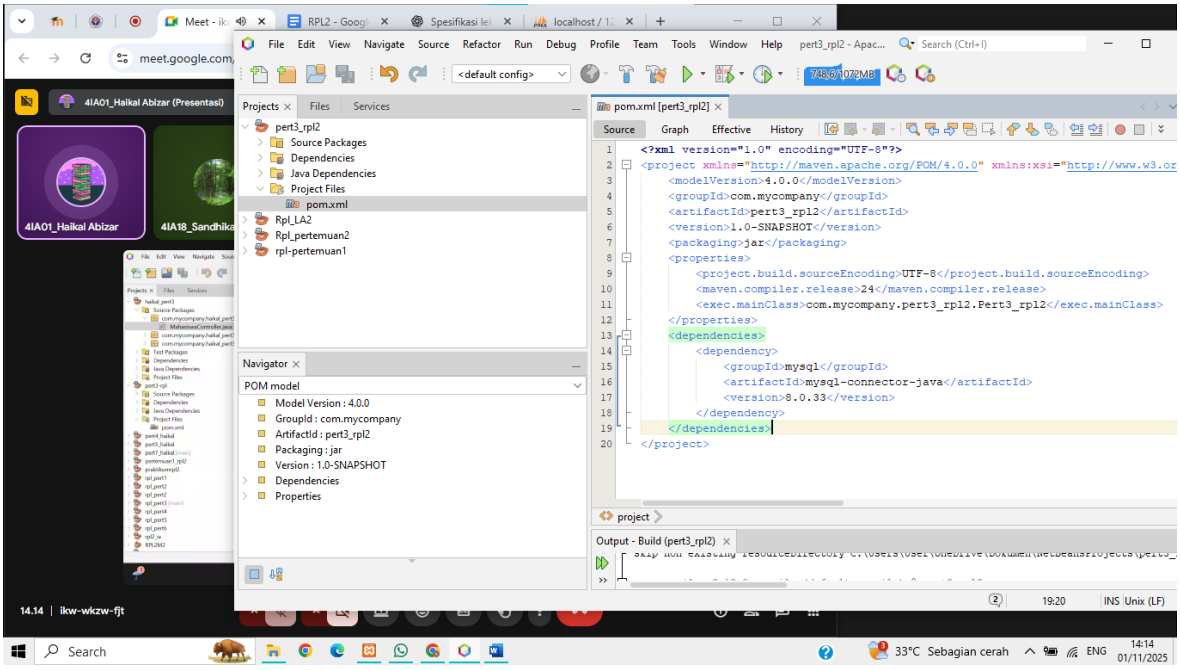
NAMA : Andi Purnama

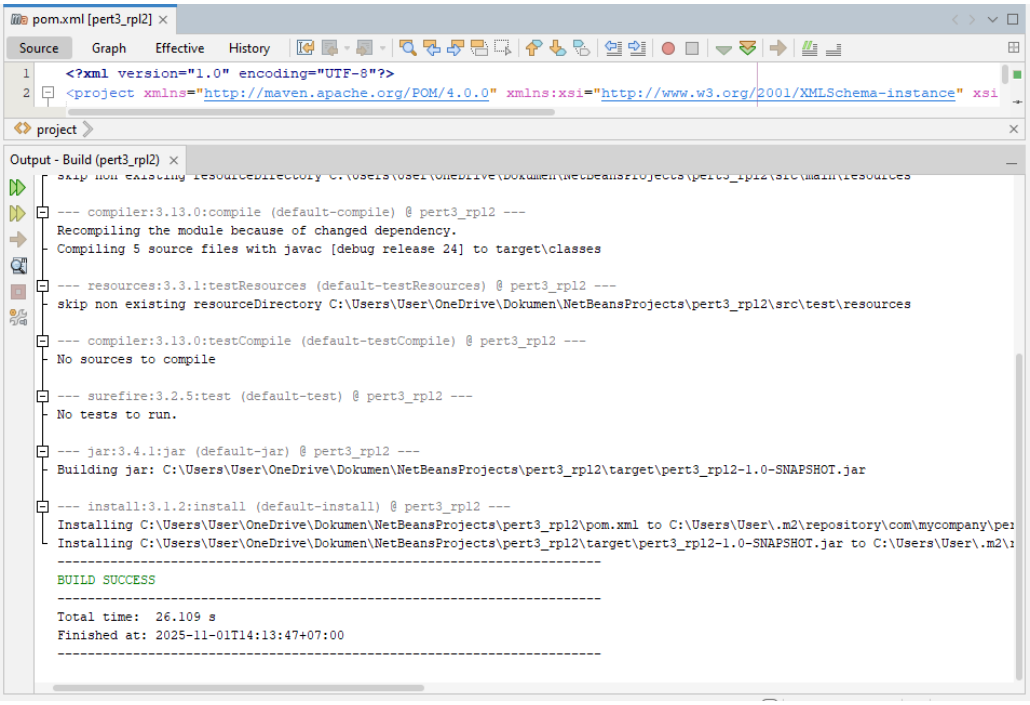
NPM : 50422215

KELAS : 4IA18

MATERI :

MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2





MahasiswaController.java

Source | History

```java
import com.mycompany.pert3_rpl2.Model.MahasiswaDAO;
import com.mycompany.pert3_rpl2.Model.ModelMahasiswa;
import java.util.List;

/**
 *
 * @author User
 */
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("============================");
            }
        }
    }
```

com.mycompany.pert3_rpl2.Controller.MahasiswaController > displayMahasiswaList

MahasiswaController.java

Source | History

```java
        }
    }

    // READ ALL (Menampilkan semua mahasiswa)
    public void displayAllMahasiswa(){
        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("Mahasiswa berhasil diperbarui!");
    }

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

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

MahasiswaController.java × MahasiswaDAO.java ×

Source History

```
85                  pstmt.setInt(5, mahasiswa.getId());
86                  pstmt.executeUpdate();
87              } catch(SQLException e){
                    e.printStackTrace();
89              }
90          }
91
92          public void deleteMahasiswa(int id){
93              String sql = "DELETE FROM mahasiswa WHERE id = ?";
94              try{
95                  PreparedStatement pstmt = connection.prepareStatement(sql);
96                  pstmt.setInt(1, id);
97                  pstmt.executeUpdate();
98              } catch(SQLException e){
                    e.printStackTrace();
100             }
101         }
102
103         // Method untuk menutup koneksi database
104         public void closeConnection() {
105             try {
106                 if (connection != null) {
107                     connection.close();
108                 }
109             } catch (SQLException e) {
                    e.printStackTrace();
111             }
112         }
113
114     }
115
```

MahasiswaController.java × MahasiswaDAO.java × ModelMahasiswa.java ×

Source History

```
34      public String getNpm() {
35          return npm;
36      }
37
38      public void setNpm(String npm) {
39          this.npm = npm;
40      }
41
42      public String getNama() {
43          return nama;
44      }
45
46      public void setNama(String nama) {
47          this.nama = nama;
48      }
49
50      public int getSemester() {
51          return semester;
52      }
53
54      public void setSemester(int semester) {
55          this.semester = semester;
56      }
57
58      public float getIpk() {
59          return ipk;
60      }
61
62      public void setIpk(float ipk) {
63          this.ipk = ipk;
64      }
```

MahasiswaView.java

Source | History

```
61                      String namaBaru = scanner.next();
62                      System.out.print("Masukkan Semester Baru: ");
63                      int semesterBaru = scanner.nextInt();
64                      System.out.print("Masukkan IPK Baru: ");
65                      float ipkBaru = scanner.nextFloat();
66
67                      mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semesterBaru, ipkBaru);
68                      break;
69
70                  case 4:
71                      System.out.print("Masukkan ID Mahasiswa: ");
72                      int idHapus = scanner.nextInt();
73                      mahasiswaController.deleteMahasiswa(idHapus);
74                      break; // ✅ ditambahkan
75
76                  case 5:
77                      mahasiswaController.checkDatabaseConnection();
78                      break;
79
80                  case 6:
81                      mahasiswaController.closeConnection();
82                      System.out.println("Program selesai.");
83                      scanner.close();
84                      return;
85
86                  default:
87                      System.out.println("Input tidak valid");
88              }
89          }
90      }
91  }
```

Output - Run (MahasiswaView)

```
------------------------------------[ jar ]-----------------------------------
The artifact mysql:mysql-connector-java:jar:8.0.33 has been relocated to com.mysql:mysql-connector-j:jar:8.0.33: MySQL Connector/J artifacts moved to reverse-DNS compliant Maven 2+ coor

--- resources:3.3.1:resources (default-resources) @ pert3_rpl2 ---
skip non existing resourceDirectory C:\Users\User\OneDrive\Dokumen\NetBeansProjects\pert3_rpl2\src\main\resources

--- compiler:3.13.0:compile (default-compile) @ pert3_rpl2 ---
Nothing to compile - all classes are up to date.

--- exec:3.1.0:exec (default-cli) @ pert3_rpl2 ---
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 5
Koneksi ke db berhasil
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
```

Output | Run (MahasiswaView) | (1 more...) | 1:1 | INS

Search | 33°C Sebagian cerah | ENG | 14:54 01/11/2025