

Nama: Shaun Patrick Hendra

NPM: 51421416

Kelas: 4IA28

Pertemuan 5

Source Code:

SpringBootApplication

```
1  package com.mahasiswa;
2  import com.mahasiswa.controller.MahasiswaController;
3  import org.springframework.beans.factory.annotation.Autowired;
4  import org.springframework.boot.CommandLineRunner;
5  import org.springframework.boot.SpringApplication;
6  import org.springframework.boot.autoconfigure.SpringBootApplication;
7
8  /**
9   *
10  * @author ASUS
11  */
12  @SpringBootApplication
13  public class Pertemuan5SpringBootApplication implements CommandLineRunner{
14
15      @Autowired
16      private MahasiswaController mhsController;
17      public static void main(String[] args) {
18          SpringApplication.run(Pertemuan5SpringBootApplication.class, args);
19      }
20
21      @Override
22      public void run(String... args) throws Exception {
23          mhsController.tampilkanMenu();
24      }
25
26  }
```

MahasiswaController

```
1  package com.mahasiswa.controller;
2
3  import com.mahasiswa.model.ModelMahasiswa;
4  import com.mahasiswa.repository.MahasiswaRepository;
5  import org.springframework.beans.factory.annotation.Autowired;
6  import org.springframework.stereotype.Controller;
7
8  import java.util.List;
9  import java.util.Scanner;
10
11  @Controller
12  public class MahasiswaController {
13
14      @Autowired
15      private MahasiswaRepository mahasiswaRepository;
16
17      public void tampilkanMenu() {
18          Scanner scanner = new Scanner(System.in);
19          int opsi;
20
21          do {
22              System.out.println("\nMenu:");
23              System.out.println("1. Tampilkan semua mahasiswa");
24              System.out.println("2. Tambah mahasiswa baru");
25              System.out.println("3. Cek koneksi database");
26              System.out.println("4. Keluar");
27              System.out.print("Pilih opsi: ");
```

```

28     opsi = scanner.nextInt();
29     scanner.nextLine(); // menangkap newline
30
31     switch (opsi) {
32     case 1:
33         tampilkanSemuaMahasiswa();
34         break;
35     case 2:
36         tambahMahasiswa(scanner);
37         break;
38     case 3:
39         cekKoneksi();
40         break;
41     case 4:
42         System.out.println("Keluar dari program.");
43         break;
44     default:
45         System.out.println("Ops! tidak valid, coba lagi.");
46     }
47
48     } while (opsi != 4);
49 }
50
51 private void tampilkanSemuaMahasiswa() {
52     List<ModelMahasiswa> mahasiswaList = mahasiswaRepository.findAll();
53     if (mahasiswaList.isEmpty()) {
54         System.out.println("Tidak ada data mahasiswa.");
55     } else {
56         mahasiswaList.forEach(mahasiswa -> System.out.println(mahasiswa));
57     }
58 }

```

Activate Windows

Go to Settings to activate Windows.

```

60 private void tambahMahasiswa(Scanner scanner) {
61     System.out.print("Masukkan NPM : ");
62     String npm = scanner.nextLine();
63     System.out.print("Masukkan Nama : ");
64     String nama = scanner.nextLine();
65     System.out.print("Masukkan Semester : ");
66     int semester = scanner.nextInt();
67     System.out.print("Masukkan IPK : ");
68     float ipk = scanner.nextFloat();
69
70     ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester, ipk);
71     mahasiswaRepository.save(mahasiswa);
72     System.out.println("Mahasiswa berhasil ditambahkan.");
73 }
74
75 private void cekKoneksi() {
76     try {
77         mahasiswaRepository.findAll();
78         System.out.println("Koneksi ke database berhasil.");
79     } catch (Exception e) {
80         System.out.println("Gagal terhubung ke database.");
81     }
82 }
83 }

```

ModelMahasiswa

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to cha
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this ten
4  */
5  package com.mahasiswa.model;
6
7  import jakarta.persistence.*;
8
9  /**
10   *
11   * @author LAB_CENGKARENG
12   */
13  @Entity
14  @Table(name = "mahasiswa")
15  public class ModelMahasiswa {
16      @Id
17      @GeneratedValue(strategy = GenerationType.IDENTITY)
18      @Column(name = "id")
19      private int id;
20
21      @Column(name = "npm", nullable = false, length = 8)
22      private String npm;
23
24      @Column(name = "nama", nullable = false, length = 50)
25      private String nama;
26
27      @Column(name = "semester")
28      private int semester;
29
30      @Column(name = "ipk")
```

Activate Windows

```
31     private float ipk;
32
33     public ModelMahasiswa() {
34     }
35
36
37     public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk) {
38         this.id = id;
39         this.npm = npm;
40         this.nama = nama;
41         this.semester = semester;
42         this.ipk = ipk;
43     }
44
45     public int getId() {
46         return id;
47     }
48
49     public void setId(int id) {
50         this.id = id;
51     }
52
53     public String getNpm() {
54         return npm;
55     }
56
57     public void setNpm(String npm) {
58         this.npm = npm;
59     }
60
```

```

61 public String getNama() {
62     return nama;
63 }
64
65 public void setNama(String nama) {
66     this.nama = nama;
67 }
68
69 public int getSemester() {
70     return semester;
71 }
72
73 public void setSemester(int semester) {
74     this.semester = semester;
75 }
76
77 public float getIpk() {
78     return ipk;
79 }
80
81 public void setIpk(float ipk) {
82     this.ipk = ipk;
83 }
84
85 @Override
86 public String toString() {
87     return "Mahasiswa{" +
88         "id=" + id +
89         ", npm='" + npm + '\'' +
90         ", nama='" + nama + '\'' +
91
92         ", semester='" + semester + '\'' +
93         ", ipk='" + ipk + '\'' +
94         '}';
95 }
96

```

MahasiswaRepository

```
1 package com.mahasiswa.repository;
2
3 import com.mahasiswa.model.ModelMahasiswa;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.stereotype.Repository;
6
7 @Repository
8 public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Long> {
9
10 }
```

Pom.xml

```
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"
3         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <groupId>com.mycompany</groupId>
6     <artifactId>rpl_shaun_spring</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.rpl_shaun_spring.Rpl_shaun_spring</exec.mainClass>
14     </properties>
15     <parent>
16         <groupId>org.springframework.boot</groupId>
17         <artifactId>spring-boot-starter-parent</artifactId>
18         <version>3.3.3</version>
19         <relativePath/>
20     </parent>
21     <dependencies>
22         <!-- Hibernate + Spring Data JPA -->
23         <dependency>
24             <groupId>org.springframework.boot</groupId>
25             <artifactId>spring-boot-starter-data-jpa</artifactId>
26         </dependency>
27
28         <!-- MySQL Connector -->
29         <dependency>
30             <groupId>mysql</groupId>
```

Activate Windows


```

31         <artifactId>mysql-connector-java</artifactId>
32         <version>8.0.33</version>
33     </dependency>
34
35     <!-- Spring Boot Web dependency (for MVC if needed) -->
36     <dependency>
37         <groupId>org.springframework.boot</groupId>
38         <artifactId>spring-boot-starter-web</artifactId>
39     </dependency>
40
41     <!-- Testing dependencies -->
42     <dependency>
43         <groupId>org.springframework.boot</groupId>
44         <artifactId>spring-boot-starter-test</artifactId>
45         <scope>test</scope>
46     </dependency>
47 </dependencies>
48
49 <build>
50     <plugins>
51         <plugin>
52             <groupId>org.springframework.boot</groupId>
53             <artifactId>spring-boot-maven-plugin</artifactId>
54         </plugin>
55     </plugins>
56 </build>
57 </project>

```

Output:

Menu:

1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar

Pilih opsi: 1

Hibernate: select mml_0.id,mml_0.ipk,mml_0.nama,mml_0.npm,mml_0.semester from mahasiswa mml_0

Mahasiswa{id=2, npm='51421416', nama='Shaun Patrick Hendra', semester='7', ipk='4.0'}

Mahasiswa{id=3, npm='5728375', nama='JUJU', semester='9', ipk='3.0'}

Mahasiswa{id=4, npm='5372675', nama='KAKA', semester='6', ipk='5.0'}

Menu:

1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar

Activate Windows

Nama: Shaun Patrick Hendra

NPM: 51421416

Kelas: 4IA28

Pertemuan 6

Source Code:

MahasiswaApp

```
1 package com.mahasiswa;
2
3 import com.mahasiswa.controller.MahasiswaController;
4 import com.mahasiswa.service.MahasiswaService;
5 import com.mahasiswa.view.MahasiswaView;
6 import org.springframework.boot.ApplicationArguments;
7 import org.springframework.boot.ApplicationRunner;
8 import org.springframework.boot.SpringApplication;
9 import org.springframework.boot.autoconfigure.SpringBootApplication;
10 import org.springframework.beans.factory.annotation.Autowired;
11 import org.springframework.context.ApplicationContext;
12
13 @SpringBootApplication
14 public class MahasiswaApp implements ApplicationRunner {
15
16     @Autowired
17     private MahasiswaService mahasiswaService;
18
19     public static void main(String[] args) {
20         System.setProperty("java.awt.headless", "false"); // Disable headless mode
21
22         // Start the Spring application and get the application context
23         ApplicationContext context = SpringApplication.run(MahasiswaApp.class, args);
24
25         // Instantiate the view and inject the controller manually
26         MahasiswaController controller = context.getBean(MahasiswaController.class);
27         MahasiswaView mahasiswaView = new MahasiswaView(controller);
28         mahasiswaView.setVisible(true);
29     }
30
31     @Override
32     public void run(ApplicationArguments args) throws Exception {
33         // Implement this method if you need to execute logic after Spring
34         // Otherwise, you can leave it as is.
35     }
36 }
```

Activate Windows

MahasiswaController

```
1 package com.mahasiswa.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.web.bind.annotation.*;
5 import com.mahasiswa.model.ModelMahasiswa;
6 import com.mahasiswa.service.MahasiswaService;
7
8 import java.util.List;
9 import org.springframework.stereotype.Controller;
10
11
12 @Controller
13 public class MahasiswaController {
14
15     @Autowired
16     private MahasiswaService mahasiswaService;
17
18     // Add new Mahasiswa
19     public String addMahasiswa(@RequestBody ModelMahasiswa mhs) {
20         mahasiswaService.addMhs(mhs);
21         return "Mahasiswa added successfully";
22     }
23
24     // Get Mahasiswa by ID
25     public ModelMahasiswa getMahasiswa(@PathVariable int id) {
26         return mahasiswaService.getMhs(id);
27     }
28
29     // Update Mahasiswa
30     public String updateMahasiswa(@RequestBody ModelMahasiswa mhs) {
31         mahasiswaService.updateMhs(mhs);
32
33         return "Mahasiswa updated successfully";
34     }
35
36     // Delete Mahasiswa by ID
37     public String deleteMahasiswa(@PathVariable int id) {
38         mahasiswaService.deleteMhs(id);
39         return "Mahasiswa deleted successfully";
40     }
41
42     // Get all Mahasiswa
43     public List<ModelMahasiswa> getAllMahasiswa() {
44         return mahasiswaService.getAllMahasiswa();
45     }
46 }
```

ModelMahasiswa

```
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
4   */
5   package com.mahasiswa.model;
6
7   import jakarta.persistence.*;
8
9   /**
10    *
11    * @author LAB_CENGKARENG
12    */
13   @Entity
14   @Table(name = "mahasiswa")
15   public class ModelMahasiswa {
16       @Id
17       @GeneratedValue(strategy = GenerationType.IDENTITY)
18       @Column(name = "id")
19       private int id;
20
21       @Column(name = "npm", nullable = false, length = 8)
22       private String npm;
23
24       @Column(name = "nama", nullable = false, length = 50)
25       private String nama;
26
27       @Column(name = "semester")
28       private int semester;
29
30       @Column(name = "ipk")
```

Activate

```
31     private float ipk;
32
33     public ModelMahasiswa() {
34     }
35
36
37     public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk) {
38         this.id = id;
39         this.npm = npm;
40         this.nama = nama;
41         this.semester = semester;
42         this.ipk = ipk;
43     }
44
45     public int getId() {
46         return id;
47     }
48
49     public void setId(int id) {
50         this.id = id;
51     }
52
53     public String getNpm() {
54         return npm;
55     }
56
57     public void setNpm(String npm) {
58         this.npm = npm;
59     }
60
```

```
61  public String getNama() {  
62      return nama;  
63  }  
64  
65  public void setNama(String nama) {  
66      this.nama = nama;  
67  }  
68  
69  public int getSemester() {  
70      return semester;  
71  }  
72  
73  public void setSemester(int semester) {  
74      this.semester = semester;  
75  }  
76  
77  public float getIpk() {  
78      return ipk;  
79  }  
80  
81  public void setIpk(float ipk) {  
82      this.ipk = ipk;  
83  }  
84  
85  
86  }  
87
```

ModelTabelMahasiswa

```

1  package com.mahasiswa.model;
2  import javax.swing.table.AbstractTableModel;
3  import java.util.List;
4
5  public class ModelTabelMahasiswa extends AbstractTableModel{
6      private List<ModelMahasiswa> mahasiswaList;
7      private String[] columnNames = {"ID", "NPM", "Nama", "Semester", "IPK"};
8
9      public ModelTabelMahasiswa(List<ModelMahasiswa> mahasiswaList) {
10         this.mahasiswaList = mahasiswaList;
11     }
12
13     @Override
14     public int getRowCount() {
15         return mahasiswaList.size(); // Jumlah baris sesuai dengan jumlah data mahasiswa
16     }
17
18     @Override
19     public int getColumnCount() {
20         return columnNames.length; // Jumlah kolom sesuai dengan jumlah elemen dalam columnNames
21     }
22
23     @Override
24     public Object getValueAt(int rowIndex, int columnIndex) {
25         ModelMahasiswa mahasiswa = mahasiswaList.get(rowIndex);
26         switch (columnIndex) {
27             case 0:
28                 return mahasiswa.getId();
29             case 1:
30                 return mahasiswa.getNpm();

```

Activate Windows

```

31         case 2:
32             return mahasiswa.getNama();
33         case 3:
34             return mahasiswa.getSemester();
35         case 4:
36             return mahasiswa.getIpk();
37         default:
38             return null;
39     }
40 }
41
42 @Override
43 public String getColumnName(int column) {
44     return columnNames[column]; // Mengatur nama kolom
45 }
46
47 @Override
48 public boolean isCellEditable(int rowIndex, int columnIndex) {
49     return false; // Semua sel tidak dapat diedit
50 }
51
52 // Method untuk menambahkan atau memodifikasi data, jika dibutuhkan
53 public void setMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
54     this.mahasiswaList = mahasiswaList;
55     fireTableDataChanged(); // Memberitahu JTable bahwa data telah berubah
56 }
57 }

```

MahasiswaRepository

```

1 package com.mahasiswa.repository;
2
3 import com.mahasiswa.model.ModelMahasiswa;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.stereotype.Repository;
6
7 @Repository
8 public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Integer> {
9     public Object findById(int id);
10
11     public void deleteById(int id);
12 }

```


MahasiswaService

```
1 package com.mahasiswa.service;
2
3 import com.mahasiswa.model.ModelMahasiswa;
4 import com.mahasiswa.repository.MahasiswaRepository;
5 import jakarta.transaction.Transactional;
6 import java.util.List;
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.stereotype.Service;
9
10 @Service
11 public class MahasiswaService {
12
13     @Autowired
14     private MahasiswaRepository repository;
15
16     public void addMhs(ModelMahasiswa mhs) {
17         repository.save(mhs);
18     }
19
20     public ModelMahasiswa getMhs(int id) {
21         ModelMahasiswa mahasiswa = (ModelMahasiswa) repository.findById(id);
22         return mahasiswa != null ? mahasiswa : null;
23     }
24
25     public void updateMhs(ModelMahasiswa mhs) {
26         repository.save(mhs);
27     }
28
29     @Transactional
30     public void deleteMhs(int id) {
31         repository.deleteById(id);
32     }
33
34     public List<ModelMahasiswa> getAllMahasiswa() {
35         return repository.findAll();
36     }
37 }
```

Activate Windows

MahasiswaView

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change thi
3   * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
4   */
5   package com.mahasiswa.view;
6
7   import com.mahasiswa.controller.MahasiswaController;
8   import com.mahasiswa.model.ModelMahasiswa;
9   import com.mahasiswa.model.ModelTabelMahasiswa;
10  import java.util.List;
11  import javax.swing.JLabel;
12  import javax.swing.JOptionPane;
13  import javax.swing.JPanel;
14  import javax.swing.JTextField;
15
16  /**
17   *
18   * @author LAB_CENGKARENG
19   */
20  public class MahasiswaView extends javax.swing.JFrame {
21      private MahasiswaController controller;
22
23      /**
24       * Creates new form MahasiswaView
25       */
26      public MahasiswaView(MahasiswaController controller) {
27          this.controller = controller;
28          initComponents();
29          loadMahasiswaTable();
30      }
```

Activate Windows

```

31
32 private MahasiswaView() {
33     throw new UnsupportedOperationException("Not supported yet."); // Generated from
34 }
35
36 public void loadMahasiswaTable() {
37     // Ambil data dari controller
38     List<ModelMahasiswa> listMahasiswa = controller.getAllMahasiswa();
39
40     // Buat model tabel kustom dengan data mahasiswa
41     ModelTabelMahasiswa tableModel = new ModelTabelMahasiswa(listMahasiswa);
42
43     // Set model pada JTable
44     dataTable.setModel(tableModel);
45 }
46
47 /**
48  * This method is called from within the constructor to initialize the form.
49  * WARNING: Do NOT modify this code. The content of this method is always
50  * regenerated by the Form Editor.
51  */
52 @SuppressWarnings("unchecked")
53 Generated Code
192
193 private void npmFieldActionPerformed(java.awt.event.ActionEvent evt) {
194     // TODO add your handling code here:
195 }
196
197 private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) {
198     String npm = getNpmField().getText();

```

Activate Windows

```

199     String nama = getNamaField().getText();
200     int semester = Integer.parseInt(getSemesterField().getText());
201     float ipk = Float.parseFloat(getIpkField().getText());
202     ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester, ipk);
203     System.out.println(mahasiswa.getIpk());
204     System.out.println(mahasiswa.getNama());
205     System.out.println(mahasiswa.getSemester());
206     System.out.println(mahasiswa.getNpm());
207
208     controller.addMahasiswa(mahasiswa);
209     loadMahasiswaTable();
210 }
211
212 private void buangButtonActionPerformed(java.awt.event.ActionEvent evt) {
213     JTextField idField = new JTextField(5);
214
215     // Membuat panel untuk menampung JTextField
216     JPanel panel = new JPanel();
217     panel.add(new JLabel("Masukkan ID yang ingin dihapus:"));
218     panel.add(idField);
219
220     // Menampilkan dialog box dengan JTextField, tombol OK, dan Cancel
221     int result = JOptionPane.showConfirmDialog(null, panel,
222         "Hapus Mahasiswa", JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);
223
224     // Jika tombol OK ditekan
225     if (result == JOptionPane.OK_OPTION) {
226         try {
227             // Mengambil input ID dan memanggil metode deleteMhs
228             int id = Integer.parseInt(idField.getText());

```

Activate Windows

```

229         controller.deleteMahasiswa(id);
230         JOptionPane.showMessageDialog(null, "Data berhasil dihapus.", "Sukses", JOptionPane.INFORMATION_MESSAGE);
231     } catch (NumberFormatException e) {
232         // Menangani error jika ID yang dimasukkan bukan angka
233         JOptionPane.showMessageDialog(null, "ID harus berupa angka.", "Error", JOptionPane.ERROR_MESSAGE);
234     }
235 }
236 loadMahasiswaTable();
237 }
238
239 public JTextField getIpkField() {
240     return ipkField;
241 }
242
243 public void setIpkField(JTextField ipkField) {
244     this.ipkField = ipkField;
245 }
246
247 public JTextField getNamaField() {
248     return namaField;
249 }
250
251 public void setNamaField(JTextField namaField) {
252     this.namaField = namaField;
253 }
254
255 public JTextField getNpmField() {
256     return npmField;
257 }
258

```

Activate Windows

Go to Settings to activate Windows.


Pom.xml

```
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-i
3     <modelVersion>4.0.0</modelVersion>
4     <groupId>com.mycompany</groupId>
5     <artifactId>rpl_shaun_springAop</artifactId>
6     <version>1.0-SNAPSHOT</version>
7     <packaging>jar</packaging>
8     <properties>
9         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
10        <maven.compiler.source>21</maven.compiler.source>
11        <maven.compiler.target>21</maven.compiler.target>
12        <exec.mainClass>com.mycompany.rpl_shaun_springaop.Rpl_shaun_springAop</exec.mainClass>
13    </properties>
14    <parent>
15        <groupId>org.springframework.boot</groupId>
16        <artifactId>spring-boot-starter-parent</artifactId>
17        <version>3.3.3</version>
18        <relativePath/>
19    </parent>
20
21    <dependencies>
22        <!-- Hibernate + Spring Data JPA -->
23        <dependency>
24            <groupId>org.springframework.boot</groupId>
25            <artifactId>spring-boot-starter-data-jpa</artifactId>
26        </dependency>
27
28        <!-- MySQL Connector -->
29        <dependency>
30            <groupId>mysql</groupId>
```

Activate Windows

```
31         <artifactId>mysql-connector-java</artifactId>
32         <version>8.0.33</version>
33     </dependency>
34
35     <!-- Spring Boot Web dependency (for MVC if needed) -->
36     <dependency>
37         <groupId>org.springframework.boot</groupId>
38         <artifactId>spring-boot-starter-web</artifactId>
39     </dependency>
40
41     <!-- Testing dependencies -->
42     <dependency>
43         <groupId>org.springframework.boot</groupId>
44         <artifactId>spring-boot-starter-test</artifactId>
45         <scope>test</scope>
46     </dependency>
47 </dependencies>
48
49 <build>
50     <plugins>
51         <plugin>
52             <groupId>org.springframework.boot</groupId>
53             <artifactId>spring-boot-maven-plugin</artifactId>
54         </plugin>
55     </plugins>
56 </build>
57 </project>
```

Output Simpan:

— □ ×

NPM

5117232

Nama

Kiki

Semester

5

IPK

5


Simpan

Buang

ID	NPM	Nama	Semester	IPK
2	51421416	Shaun	7	4.0
3	5234242	Juju	9	3.0
4	51456227	Baba	6	2.5
5	51421415	Maga	6	3.3
6	5117232	Kiki	5	5.0

Active
Go To S

Output Buang:

— □ ×

NPM

5117232

Nama

Kiki

Semester

5

IPK

5

Simpan

Buang

ID	NPM	Nama	Semester	IPK
2	51421416	Shaun	7	4.0
3	5234242	Juju	9	3.0
4	51456227	Baba	6	2.5
5	51421415	Maga	6	3.3

Active
Go to S