

## **LAPORAN AKHIR PRAKTIKUM**

Mata Praktikum : Rekayasa Perangkat Lunak 2

Kelas : 4IA26

Praktikum ke- : 5

Tanggal : 22 November 2024

Materi : AOP (ASPECT ORIENTED PROGRAMMING)

NPM : 50421002

Nama : Abdan Syakur

Ketua Asisten :

Nama Asisten :

Paraf Asisten :

Jumlah Lembar : 17



**LABORATORIUM TEKNIK INFORMATIKA**

**UNIVERSITAS GUNADARMA**

**2023/2024**

## LISTING

```
C: > Users > abdan > Documents > LABTI > Rekayasa Perangkat Lunak > Pert5 > ACT5 > src > act5 > latihan.java > latihan > saveData()
4  */
5  package act5;
6
7  import javax.swing.*;
8  import java.sql.*;
9
10 /**
11  *
12  * @author abdan
13  */
14 public class latihan extends javax.swing.JFrame {
15
16     private Connection conn;
17     private PreparedStatement pstmt;
18     private Statement stmt;
19     private ResultSet rs;
20
21
22     /**
23      * Creates new form latihan
24      */
25     public latihan() {
26         initComponents();
27         connectDatabase();
28     }
29
30     private void connectDatabase() {
31         try {
32             // Replace with your database details
33             String url = "jdbc:mysql://localhost:3306/latihan";
34             String username = "root";
35             String password = "abdan_014";
36             conn = DriverManager.getConnection(url, username, password);
37         } catch (SQLException e) {
38             JOptionPane.showMessageDialog(this, "Database connection error: " + e.getMessage());
39         }
40     }
41
```

```
43     private void saveData() {
44         String nama = jTextField1.getText();
45         String jenisLatihan = (String) jComboBox1.getSelectedItem();
46         String gerakan = jTextField2.getText();
47         String waktuLatihan = jTextField3.getText();
48
49         if (nama.isEmpty() || gerakan.isEmpty() || waktuLatihan.isEmpty()) {
50             JOptionPane.showMessageDialog(this, message: "Semua field harus diisi!", title: "Error", JOptionPane.ERROR_MESSAGE);
51         } else {
52             try {
53                 String query = "INSERT INTO latihan (nama, jenis_latihan, gerakan, waktu_latihan) VALUES (?, ?, ?, ?)";
54                 pstmt = conn.prepareStatement(query);
55                 pstmt.setString(parameterIndex:1, nama);
56                 pstmt.setString(parameterIndex:2, jenisLatihan);
57                 pstmt.setString(parameterIndex:3, gerakan);
58                 pstmt.setString(parameterIndex:4, waktuLatihan);
59                 pstmt.executeUpdate();
60                 JOptionPane.showMessageDialog(this, message: "Data berhasil disimpan!", title: "Informasi", JOptionPane.INFORMATION_MESSAGE);
61             } catch (SQLException e) {
62                 JOptionPane.showMessageDialog(this, "Error saving data: " + e.getMessage(), title: "Error", JOptionPane.ERROR_MESSAGE);
63             }
64         }
65     }
66
```

```

67 private void editData() {
68     String nama = jTextField1.getText();
69     String jenisLatihan = (String) jComboBox1.getSelectedItem();
70     String gerakan = jTextField2.getText();
71     String waktuLatihan = jTextField3.getText();
72
73     if (nama.isEmpty() || gerakan.isEmpty() || waktuLatihan.isEmpty()) {
74         JOptionPane.showMessageDialog(this, message:"Semua field harus diisi!", title:"Error", JOptionPane.ERROR_MESSAGE);
75     } else {
76         try {
77             String query = "UPDATE latihan SET jenis_latihan = ?, gerakan = ?, waktu_latihan = ? WHERE nama = ?";
78             pstmt = conn.prepareStatement(query);
79             pstmt.setString(parameterIndex:1, jenisLatihan);
80             pstmt.setString(parameterIndex:2, gerakan);
81             pstmt.setString(parameterIndex:3, waktuLatihan);
82             pstmt.setString(parameterIndex:4, nama);
83             pstmt.executeUpdate();
84             JOptionPane.showMessageDialog(this, message:"Data berhasil diperbarui!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
85         } catch (SQLException e) {
86             JOptionPane.showMessageDialog(this, "Error updating data: " + e.getMessage(), title:"Error", JOptionPane.ERROR_MESSAGE);
87         }
88     }
89 }
90

```

```

92 private void deleteData() {
93     String nama = jTextField1.getText();
94
95     if (nama.isEmpty()) {
96         JOptionPane.showMessageDialog(this, message:"Nama harus diisi!", title:"Error", JOptionPane.ERROR_MESSAGE);
97     } else {
98         try {
99             String query = "DELETE FROM latihan WHERE nama = ?";
100             pstmt = conn.prepareStatement(query);
101             pstmt.setString(parameterIndex:1, nama);
102             pstmt.executeUpdate();
103             jTextField1.setText("");
104             jTextField2.setText("");
105             jTextField3.setText("");
106             jComboBox1.setSelectedIndex(anIndex:0);
107             JOptionPane.showMessageDialog(this, message:"Data berhasil dihapus!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
108         } catch (SQLException e) {
109             JOptionPane.showMessageDialog(this, "Error deleting data: " + e.getMessage(), title:"Error", JOptionPane.ERROR_MESSAGE);
110         }
111     }
112 }
113
114 private void SimpanActionPerformed(java.awt.event.ActionEvent evt) {
115     saveData();
116 }
117
118 private void EditActionPerformed(java.awt.event.ActionEvent evt) {
119     editData();
120 }
121
122 private void HapusActionPerformed(java.awt.event.ActionEvent evt) {
123     deleteData();
124 }
125

```

```

132 @SuppressWarnings("unchecked")
133 // <editor-fold defaultstate="collapsed" desc="Generated Code">///GEN-BEGIN: initComponents
134 private void initComponents() {
135
136     jLabel1 = new javax.swing.JLabel();
137     jTextField1 = new javax.swing.JTextField();
138     jTextField2 = new javax.swing.JTextField();
139     jComboBox1 = new javax.swing.JComboBox<>();
140     jTextField3 = new javax.swing.JTextField();
141     jLabel2 = new javax.swing.JLabel();
142     jLabel3 = new javax.swing.JLabel();
143     jLabel4 = new javax.swing.JLabel();
144     jLabel5 = new javax.swing.JLabel();
145     Simpan = new javax.swing.JButton();
146     Edit = new javax.swing.JButton();
147     Hapus = new javax.swing.JButton();
148
149     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
150
151     jLabel1.setText(text:"Input Latihan");
152
153     jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Menurunkan Berat Badan", "Menaikan Berat Badan" }));
154     jComboBox1.addActionListener(new java.awt.event.ActionListener() {
155         public void actionPerformed(java.awt.event.ActionEvent evt) {
156             jComboBox1ActionPerformed(evt);
157         }
158     });
159

```



```

215 layout.setVerticalGroup(
216     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
217     .addGroup(layout.createSequentialGroup()
218         .addGap(min:44, pref:44, max:44)
219         .addComponent(jLabel1)
220         .addGap(min:51, pref:51, max:51)
221         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
222             .addGroup(layout.createSequentialGroup()
223                 .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
224                 .addGap(min:26, pref:26, max:26)
225                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
226                     .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
227                     .addComponent(jLabel2))
228                 .addGap(min:28, pref:28, max:28)
229                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
230                     .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
231                     .addComponent(jLabel4)))
232             .addComponent(jLabel3))
233         .addGap(min:35, pref:35, max:35)
234         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
235             .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
236             .addComponent(jLabel5))
237         .addGap(min:28, pref:28, max:28)
238         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
239             .addComponent(Simpan)
240             .addComponent(Edit)
241             .addComponent(Hapus))
242         .addContainerGap(pref:273, Short.MAX_VALUE))
243     );
244
245 pack();
246 } // </editor-fold> // GEN-END: initComponents

```

```

248 private void SimpanActionPerformed(java.awt.event.ActionEvent evt) {
249     String nama = jTextField1.getText();
250     String jenisLatihan = (String) jComboBox1.getSelectedItem();
251     String gerakan = jTextField2.getText();
252     String waktuLatihan = jTextField3.getText();
253
254     if (nama.isEmpty() || gerakan.isEmpty() || waktuLatihan.isEmpty()) {
255         JOptionPane.showMessageDialog(this, message:"Semua field harus diisi!", title:"Error", JOptionPane.ERROR_MESSAGE);
256     } else {
257         // Simpan data (bisa disimpan ke database atau struktur data)
258         JOptionPane.showMessageDialog(this, message:"Data berhasil disimpan!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
259     }
260 }
261
262 private void EditActionPerformed(java.awt.event.ActionEvent evt) {
263     String nama = jTextField1.getText();
264     String jenisLatihan = (String) jComboBox1.getSelectedItem();
265     String gerakan = jTextField2.getText();
266     String waktuLatihan = jTextField3.getText();
267
268     if (nama.isEmpty() || gerakan.isEmpty() || waktuLatihan.isEmpty()) {
269         JOptionPane.showMessageDialog(this, message:"Semua field harus diisi!", title:"Error", JOptionPane.ERROR_MESSAGE);
270     } else {
271         // Edit data (misalnya memperbarui data yang sudah ada)
272         JOptionPane.showMessageDialog(this, message:"Data berhasil diperbarui!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
273     }
274 }
275
276 private void HapusActionPerformed(java.awt.event.ActionEvent evt) {
277     // Clear all text fields and combo box
278     jTextField1.setText("");
279     jTextField2.setText("");
280     jTextField3.setText("");
281     jComboBox1.setSelectedIndex(anIndex:0);
282
283     JOptionPane.showMessageDialog(this, message:"Data berhasil dihapus!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
284 }

```

```

290 private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jComboBox1ActionPerformed
291     // TODO: add your handling code here:
292     String nama = jTextField1.getText();
293     String jenisLatihan = (String) jComboBox1.getSelectedItem();
294     String gerakan = jTextField2.getText();
295     String waktuLatihan = jTextField3.getText();
296
297     if (nama.isEmpty() || gerakan.isEmpty() || waktuLatihan.isEmpty()) {
298         JOptionPane.showMessageDialog(this, message:"Semua field harus diisi!", title:"Error", JOptionPane.ERROR_MESSAGE);
299     } else {
300         // Edit data (misalnya memperbarui data yang sudah ada)
301         JOptionPane.showMessageDialog(this, message:"Data berhasil diperbarui!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
302     }
303 } //GEN-LAST:event_jComboBox1ActionPerformed
304 // Action for the 'Hapus' button
305 private void HapusActionPerformed() {
306     // Clear all text fields and combo box
307     jTextField1.setText("");
308     jTextField2.setText("");
309     jTextField3.setText("");
310     jComboBox1.setSelectedIndex(anIndex:0);
311
312     JOptionPane.showMessageDialog(this, message:"Data berhasil dihapus!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
313 }
314

```

```

315  /**
316   * @param args the command line arguments
317   */
318  Run | Debug
319  public static void main(String args[]) {
320      java.awt.EventQueue.invokeLater(new Runnable() {
321          public void run() {
322              new latihan().setVisible(b:true);
323          }
324      });
325      /* Set the Nimbus look and feel */
326      <editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
327      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
328       * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
329       */
330      try {
331          for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
332              if ("Nimbus".equals(info.getName())) {
333                  javax.swing.UIManager.setLookAndFeel(info.getClassName());
334                  break;
335              }
336          } catch (ClassNotFoundException ex) {
337              java.util.logging.Logger.getLogger(latihan.class.getName()).log(java.util.logging.Level.SEVERE, msg:null, ex);
338          } catch (InstantiationException ex) {
339              java.util.logging.Logger.getLogger(latihan.class.getName()).log(java.util.logging.Level.SEVERE, msg:null, ex);
340          } catch (IllegalAccessException ex) {
341              java.util.logging.Logger.getLogger(latihan.class.getName()).log(java.util.logging.Level.SEVERE, msg:null, ex);
342          } catch (javax.swing.UnsupportedLookAndFeelException ex) {
343              java.util.logging.Logger.getLogger(latihan.class.getName()).log(java.util.logging.Level.SEVERE, msg:null, ex);
344          }
345      } </editor-fold>
346
347      /* Create and display the form */
348      java.awt.EventQueue.invokeLater(new Runnable() {
349          public void run() {
350              new latihan().setVisible(b:true);
351          }
352      });
353  }

```

```

355  // Variables declaration - do not modify//GEN-BEGIN:variables
356  private javax.swing.JButton Edit;
357  private javax.swing.JButton Hapus;
358  private javax.swing.JButton Simpan;
359  private javax.swing.JComboBox<String> jComboBox1;
360  private javax.swing.JLabel jLabel1;
361  private javax.swing.JLabel jLabel2;
362  private javax.swing.JLabel jLabel3;
363  private javax.swing.JLabel jLabel4;
364  private javax.swing.JLabel jLabel5;
365  private javax.swing.JTextField jTextField1;
366  private javax.swing.JTextField jTextField2;
367  private javax.swing.JTextField jTextField3;
368  // End of variables declaration//GEN-END:variables
369  }

```

## LOGIKA

```
C: > Users > abdan > Documents > LABTI > Rekayasa Perangkat Lunak > Pert5 > ACT5 > src > act5 > latihan.java > latihan > saveData()
4  */
5  package act5;
6
7  import javax.swing.*;
8  import java.sql.*;
9
10 /**
11  *
12  * @author abdan
13  */
14 public class latihan extends javax.swing.JFrame {
15
16     private Connection conn;
17     private PreparedStatement pstmt;
18     private Statement stmt;
19     private ResultSet rs;
20
21
22     /**
23      * Creates new form latihan
24      */
25     public latihan() {
26         initComponents();
27         connectDatabase();
28     }
29
30     private void connectDatabase() {
31         try {
32             // Replace with your database details
33             String url = "jdbc:mysql://localhost:3306/latihan";
34             String username = "root";
35             String password = "abdan_014";
36             conn = DriverManager.getConnection(url, username, password);
37         } catch (SQLException e) {
38             JOptionPane.showMessageDialog(this, "Database connection error: " + e.getMessage());
39         }
40     }
41 }
```

Kode di atas adalah program Java dengan GUI menggunakan **Swing** yang terhubung ke database MySQL. Program ini menggunakan package `act5` dengan import `javax.swing.*` untuk komponen GUI dan `java.sql.*` untuk koneksi database. Class `latihan` adalah turunan dari `JFrame` yang memiliki properti untuk koneksi database (`Connection`, `PreparedStatement`, `Statement`, dan `ResultSet`). Pada konstruktor, metode `initComponents()` digunakan untuk inisialisasi GUI, diikuti pemanggilan `connectDatabase()` untuk menghubungkan aplikasi ke database MySQL menggunakan URL, username, dan password. Jika terjadi kesalahan koneksi, akan ditampilkan pesan error melalui dialog Swing.

```

43 private void saveData() {
44     String nama = jTextField1.getText();
45     String jenisLatihan = (String) jComboBox1.getSelectedItem();
46     String gerakan = jTextField2.getText();
47     String waktuLatihan = jTextField3.getText();
48
49     if (nama.isEmpty() || gerakan.isEmpty() || waktuLatihan.isEmpty()) {
50         JOptionPane.showMessageDialog(this, message:"Semua field harus diisi!", title:"Error", JOptionPane.ERROR_MESSAGE);
51     } else {
52         try {
53             String query = "INSERT INTO latihan (nama, jenis_latihan, gerakan, waktu_latihan) VALUES (?, ?, ?, ?)";
54             pstmt = conn.prepareStatement(query);
55             pstmt.setString(parameterIndex:1, nama);
56             pstmt.setString(parameterIndex:2, jenisLatihan);
57             pstmt.setString(parameterIndex:3, gerakan);
58             pstmt.setString(parameterIndex:4, waktuLatihan);
59             pstmt.executeUpdate();
60             JOptionPane.showMessageDialog(this, message:"Data berhasil disimpan!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
61         } catch (SQLException e) {
62             JOptionPane.showMessageDialog(this, "Error saving data: " + e.getMessage(), title:"Error", JOptionPane.ERROR_MESSAGE);
63         }
64     }
65 }
66

```

Kode di atas adalah metode `saveData()` yang digunakan untuk menyimpan data dari form GUI ke dalam database. Pertama, data yang dimasukkan oleh pengguna diambil dari komponen `jTextField` dan `jComboBox`, yaitu nama, jenis latihan, gerakan, dan waktu latihan. Jika ada field yang kosong, aplikasi akan menampilkan pesan error melalui `JOptionPane`. Jika semua field terisi, program akan mencoba menyimpan data ke tabel `latihan` di database menggunakan query `INSERT INTO`. Data tersebut dimasukkan dengan menggunakan `PreparedStatement` untuk mencegah SQL injection. Setelah data berhasil disimpan, aplikasi akan menampilkan pesan konfirmasi bahwa data berhasil disimpan. Jika terjadi error, pesan error akan ditampilkan.

```

67 private void editData() {
68     String nama = jTextField1.getText();
69     String jenisLatihan = (String) jComboBox1.getSelectedItem();
70     String gerakan = jTextField2.getText();
71     String waktuLatihan = jTextField3.getText();
72
73     if (nama.isEmpty() || gerakan.isEmpty() || waktuLatihan.isEmpty()) {
74         JOptionPane.showMessageDialog(this, message:"Semua field harus diisi!", title:"Error", JOptionPane.ERROR_MESSAGE);
75     } else {
76         try {
77             String query = "UPDATE latihan SET jenis_latihan = ?, gerakan = ?, waktu_latihan = ? WHERE nama = ?";
78             pstmt = conn.prepareStatement(query);
79             pstmt.setString(parameterIndex:1, jenisLatihan);
80             pstmt.setString(parameterIndex:2, gerakan);
81             pstmt.setString(parameterIndex:3, waktuLatihan);
82             pstmt.setString(parameterIndex:4, nama);
83             pstmt.executeUpdate();
84             JOptionPane.showMessageDialog(this, message:"Data berhasil diperbarui!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
85         } catch (SQLException e) {
86             JOptionPane.showMessageDialog(this, "Error updating data: " + e.getMessage(), title:"Error", JOptionPane.ERROR_MESSAGE);
87         }
88     }
89 }
90

```

Kode di atas adalah metode `editData()` yang digunakan untuk memperbarui data yang ada di dalam database. Data yang dimasukkan oleh pengguna diambil dari komponen `jTextField` dan `jComboBox` untuk nama, jenis latihan, gerakan, dan waktu latihan. Jika ada field yang kosong, aplikasi akan menampilkan pesan error menggunakan `JOptionPane`. Jika semua field terisi, program akan mencoba menjalankan query SQL `UPDATE` untuk memperbarui data berdasarkan nama yang sudah ada di tabel `latihan`. `PreparedStatement` digunakan untuk



menghindari SQL injection. Setelah data berhasil diperbarui, aplikasi akan menampilkan pesan konfirmasi. Jika terjadi error, pesan error akan ditampilkan.

```
92 ~ private void deleteData() {
93     String nama = jTextField1.getText();
94
95 ~     if (nama.isEmpty()) {
96         JOptionPane.showMessageDialog(this, message:"Nama harus diisi!", title:"Error", JOptionPane.ERROR_MESSAGE);
97 ~     } else {
98 ~         try {
99             String query = "DELETE FROM latihan WHERE nama = ?";
100             pstmt = conn.prepareStatement(query);
101             pstmt.setString(parameterIndex:1, nama);
102             pstmt.executeUpdate();
103             jTextField1.setText("");
104             jTextField2.setText("");
105             jTextField3.setText("");
106             jComboBox1.setSelectedIndex(anIndex:0);
107             JOptionPane.showMessageDialog(this, message:"Data berhasil dihapus!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
108 ~         } catch (SQLException e) {
109             JOptionPane.showMessageDialog(this, "Error deleting data: " + e.getMessage(), title:"Error", JOptionPane.ERROR_MESSAGE);
110         }
111     }
112 }
113
114 ~ private void SimpanActionPerformed(java.awt.event.ActionEvent evt) {
115     saveData();
116 }
117
118 ~ private void EditActionPerformed(java.awt.event.ActionEvent evt) {
119     editData();
120 }
121
122 ~ private void HapusActionPerformed(java.awt.event.ActionEvent evt) {
123     deleteData();
124 }
```

Kode di atas berisi metode `deleteData()` yang digunakan untuk menghapus data dari database berdasarkan nama yang dimasukkan pengguna. Jika field nama kosong, aplikasi akan menampilkan pesan error menggunakan `JOptionPane`. Jika nama terisi, aplikasi akan menjalankan query `DELETE FROM latihan WHERE nama = ?` untuk menghapus data yang sesuai dengan nama tersebut. Setelah data dihapus, semua field akan dikosongkan dan `jComboBox` akan direset ke indeks pertama. Jika terjadi kesalahan dalam proses penghapusan, pesan error akan ditampilkan. Selain itu, ada tiga metode event handler (`SimpanActionPerformed`, `EditActionPerformed`, dan `HapusActionPerformed`) yang masing-masing memanggil metode `saveData()`, `editData()`, dan `deleteData()` saat tombol terkait diklik.

```

132 @SuppressWarnings("unchecked")
133 // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents
134 private void initComponents() {
135
136     jLabel1 = new javax.swing.JLabel();
137     jTextField1 = new javax.swing.JTextField();
138     jTextField2 = new javax.swing.JTextField();
139     jComboBox1 = new javax.swing.JComboBox<>();
140     jTextField3 = new javax.swing.JTextField();
141     jLabel2 = new javax.swing.JLabel();
142     jLabel3 = new javax.swing.JLabel();
143     jLabel4 = new javax.swing.JLabel();
144     jLabel5 = new javax.swing.JLabel();
145     Simpan = new javax.swing.JButton();
146     Edit = new javax.swing.JButton();
147     Hapus = new javax.swing.JButton();
148
149     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
150
151     jLabel1.setText(text:"Input Latihan");
152
153     jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Menurunkan Berat Badan", "Menaikan Berat Badan" }));
154     jComboBox1.addActionListener(new java.awt.event.ActionListener() {
155         public void actionPerformed(java.awt.event.ActionEvent evt) {
156             jComboBox1ActionPerformed(evt);
157         }
158     });

```

Kode di atas adalah bagian dari antarmuka pengguna (GUI) yang dibangun dengan **Swing** di Java untuk aplikasi input data latihan. Terdapat elemen-elemen seperti label `jLabel1` yang menampilkan teks "Input Latihan", tiga buah text field (`jTextField1`, `jTextField2`, dan `jTextField3`) untuk mengisi nama, gerakan, dan waktu latihan, serta sebuah combo box (`jComboBox1`) yang memungkinkan pengguna memilih jenis latihan, yakni "Menurunkan Berat Badan" atau "Menaikan Berat Badan". Selain itu, terdapat tiga tombol (`Simpan`, `Edit`, dan `Hapus`) yang masing-masing akan menjalankan fungsi untuk menyimpan, mengedit, atau menghapus data latihan. Event handler `jComboBox1ActionPerformed` menangani aksi yang terjadi ketika pengguna memilih salah satu jenis latihan dari dropdown. Kode ini menyusun elemen-elemen GUI yang akan ditampilkan dalam aplikasi.

```

160     jTextField3.addActionListener(new java.awt.event.ActionListener() {
161         public void actionPerformed(java.awt.event.ActionEvent evt) {
162             jTextField3ActionPerformed(evt);
163         }
164     });
165
166     jLabel2.setText(text:"Jenis Latihan");
167
168     jLabel3.setText(text:"Nama");
169
170     jLabel4.setText(text:"Gerakan");
171
172     jLabel5.setText(text:"Waktu Latihan");
173
174     Simpan.setText(text:"Simpan");
175
176     Edit.setText(text:"Edit");
177
178     Hapus.setText(text:"Hapus");
179

```

Kode di atas melanjutkan pembuatan antarmuka pengguna (GUI) dalam aplikasi Java menggunakan **Swing** dengan menambahkan beberapa elemen dan fungsionalitas. Sebuah `ChangeListener` ditambahkan pada `TextField3` untuk menangani aksi ketika pengguna menekan tombol "Enter" setelah mengisi field tersebut, yang akan memicu metode `TextField3ActionPerformed`. Selain itu, terdapat beberapa label yang menjelaskan field input, seperti `Label2` untuk "Jenis Latihan", `Label3` untuk "Nama", `Label4` untuk "Gerakan", dan `Label5` untuk "Waktu Latihan". Tiga tombol `Simpan`, `Edit`, dan `Hapus` diberi teks sesuai dengan fungsinya masing-masing, yaitu untuk menyimpan, mengedit, dan menghapus data. Dengan demikian, bagian ini menyelesaikan pengaturan elemen-elemen untuk form input latihan, lengkap dengan event listener dan label yang sesuai.

```

180     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
181     getContentPane().setLayout(layout);
182     layout.setHorizontalGroup(
183         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
184             .addGroup(layout.createSequentialGroup()
185                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
186                     .addGroup(layout.createSequentialGroup()
187                         .addGap(min:141, pref:141, max:141)
188                         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, resizable:false)
189                             .addGroup(layout.createSequentialGroup()
190                                 .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, pref:37, javax.swing.GroupLayout.PREFERRED_SIZE)
191                                 .addGap(min:131, pref:131, max:131)
192                                 .addComponent(jTextField1)
193                             )
194                             .addGroup(layout.createSequentialGroup()
195                                 .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, pref:75, javax.swing.GroupLayout.PREFERRED_SIZE)
196                                 .addComponent(jLabel4)
197                                 .addComponent(jLabel5)
198                                 .addGap(min:92, pref:92, max:92)
199                                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
200                                     .addComponent(jComboBox1, min:0, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
201                                     .addComponent(jTextField2)
202                                     .addComponent(jTextField3)
203                                 )
204                                 .addGroup(layout.createSequentialGroup()
205                                     .addComponent(Simpan)
206                                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
207                                     .addComponent(Edit)
208                                     .addGap(min:12, pref:12, max:12)
209                                     .addComponent(Hapus)
210                                     .addGap(min:0, pref:0, Short.MAX_VALUE))
211                                 )
212                             )
213                         .addGap(min:422, pref:422, max:422)
214                         .addComponent(jLabel1))
215                 .addContainerGap(pref:650, Short.MAX_VALUE))
216     );

```

Kode di atas adalah bagian dari pengaturan layout untuk elemen-elemen GUI di aplikasi Java menggunakan **GroupLayout**. `GroupLayout` adalah layout manager yang digunakan untuk mengatur posisi dan ukuran komponen dalam antarmuka pengguna secara fleksibel. Dalam kode ini, layout diatur secara horizontal (dengan `layout.setHorizontalGroup`) untuk berbagai komponen seperti label, text field, combo box, dan tombol. Beberapa komponen yang diatur termasuk `Label3` (label "Nama"), `TextField1` (field untuk nama), `Label2`, `Label4`, dan `Label5` (label untuk jenis latihan, gerakan, dan waktu latihan), serta `ComboBox1` (combo box untuk memilih jenis latihan), `TextField2` dan `TextField3` (untuk gerakan dan waktu latihan). Di bagian bawah, tombol `Simpan`, `Edit`, dan `Hapus` diatur dalam satu baris. Pengaturan ini memastikan bahwa semua elemen GUI ditata dengan baik di dalam jendela aplikasi dengan jarak dan posisi yang teratur.

```

215 layout.setVerticalGroup(
216     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
217     .addGroup(layout.createSequentialGroup()
218         .addGap(min:44, pref:44, max:44)
219         .addComponent(jLabel1)
220         .addGap(min:51, pref:51, max:51)
221         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
222             .addGroup(layout.createSequentialGroup()
223                 .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
224                 .addGap(min:26, pref:26, max:26)
225                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
226                     .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
227                     .addComponent(jLabel2))
228                 .addGap(min:28, pref:28, max:28)
229                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
230                     .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
231                     .addComponent(jLabel4))
232                 .addComponent(jLabel3))
233             .addGap(min:35, pref:35, max:35)
234             .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
235                 .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
236                 .addComponent(jLabel5))
237             .addGap(min:28, pref:28, max:28)
238             .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
239                 .addComponent(Simpan)
240                 .addComponent(Edit)
241                 .addComponent(Hapus))
242             .addContainerGap(pref:273, Short.MAX_VALUE))
243     );
244
245 pack();
246 } // </editor-fold> // GEN-END: initComponents

```

Kode di atas adalah bagian dari pengaturan layout vertikal untuk elemen-elemen GUI menggunakan  **GroupLayout**  di aplikasi Java. Dalam kode ini, `layout.setVerticalGroup` digunakan untuk mengatur posisi vertikal dari berbagai komponen seperti label, text field, combo box, dan tombol. Elemen-elemen tersebut diatur secara berurutan dengan jarak antar elemen yang disesuaikan menggunakan `addGap` untuk memberikan ruang antar elemen. Dimulai dengan `jLabel1` (judul aplikasi), diikuti oleh input untuk nama (`jTextField1`), jenis latihan (`jComboBox1`), gerakan (`jTextField2`), dan waktu latihan (`jTextField3`). Setiap elemen input diatur agar sesuai dengan label yang relevan seperti `jLabel2`, `jLabel4`, dan `jLabel5`. Setelah input, ada tiga tombol (`Simpan`, `Edit`, dan `Hapus`) yang disusun secara horizontal. Secara keseluruhan, layout vertikal ini memastikan elemen-elemen GUI ditata dengan rapi dari atas ke bawah dalam jendela aplikasi, dengan jarak yang memadai antara masing-masing elemen untuk meningkatkan keterbacaan dan kenyamanan pengguna.

```

248 private void SimpanActionPerformed(java.awt.event.ActionEvent evt) {
249     String nama = jTextField1.getText();
250     String jenisLatihan = (String) jComboBox1.getSelectedItem();
251     String gerakan = jTextField2.getText();
252     String waktuLatihan = jTextField3.getText();
253
254     if (nama.isEmpty() || gerakan.isEmpty() || waktuLatihan.isEmpty()) {
255         JOptionPane.showMessageDialog(this, message:"Semua field harus diisi!", title:"Error", JOptionPane.ERROR_MESSAGE);
256     } else {
257         // Simpan data (bisa disimpan ke database atau struktur data)
258         JOptionPane.showMessageDialog(this, message:"Data berhasil disimpan!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
259     }
260 }
261
262 private void EditActionPerformed(java.awt.event.ActionEvent evt) {
263     String nama = jTextField1.getText();
264     String jenisLatihan = (String) jComboBox1.getSelectedItem();
265     String gerakan = jTextField2.getText();
266     String waktuLatihan = jTextField3.getText();
267
268     if (nama.isEmpty() || gerakan.isEmpty() || waktuLatihan.isEmpty()) {
269         JOptionPane.showMessageDialog(this, message:"Semua field harus diisi!", title:"Error", JOptionPane.ERROR_MESSAGE);
270     } else {
271         // Edit data (misalnya memperbarui data yang sudah ada)
272         JOptionPane.showMessageDialog(this, message:"Data berhasil diperbarui!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
273     }
274 }
275
276 private void HapusActionPerformed(java.awt.event.ActionEvent evt) {
277     // Clear all text fields and combo box
278     jTextField1.setText("");
279     jTextField2.setText("");
280     jTextField3.setText("");
281     jComboBox1.setSelectedIndex(0);
282
283     JOptionPane.showMessageDialog(this, message:"Data berhasil dihapus!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
284 }

```

Kode di atas mengimplementasikan tiga aksi tombol dalam aplikasi Java menggunakan `JOptionPane` untuk menampilkan pesan informasi atau error. Pada aksi **SimpanActionPerformed**, data diambil dari `jTextField1`, `jComboBox1`, `jTextField2`, dan `jTextField3` (nama, jenis latihan, gerakan, dan waktu latihan). Jika ada field yang kosong, muncul pesan error, jika tidak, ditampilkan pesan bahwa data berhasil disimpan (meskipun hanya disimpan dalam memori, bukan database). Pada aksi **EditActionPerformed**, data juga diambil dari field input, diperiksa apakah ada yang kosong, dan jika terisi lengkap, ditampilkan pesan bahwa data berhasil diperbarui (meskipun tanpa implementasi nyata untuk pembaruan). Sedangkan pada **HapusActionPerformed**, seluruh teks di `jTextField1`, `jTextField2`, `jTextField3` dihapus, dan `jComboBox1` disetel ke index pertama, lalu pesan bahwa data berhasil dihapus ditampilkan. Ketiga aksi ini memberikan fungsionalitas untuk menangani penyimpanan, pengeditan, dan penghapusan data melalui antarmuka pengguna.

```

290 private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_jComboBox1ActionPerformed
291     // TODO add your handling code here:
292     String nama = jTextField1.getText();
293     String jenisLatihan = (String) jComboBox1.getSelectedItem();
294     String gerakan = jTextField2.getText();
295     String waktuLatihan = jTextField3.getText();
296
297     if (nama.isEmpty() || gerakan.isEmpty() || waktuLatihan.isEmpty()) {
298         JOptionPane.showMessageDialog(this, message:"Semua field harus diisi!", title:"Error", JOptionPane.ERROR_MESSAGE);
299     } else {
300         // Edit data (misalnya memperbarui data yang sudah ada)
301         JOptionPane.showMessageDialog(this, message:"Data berhasil diperbarui!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
302     }
303 }GEN-LAST:event_jComboBox1ActionPerformed
304 // Action for the 'Hapus' button
305 private void HapusActionPerformed() {
306     // Clear all text fields and combo box
307     jTextField1.setText("");
308     jTextField2.setText("");
309     jTextField3.setText("");
310     jComboBox1.setSelectedIndex(0);
311
312     JOptionPane.showMessageDialog(this, message:"Data berhasil dihapus!", title:"Informasi", JOptionPane.INFORMATION_MESSAGE);
313 }
314

```

Kode diatas berikan berisi dua metode untuk menangani aksi pengguna dalam aplikasi Java. Pada metode `jComboBox1ActionPerformed`, ketika pengguna memilih item dari `jComboBox1`, program akan mengambil nilai dari `jTextField1`, `jTextField2`, `jTextField3`, dan pilihan `jComboBox1`. Jika ada field yang kosong, akan muncul pesan error yang meminta pengguna untuk mengisi semua field, namun jika semua field terisi, pesan informasi akan muncul yang mengatakan bahwa data berhasil diperbarui. Pada metode `HapusActionPerformed`, ketika dipanggil, metode ini akan menghapus isi dari `jTextField1`, `jTextField2`, `jTextField3`, dan mengatur `jComboBox1` kembali ke nilai default (indeks pertama). Setelah itu, pesan akan muncul yang memberi tahu pengguna bahwa data telah berhasil dihapus.

```

315  /**
316   * @param args the command line arguments
317   */
318  public static void main(String args[]) {
319      java.awt.EventQueue.invokeLater(new Runnable() {
320          public void run() {
321              new latihan().setVisible(b:true);
322          }
323      });
324      /* Set the Nimbus look and feel */
325      //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
326      /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
327       * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
328       */
329      try {
330          for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
331              if ("Nimbus".equals(info.getName())) {
332                  javax.swing.UIManager.setLookAndFeel(info.getClassName());
333                  break;
334              }
335          }
336      } catch (ClassNotFoundException ex) {
337          java.util.logging.Logger.getLogger(latihan.class.getName()).log(java.util.logging.Level.SEVERE, msg:null, ex);
338      } catch (InstantiationException ex) {
339          java.util.logging.Logger.getLogger(latihan.class.getName()).log(java.util.logging.Level.SEVERE, msg:null, ex);
340      } catch (IllegalAccessException ex) {
341          java.util.logging.Logger.getLogger(latihan.class.getName()).log(java.util.logging.Level.SEVERE, msg:null, ex);
342      } catch (javax.swing.UnsupportedLookAndFeelException ex) {
343          java.util.logging.Logger.getLogger(latihan.class.getName()).log(java.util.logging.Level.SEVERE, msg:null, ex);
344      }
345      //</editor-fold>
346
347      /* Create and display the form */
348      java.awt.EventQueue.invokeLater(new Runnable() {
349          public void run() {
350              new latihan().setVisible(b:true);
351          }
352      });
353

```

Pada kode ini, metode `main` berfungsi untuk menjalankan aplikasi Java Swing. Pertama, kode ini mengatur tampilan antarmuka grafis menggunakan \*Look and Feel\* Nimbus, jika tersedia, dengan mencoba mengatur tampilan menggunakan `UIManager.setLookAndFeel`. Jika Nimbus tidak tersedia atau terjadi kesalahan, aplikasi akan tetap menggunakan tampilan default. Setelah itu, aplikasi akan menampilkan form `latihan` (kelas yang mungkin merupakan form GUI aplikasi ini) dengan membuat instance baru dan memanggil metode `setVisible(true)` di dalam \*Event Dispatch Thread\* menggunakan `EventQueue.invokeLater`. Ini memastikan bahwa form akan dibuka di thread yang aman untuk pembaruan antarmuka pengguna.

```

355 // Variables declaration - do not modify//GEN-BEGIN:variables
356 private javax.swing.JButton Edit;
357 private javax.swing.JButton Hapus;
358 private javax.swing.JButton Simpan;
359 private javax.swing.JComboBox<String> jComboBox1;
360 private javax.swing.JLabel jLabel1;
361 private javax.swing.JLabel jLabel2;
362 private javax.swing.JLabel jLabel3;
363 private javax.swing.JLabel jLabel4;
364 private javax.swing.JLabel jLabel5;
365 private javax.swing.JTextField jTextField1;
366 private javax.swing.JTextField jTextField2;
367 private javax.swing.JTextField jTextField3;
368 // End of variables declaration//GEN-END:variables
369 }

```

Bagian kode ini mendeklarasikan variabel-variabel yang digunakan untuk elemen-elemen GUI (Graphical User Interface) dalam kelas `latihan`. Variabel-variabel ini adalah komponen-komponen antarmuka pengguna yang mencakup tombol-tombol (`Edit`, `Hapus`, `Simpan`), sebuah \*combo box\* (`jComboBox1`), label-label (`jLabel1`, `jLabel2`, `jLabel3`, `jLabel4`, `jLabel5`), serta \*text fields\* untuk input teks (`jTextField1`, `jTextField2`, `jTextField3`). Komentar `//GEN-BEGIN` dan `//GEN-END` menunjukkan bagian kode yang dihasilkan secara otomatis oleh \*GUI designer\* (misalnya, NetBeans) dan biasanya tidak perlu diubah.



## OUTPUT

Output - ACT5 (run) #2

run:  
BUILD SUCCESSFUL (total time: 24 seconds)

Input Latihan

Nama

abdan syakur

Jenis Latihan

Menurunkan Berat Badan

Gerakan

push up

Waktu Latihan

tujuh hari tujuh malam

Simpan

Edit

Hapus