

ACTIVITY PERTEMUAN 3

NAMA : Felix Windriyareksa Hardyan

NPM : 50421506

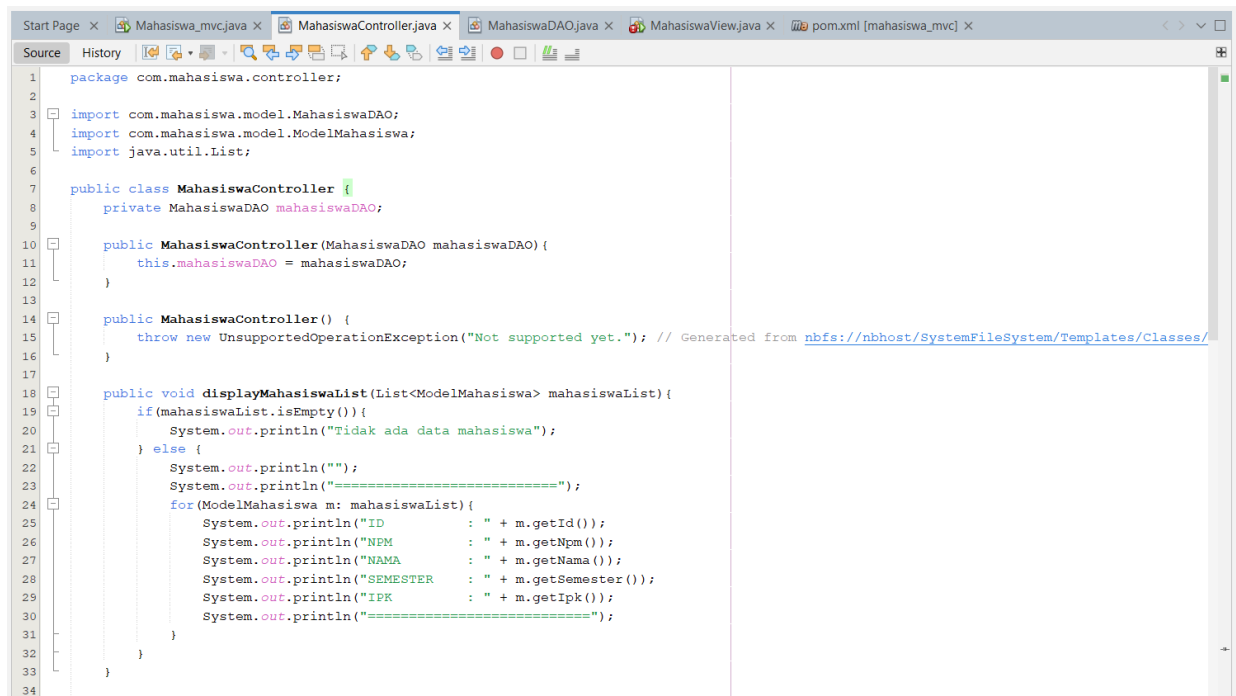
KELAS : 4IA12

MATERI : Model Controller View

MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2

(Screenshoot langkah-langkah sesuai video pembelajaran dan jelaskan dengan ringkas)

1. MahasiswaController



```
1 package com.mahasiswa.controller;
2
3 import com.mahasiswa.model.MahasiswaDAO;
4 import com.mahasiswa.model.ModelMahasiswa;
5 import java.util.List;
6
7 public class MahasiswaController {
8     private MahasiswaDAO mahasiswaDAO;
9
10    public MahasiswaController(MahasiswaDAO mahasiswaDAO) {
11        this.mahasiswaDAO = mahasiswaDAO;
12    }
13
14    public MahasiswaController() {
15        throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/
16    }
17
18    public void displayMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
19        if (mahasiswaList.isEmpty()) {
20            System.out.println("Tidak ada data mahasiswa");
21        } else {
22            System.out.println("");
23            System.out.println("=====");
24            for (ModelMahasiswa m: mahasiswaList) {
25                System.out.println("ID           : " + m.getId());
26                System.out.println("NPM          : " + m.getNpm());
27                System.out.println("NAMA         : " + m.getNama());
28                System.out.println("SEMESTER     : " + m.getSemester());
29                System.out.println("IPK          : " + m.getIpk());
30                System.out.println("=====");
31            }
32        }
33    }
34 }
```

```
Start Page x Mahasiswa_mvc.java x MahasiswaController.java x MahasiswaDAO.java x MahasiswaView.java x pom.xml [mahasiswa_mvc] x
Source History
49 }
50
51 // READ ALL (Menampilkan semua mahasiswa)
52 public void displayAllMahasiswa() {
53     List<ModelMahasiswa> mahasiswaList = mahasiswaDAO.getAllMahasiswa();
54     displayMahasiswaList(mahasiswaList);
55 }
56
57 public void addMahasiswa(String npm, String nama, int semester, float ipk) {
58     ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(0, npm, nama, semester, ipk);
59     System.out.println("Controller Data: " + npm + nama + semester + ipk);
60     System.out.println(mahasiswaBaru);
61     mahasiswaDAO.addMahasiswa(mahasiswaBaru);
62     displayMessage("Mahasiswa berhasil ditambahkan!");
63 }
64
65 public void updateMahasiswa(int id, String npm, String nama, int semester, float ipk) {
66     ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id, npm, nama, semester, ipk);
67     mahasiswaDAO.updateMahasiswa(mahasiswaBaru);
68     displayMessage("Mahasiswa berhasil diperbarui!");
69 }
70
71 public void deleteMahasiswa(int id) {
72     mahasiswaDAO.deleteMahasiswa(id);
73     displayMessage("Mahasiswa Berhasil Dihapus!");
74 }
75
76 public void closeConnection() {
77     mahasiswaDAO.closeConnection();
78 }
79 }
```

2. MahasiswaDAO

```
Start Page x Mahasiswa_mvc.java x MahasiswaController.java x MahasiswaDAO.java x MahasiswaView.java x pom.xml [mahasiswa_mvc] x
Source History
1 package com.mahasiswa.model;
2
3 import java.sql.*;
4 import java.util.ArrayList;
5 import java.util.List;
6
7 public class MahasiswaDAO {
8     private Connection connection;
9
10    public MahasiswaDAO() {
11        try {
12            Class.forName("com.mysql.cj.jdbc.Driver");
13            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/mahasiswa_mvc", "root", "");
14        } catch (Exception e) {
15            e.printStackTrace();
16        }
17    }
18
19    public boolean checkConnection() {
20        try {
21            if (connection != null && !connection.isClosed()) {
22                return true; //koneksi berhasil
23            }
24        } catch (SQLException e) {
25            e.printStackTrace();
26        }
27        return false;
28    }
29 }
```

```
Source History
28 }
29
30 public void addMahasiswa(ModelMahasiswa mahasiswa){
31     String sql = "INSERT INTO mahasiswa (npm, nama, semester, ipk) VALUES (?, ?, ?, ?)";
32     try{
33         PreparedStatement pstmt = connection.prepareStatement(sql);
34         pstmt.setString(1, mahasiswa.getNpm());
35         pstmt.setString(2, mahasiswa.getNama());
36         pstmt.setInt(3, mahasiswa.getSemester());
37         pstmt.setFloat(4, mahasiswa.getIpk());
38         pstmt.executeUpdate();
39     } catch(SQLException e){
40         e.printStackTrace();
41     }
42 }
43
44 public List<ModelMahasiswa> getAllMahasiswa(){
45     List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
46     String sql = "SELECT * FROM mahasiswa";
47     try{
48         Statement stmt = connection.createStatement();
49         ResultSet rs = stmt.executeQuery(sql);
50         while(rs.next()){
51             mahasiswaList.add(new ModelMahasiswa(
52                 rs.getInt("id"),
53                 rs.getString("npm"),
54                 rs.getString("nama"),
55                 rs.getInt("semester"),
56                 rs.getFloat("ipk")
57             ));
58         }
59     } catch(SQLException e){
60         e.printStackTrace();
61     }
62     return mahasiswaList;
63 }
64
65 public void updateMahasiswa(ModelMahasiswa mahasiswa){
66     String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?";
67     try{
68         PreparedStatement pstmt = connection.prepareStatement(sql);
69         pstmt.setString(1, mahasiswa.getNpm());
70         pstmt.setString(2, mahasiswa.getNama());
71         pstmt.setInt(3, mahasiswa.getSemester());
72         pstmt.setFloat(4, mahasiswa.getIpk());
73         pstmt.setInt(5, mahasiswa.getId());
74         pstmt.executeUpdate();
75     } catch(SQLException e){
76         e.printStackTrace();
77     }
78 }
79
80 public void deleteMahasiswa(int id){
81     String sql = "DELETE FROM mahasiswa WHERE id = ?";
82     try{
83         PreparedStatement pstmt = connection.prepareStatement(sql);
84         pstmt.setInt(1, id);
85         pstmt.executeUpdate();
86     } catch(SQLException e){
87         e.printStackTrace();
88     }
89 }
90
91 // Method untuk menutup koneksi database
92 public void closeConnection() {
93     try {
94         if (connection != null) {
95             connection.close();
96         }
97     } catch (SQLException e) {
98         e.printStackTrace();
99     }
100 }
```

3. ModelMahasiswa

```
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package com.mahasiswa.model;
6
7   /**
8    *
9    * @author felix
10   */
11  public class ModelMahasiswa {
12      private int id;
13      private String npm;
14      private String nama;
15      private int semester;
16      private float ipk;
17
18      public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk) {
19          this.id = id;
20          this.npm = npm;
21          this.nama = nama;
22          this.semester = semester;
23          this.ipk = ipk;
24      }
25
26      public int getId() {
27          return id;
28      }
29
30      public void setId(int id) {
31          this.id = id;
32      }
33
34      public String getNpm() {
35          return npm;
36      }
37
38      public String getNpm() {
39          return npm;
40      }
41
42      public void setNpm(String npm) {
43          this.npm = npm;
44      }
45
46      public String getNama() {
47          return nama;
48      }
49
50      public void setNama(String nama) {
51          this.nama = nama;
52      }
53
54      public int getSemester() {
55          return semester;
56      }
57
58      public void setSemester(int semester) {
59          this.semester = semester;
60      }
61
62      public float getIpk() {
63          return ipk;
64      }
65
66      public void setIpk(float ipk) {
67          this.ipk = ipk;
68      }
69  }
```

4. MahasiswaView

```
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package com.mahasiswa.view;
6
7   import com.mahasiswa.controller.MahasiswaController;
8   import com.mahasiswa.model.MahasiswaDAO;
9   import java.util.Scanner;
10
11  /**
12   *
13   * @author felix
14   */
15  public class MahasiswaView {
16      public static void main(String[] args) {
17          MahasiswaDAO mahasiswaDAO = new MahasiswaDAO();
18          MahasiswaController mahasiswaController = new MahasiswaController();
19
20          Scanner scanner = new Scanner(System.in);
21          int pilihan;
22
23          while(true) {
24              System.out.println("Menu: ");
25              System.out.println("1. Tampilkan seluruh mhs");
26              System.out.println("2. Tambahkan mhs");
27              System.out.println("3. Update mhs");
28              System.out.println("4. Hapus mhs");
29              System.out.println("5. Cek koneksi");
30              System.out.println("6. Keluar");
31              System.out.println("PILIH OPSI: ");
32              pilihan = scanner.nextInt();
33              scanner.nextLine();
```

```
Source History
34      switch(pilihan) {
35          case 1:
36              mahasiswaController.displayAllMahasiswa();
37              break;
38          case 2:
39              System.out.println("Masukkan NPM: ");
40              String npm = scanner.next();
41              System.out.println("Masukkan Nama: ");
42              String nama = scanner.next();
43              System.out.println("Masukkan Semester: ");
44              int semester = scanner.nextInt();
45              System.out.println("Masukkan IPK: ");
46              float ipk = scanner.nextFloat();
47
48              System.out.println(npm + nama + semester + ipk);
49              mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
50              break;
51          case 3:
52              System.out.println("Masukkan ID mahasiswa: ");
53              int id = scanner.nextInt();
54              scanner.nextLine();
55
56              System.out.println("Masukkan NPM: ");
57              String npmBaru = scanner.next();
58              System.out.println("Masukkan Nama: ");
59              String namaBaru = scanner.next();
60              System.out.println("Masukkan Semester: ");
61              int semesterBaru = scanner.nextInt();
62              System.out.println("Masukkan IPK: ");
63              float ipkBaru = scanner.nextFloat();
64
65              System.out.println(npmBaru + namaBaru + semesterBaru + ipkBaru);
66              mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semesterBaru, ipkBaru);
67              break;
68          case 4:
69              System.out.println("Masukkan ID Mahasiswa: ");
70
```

```
Source History
63      System.out.println("Masukkan IPK: ");
64      float ipkBaru = scanner.nextFloat();
65
66      System.out.println(npmBaru + namaBaru + semesterBaru + ipkBaru);
67      mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semesterBaru, ipkBaru);
68      break;
69  case 4:
70      System.out.println("Masukkan ID Mahasiswa: ");
71      int idHapus = scanner.nextInt();
72      mahasiswaController.deleteMahasiswa(idHapus);
73      break;
74  case 5:
75      mahasiswaController.checkDatabaseConnection();
76      break;
77  case 6:
78      mahasiswaController.closeConnection();
79      System.out.println("Program Selesai");
80      return;
81  default:
82      System.out.println("Input tidak valid");
83  }
84  }
85  }
86  }
87  }
```

5. Mahasiswa_mvc

```
Source History
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   */
4
5   package com.mycompany.mahasiswa_mvc;
6
7   /**
8    *
9    * @author felix
10   */
11   public class Mahasiswa_mvc {
12
13       public static void main(String[] args) {
14           System.out.println("Hello World!");
15       }
16   }
17   }
```

6. Pom.xml

```
Source Graph Effective History
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.a
3
4      <modelVersion>4.0.0</modelVersion>
5      <groupId>com.mycompany</groupId>
6      <artifactId>mahasiswa_mvc</artifactId>
7      <version>1.0-SNAPSHOT</version>
8      <packaging>jar</packaging>
9      <properties>
10         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
11         <maven.compiler.source>21</maven.compiler.source>
12         <maven.compiler.target>21</maven.compiler.target>
13         <exec.mainClass>com.mycompany.mahasiswa_mvc.Mahasiswa_mvc</exec.mainClass>
14     </properties>
15     <dependencies>
16         <dependency>
17             <groupId>mysql</groupId>
18             <artifactId>mysql-connector-java</artifactId>
19             <version>8.0.33</version>
20         </dependency>
21     </dependencies>
22 </project>
```

7. Output Program



The screenshot shows an IDE's output window titled "Output - Run (mahasiswa_mvc)". The window contains the following text:

```
Compiling 5 source files with javac [debug target 21] to target\classes
-----
COMPILATION ERROR :
-----
com/mahasiswa/view/MahasiswaView.java:[82,60] ';' expected
1 error
-----

BUILD FAILURE
-----

Total time: 1.052 s
Finished at: 2024-11-01T16:58:51+07:00
-----

Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:3.11.0:compile (default-compile) on project mahasiswa_mvc: Comp
com/mahasiswa/view/MahasiswaView.java:[82,60] ';' expected
```

The error message indicates a semicolon is expected at line 82, column 60 of the file `com/mahasiswa/view/MahasiswaView.java`. The build failed, and the total time was 1.052 seconds. The finished time was 2024-11-01T16:58:51+07:00. The error occurred during the compilation phase of the Maven build.