**LISTING PROGRAM**

1. Data Karyawan
2. /\*
3. \* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
4. \* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JInternalFrame.java to edit this template
5. \*/
6. package Form;
7. import Koneksi.KoneksiDb;
8. import java.awt.event.KeyEvent;
9. import java.sql.Connection;
10. import java.sql.ResultSet;
11. import java.sql.Statement;
12. import java.text.DateFormat;
13. import java.text.SimpleDateFormat;
14. import java.sql.Date;
15. import java.sql.SQLException;
16. import javax.swing.JOptionPane;
17. import javax.swing.plaf.basic.BasicInternalFrameUI;
18. import javax.swing.table.DefaultTableModel;
19. /\*\*
20. \*
21. \* @author resar
22. \*/
23. public class Karyawan extends javax.swing.JInternalFrame {
24. /\*\*
25. \* Creates new form Profile
26. \*/
27. KoneksiDb k = new KoneksiDb();
28. Connection con;
29. String nipGuru = "";
30. DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");
31. public Karyawan() throws SQLException {
32. this.con = k.getConnect();
33. initComponents();
34. this.setBorder(javax.swing.BorderFactory.createEmptyBorder(0,0,0,0));
35. BasicInternalFrameUI ui = (BasicInternalFrameUI)this.getUI();
36. ui.setNorthPane(null);
37. showGuru();
38. tblGuru.setAutoCreateRowSorter(true);
39. }
41. public void showGuru(){
42. DefaultTableModel model = (DefaultTableModel) tblGuru.getModel();
44. Statement st;
45. ResultSet rs;
47. String query = "SELECT \* FROM tbl\_alternatif";
48. String status = null;
50. // String s2[] = { "male", "female", "others" };
51. // cbJk = new JComboBox(s2);
52. //
53. try {
54. st = con.createStatement();
55. rs = st.executeQuery(query);
57. Object[] row = new Object[4];
58. int no = 0;
59. while(rs.next()){
60. //row[0] = ++no;
61. row[0] = rs.getString("id\_alternatif");
62. row[1] = rs.getString("nama");
63. row[2] = rs.getString("alamat");
64. row[3] = rs.getString("telp");
65. model.addRow(row);
66. }
67. } catch (Exception e) {
68. e.printStackTrace();
69. }
70. }


74. public void RefreshPage(){
75. DefaultTableModel model = (DefaultTableModel)tblGuru.getModel();
76. model.setRowCount(0);
77. showGuru();
78. bersih();
79. }
80. /\*\*
81. \* This method is called from within the constructor to initialize the form.
82. \* WARNING: Do NOT modify this code. The content of this method is always
83. \* regenerated by the Form Editor.
84. \*/
85. @SuppressWarnings("unchecked")
86. // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
87. private void initComponents() {
88. jScrollPane1 = new javax.swing.JScrollPane();
89. tblGuru = new javax.swing.JTable();
90. btnAdd = new javax.swing.JButton();
91. btnEdit = new javax.swing.JButton();
92. btnDelete = new javax.swing.JButton();
93. lblNIP = new javax.swing.JLabel();
94. tfNIP = new javax.swing.JTextField();
95. lblNama = new javax.swing.JLabel();
96. tfNama = new javax.swing.JTextField();
97. lblAgama = new javax.swing.JLabel();
98. tfNoTlp = new javax.swing.JTextField();
99. lblNoTelp = new javax.swing.JLabel();
100. btnReset = new javax.swing.JButton();
101. jScrollPane2 = new javax.swing.JScrollPane();
102. tfAgama = new javax.swing.JTextArea();
103. setFocusCycleRoot(false);
104. setFocusable(false);
105. setPreferredSize(new java.awt.Dimension(1200, 580));
106. setRequestFocusEnabled(false);
107. tblGuru.setAutoCreateRowSorter(true);
108. tblGuru.setBorder(javax.swing.BorderFactory.createCompoundBorder());
109. tblGuru.setModel(new javax.swing.table.DefaultTableModel(
110. new Object [][] {
111. },
112. new String [] {
113. "ID", "NAMA", "ALAMAT", "NO TELP"
114. }
115. ));
116. tblGuru.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
117. tblGuru.addMouseListener(new java.awt.event.MouseAdapter() {
118. public void mouseClicked(java.awt.event.MouseEvent evt) {
119. tblGuruMouseClicked(evt);
120. }
121. });
122. jScrollPane1.setViewportView(tblGuru);
123. btnAdd.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
124. btnAdd.setText("ADD");
125. btnAdd.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
126. btnAdd.addActionListener(new java.awt.event.ActionListener() {
127. public void actionPerformed(java.awt.event.ActionEvent evt) {
128. btnAddActionPerformed(evt);
129. }
130. });
131. btnEdit.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
132. btnEdit.setText("EDIT");
133. btnEdit.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
134. btnEdit.addActionListener(new java.awt.event.ActionListener() {
135. public void actionPerformed(java.awt.event.ActionEvent evt) {
136. btnEditActionPerformed(evt);
137. }
138. });
139. btnDelete.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
140. btnDelete.setText("DELETE");
141. btnDelete.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
142. btnDelete.addActionListener(new java.awt.event.ActionListener() {
143. public void actionPerformed(java.awt.event.ActionEvent evt) {
144. btnDeleteActionPerformed(evt);
145. }
146. });
147. lblNIP.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
148. lblNIP.setText("ID :");
149. lblNama.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
150. lblNama.setText("NAMA:");
151. lblAgama.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
152. lblAgama.setText("ALAMAT:");
153. tfNoTlp.addKeyListener(new java.awt.event.KeyAdapter() {
154. public void keyTyped(java.awt.event.KeyEvent evt) {
155. tfNoTlpKeyTyped(evt);
156. }
157. });
158. lblNoTelp.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
159. lblNoTelp.setText("NO TELP:");
160. btnReset.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
161. btnReset.setText("RESET");
162. btnReset.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
163. btnReset.addActionListener(new java.awt.event.ActionListener() {
164. public void actionPerformed(java.awt.event.ActionEvent evt) {
165. btnResetActionPerformed(evt);
166. }
167. });
168. tfAgama.setColumns(20);
169. tfAgama.setRows(5);
170. jScrollPane2.setViewportView(tfAgama);
171. javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
172. getContentPane().setLayout(layout);
173. layout.setHorizontalGroup(
174. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
175. .addGroup(layout.createSequentialGroup()
176. .addGap(30, 30, 30)
177. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
178. .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 662, javax.swing.GroupLayout.PREFERRED\_SIZE)
179. .addGroup(layout.createSequentialGroup()
180. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
181. .addGroup(layout.createSequentialGroup()
182. .addComponent(lblNama)
183. .addGap(32, 32, 32)
184. .addComponent(tfNama, javax.swing.GroupLayout.PREFERRED\_SIZE, 119, javax.swing.GroupLayout.PREFERRED\_SIZE))
185. .addGroup(layout.createSequentialGroup()
186. .addComponent(lblNIP)
187. .addGap(52, 52, 52)
188. .addComponent(tfNIP, javax.swing.GroupLayout.PREFERRED\_SIZE, 119, javax.swing.GroupLayout.PREFERRED\_SIZE)
189. .addGap(31, 31, 31)
190. .addComponent(lblAgama)))
191. .addGap(10, 10, 10)
192. .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)
193. .addGap(18, 18, 18)
194. .addComponent(lblNoTelp)
195. .addGap(18, 18, 18)
196. .addComponent(tfNoTlp, javax.swing.GroupLayout.PREFERRED\_SIZE, 119, javax.swing.GroupLayout.PREFERRED\_SIZE))
197. .addGroup(layout.createSequentialGroup()
198. .addComponent(btnAdd, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE)
199. .addGap(18, 18, 18)
200. .addComponent(btnEdit, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE)
201. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
202. .addComponent(btnDelete, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE)
203. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
204. .addComponent(btnReset, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE)))
205. .addGap(0, 486, Short.MAX\_VALUE))
206. );
207. layout.setVerticalGroup(
208. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
209. .addGroup(layout.createSequentialGroup()
210. .addGap(20, 20, 20)
211. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
212. .addGroup(layout.createSequentialGroup()
213. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
214. .addComponent(lblNIP)
215. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
216. .addComponent(tfNIP, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)
217. .addComponent(lblAgama))
218. .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
219. .addComponent(lblNoTelp)
220. .addComponent(tfNoTlp, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))
221. .addGap(13, 13, 13)
222. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
223. .addComponent(lblNama)
224. .addComponent(tfNama, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))
225. .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))
226. .addGap(28, 28, 28)
227. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
228. .addComponent(btnAdd)
229. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
230. .addComponent(btnEdit)
231. .addComponent(btnDelete)
232. .addComponent(btnReset)))
233. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
234. .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 190, javax.swing.GroupLayout.PREFERRED\_SIZE)
235. .addContainerGap(64, Short.MAX\_VALUE))
236. );
237. pack();
238. }// </editor-fold>//GEN-END:initComponents
239. private void tblGuruMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_tblGuruMouseClicked
240. DefaultTableModel model = (DefaultTableModel)tblGuru.getModel();
241. int i = tblGuru.getSelectedRow();
242. String a = model.getValueAt(i, 0).toString();
243. String b = model.getValueAt(i, 1).toString();
244. String c = model.getValueAt(i, 2).toString();
245. String d = model.getValueAt(i, 3).toString();
247. tfNIP.setText(a);
248. tfNama.setText(b);
249. tfAgama.setText(c);
250. tfNoTlp.setText(d);
251. // tfNPM.setText(model.getValueAt(i, 0).toString());
252. // tfNama.setText(model.getValueAt(i, 1).toString());
253. // cbAngkatan.getModel().setSelectedItem(model.getValueAt(i, 2).toString());
254. }//GEN-LAST:event\_tblGuruMouseClicked
255. private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnAddActionPerformed
256. // TODO add your handling code here:
257. if(checkField() == 1){
258. try {
259. Statement st = con.createStatement();
260. String sql = "INSERT into tbl\_alternatif (id\_alternatif, nama, alamat,telp) values ('"+tfNIP.getText()+"', '"+Utility.capitalizeWord(tfNama.getText())+"', '"+Utility.capitalizeWord(tfAgama.getText())+"', '"+tfNoTlp.getText()+"')";
261. st.executeUpdate(sql);
262. RefreshPage();
263. } catch (SQLException e) {
264. JOptionPane.showMessageDialog(null, e);
265. }
267. } else{
268. JOptionPane.showMessageDialog(this, "Isi Semua Field!");
269. }
270. }//GEN-LAST:event\_btnAddActionPerformed
271. private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnEditActionPerformed
272. // TODO add your handling code here:
273. if(tfNIP.equals("")){
274. JOptionPane.showMessageDialog(this, "Pilih Data Yang Ingin Di Edit!");
275. } else{
276. int conf;
277. conf = JOptionPane.showConfirmDialog(null, "Apakah Anda Yakin Ingin Mengedit Data Ini?"+tfNIP.getText(),
278. "Warning",JOptionPane.WARNING\_MESSAGE);
279. try {
280. Statement st = con.createStatement();
281. String sql = "UPDATE `tbl\_alternatif` SET `nama` = '" +Utility.capitalizeWord(tfNama.getText())+"', "+
282. "`alamat` = '"+Utility.capitalizeWord(tfAgama.getText())+"', "+
283. "`telp` = '"+tfNoTlp.getText()+"' "+
284. "WHERE `id\_alternatif` ='"+tfNIP.getText()+"'";
285. st.executeUpdate(sql);
286. RefreshPage();
287. } catch (SQLException e) {
288. JOptionPane.showMessageDialog(null, e);
289. }
290. }
291. }//GEN-LAST:event\_btnEditActionPerformed
292. private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnDeleteActionPerformed
293. // TODO add your handling code here:
294. if(tfNIP.equals(""))
295. {
296. JOptionPane.showMessageDialog(this, "Pilih Data Yang Ingin Di Hapus!");
297. }
298. else{
299. int conf;
300. conf = JOptionPane.showConfirmDialog(null, "Apakah Anda Yakin Ingin Menghapus Data Ini?",
301. "Warning",JOptionPane.WARNING\_MESSAGE);
302. if(conf == 0){
303. try {
304. Statement st = con.createStatement();
305. String sql = "DELETE from tbl\_alternatif where id\_alternatif ='"+tfNIP.getText()+"'";
306. st.executeUpdate(sql);
307. RefreshPage();
308. } catch (SQLException e) {
309. JOptionPane.showMessageDialog(null, e);
310. }
311. }
312. }
313. }//GEN-LAST:event\_btnDeleteActionPerformed
314. private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnResetActionPerformed
315. // TODO add your handling code here:
316. tfNIP.setText("");
317. tfNama.setText("");
318. tfAgama.setText("");
319. tfNoTlp.setText("");
320. }//GEN-LAST:event\_btnResetActionPerformed
321. private void tfNoTlpKeyTyped(java.awt.event.KeyEvent evt) {//GEN-FIRST:event\_tfNoTlpKeyTyped
322. char c = evt.getKeyChar();
323. if ( ((c < '0') || (c > '9')) && (c != KeyEvent.VK\_BACK\_SPACE)) {
324. evt.consume();
325. }
326. }//GEN-LAST:event\_tfNoTlpKeyTyped
327. public void bersih(){
328. tfNIP.setText("");
329. tfNama.setText("");
330. tfAgama.setText("");
331. tfNoTlp.setText("");
332. }
334. public int checkField(){
335. if(
336. tfNIP.getText().trim().isEmpty() ||
337. tfNama.getText().trim().isEmpty() ||
338. tfAgama.getText().trim().isEmpty() ||
339. tfNoTlp.getText().trim().isEmpty()
340. ) {
341. return 0;
342. } else{
343. return 1;
344. }
345. }
346. // Variables declaration - do not modify//GEN-BEGIN:variables
347. private javax.swing.JButton btnAdd;
348. private javax.swing.JButton btnDelete;
349. private javax.swing.JButton btnEdit;
350. private javax.swing.JButton btnReset;
351. private javax.swing.JScrollPane jScrollPane1;
352. private javax.swing.JScrollPane jScrollPane2;
353. private javax.swing.JLabel lblAgama;
354. private javax.swing.JLabel lblNIP;
355. private javax.swing.JLabel lblNama;
356. private javax.swing.JLabel lblNoTelp;
357. private javax.swing.JTable tblGuru;
358. private javax.swing.JTextArea tfAgama;
359. private javax.swing.JTextField tfNIP;
360. private javax.swing.JTextField tfNama;
361. private javax.swing.JTextField tfNoTlp;
362. // End of variables declaration//GEN-END:variables
363. }
364. Data Kriteria
365. /\*
366. \* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
367. \* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JInternalFrame.java to edit this template
368. \*/
369. package Form;
370. import Koneksi.KoneksiDb;
371. import java.awt.event.KeyEvent;
372. import java.sql.Connection;
373. import java.sql.ResultSet;
374. import java.sql.Statement;
375. import java.text.DateFormat;
376. import java.text.SimpleDateFormat;
377. import java.sql.Date;
378. import java.sql.SQLException;
379. import javax.swing.JOptionPane;
380. import javax.swing.plaf.basic.BasicInternalFrameUI;
381. import javax.swing.table.DefaultTableModel;
382. /\*\*
383. \*
384. \* @author resar
385. \*/
386. public class Kriteria extends javax.swing.JInternalFrame {
387. /\*\*
388. \* Creates new form Profile
389. \*/
390. KoneksiDb k = new KoneksiDb();
391. Connection con;
392. String nipGuru = "";
393. DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy");
394. public Kriteria() throws SQLException {
395. this.con = k.getConnect();
396. initComponents();
397. this.setBorder(javax.swing.BorderFactory.createEmptyBorder(0,0,0,0));
398. BasicInternalFrameUI ui = (BasicInternalFrameUI)this.getUI();
399. ui.setNorthPane(null);
400. showGuru();
401. tblGuru.setAutoCreateRowSorter(true);
402. }
404. public void showGuru(){
405. DefaultTableModel model = (DefaultTableModel) tblGuru.getModel();
407. Statement st;
408. ResultSet rs;
410. String query = "SELECT \* FROM tbl\_kriteria";
411. String status = null;
413. // String s2[] = { "male", "female", "others" };
414. // cbJk = new JComboBox(s2);
415. //
416. try {
417. st = con.createStatement();
418. rs = st.executeQuery(query);
420. Object[] row = new Object[4];
421. int no = 0;
422. while(rs.next()){
423. //row[0] = ++no;
424. row[0] = rs.getString("id\_kriteria");
425. row[1] = rs.getString("nama\_kriteria");
426. row[2] = rs.getString("bobot");
427. row[3] = rs.getString("flag");
428. model.addRow(row);
429. }
430. } catch (Exception e) {
431. e.printStackTrace();
432. }
433. }


437. public void RefreshPage(){
438. DefaultTableModel model = (DefaultTableModel)tblGuru.getModel();
439. model.setRowCount(0);
440. showGuru();
441. bersih();
442. }
443. /\*\*
444. \* This method is called from within the constructor to initialize the form.
445. \* WARNING: Do NOT modify this code. The content of this method is always
446. \* regenerated by the Form Editor.
447. \*/
448. @SuppressWarnings("unchecked")
449. // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
450. private void initComponents() {
451. jScrollPane1 = new javax.swing.JScrollPane();
452. tblGuru = new javax.swing.JTable();
453. btnAdd = new javax.swing.JButton();
454. btnEdit = new javax.swing.JButton();
455. btnDelete = new javax.swing.JButton();
456. lblNIP = new javax.swing.JLabel();
457. tfNIP = new javax.swing.JTextField();
458. lblNama = new javax.swing.JLabel();
459. tfNama = new javax.swing.JTextField();
460. lblAgama = new javax.swing.JLabel();
461. lblNoTelp = new javax.swing.JLabel();
462. btnReset = new javax.swing.JButton();
463. jScrollPane2 = new javax.swing.JScrollPane();
464. tfAgama = new javax.swing.JTextArea();
465. tfNoTlp = new javax.swing.JComboBox<>();
466. setFocusCycleRoot(false);
467. setFocusable(false);
468. setPreferredSize(new java.awt.Dimension(1200, 580));
469. setRequestFocusEnabled(false);
470. tblGuru.setAutoCreateRowSorter(true);
471. tblGuru.setBorder(javax.swing.BorderFactory.createCompoundBorder());
472. tblGuru.setModel(new javax.swing.table.DefaultTableModel(
473. new Object [][] {
474. },
475. new String [] {
476. "ID", "NAMA", "BOBOT", "TIPE"
477. }
478. ));
479. tblGuru.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
480. tblGuru.addMouseListener(new java.awt.event.MouseAdapter() {
481. public void mouseClicked(java.awt.event.MouseEvent evt) {
482. tblGuruMouseClicked(evt);
483. }
484. });
485. jScrollPane1.setViewportView(tblGuru);
486. btnAdd.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
487. btnAdd.setText("ADD");
488. btnAdd.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
489. btnAdd.addActionListener(new java.awt.event.ActionListener() {
490. public void actionPerformed(java.awt.event.ActionEvent evt) {
491. btnAddActionPerformed(evt);
492. }
493. });
494. btnEdit.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
495. btnEdit.setText("EDIT");
496. btnEdit.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
497. btnEdit.addActionListener(new java.awt.event.ActionListener() {
498. public void actionPerformed(java.awt.event.ActionEvent evt) {
499. btnEditActionPerformed(evt);
500. }
501. });
502. btnDelete.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
503. btnDelete.setText("DELETE");
504. btnDelete.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
505. btnDelete.addActionListener(new java.awt.event.ActionListener() {
506. public void actionPerformed(java.awt.event.ActionEvent evt) {
507. btnDeleteActionPerformed(evt);
508. }
509. });
510. lblNIP.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
511. lblNIP.setText("ID:");
512. lblNama.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
513. lblNama.setText("NAMA:");
514. lblAgama.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
515. lblAgama.setText("BOBOT:");
516. lblNoTelp.setFont(new java.awt.Font("Segoe UI", 0, 14)); // NOI18N
517. lblNoTelp.setText("TIPE:");
518. btnReset.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
519. btnReset.setText("RESET");
520. btnReset.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
521. btnReset.addActionListener(new java.awt.event.ActionListener() {
522. public void actionPerformed(java.awt.event.ActionEvent evt) {
523. btnResetActionPerformed(evt);
524. }
525. });
526. tfAgama.setColumns(20);
527. tfAgama.setRows(5);
528. jScrollPane2.setViewportView(tfAgama);
529. tfNoTlp.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "benefit", "cost" }));
530. javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
531. getContentPane().setLayout(layout);
532. layout.setHorizontalGroup(
533. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
534. .addGroup(layout.createSequentialGroup()
535. .addGap(30, 30, 30)
536. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
537. .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 662, javax.swing.GroupLayout.PREFERRED\_SIZE)
538. .addGroup(layout.createSequentialGroup()
539. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
540. .addGroup(layout.createSequentialGroup()
541. .addComponent(lblNama)
542. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)
543. .addComponent(tfNama, javax.swing.GroupLayout.PREFERRED\_SIZE, 119, javax.swing.GroupLayout.PREFERRED\_SIZE))
544. .addGroup(layout.createSequentialGroup()
545. .addComponent(lblNIP)
546. .addGap(52, 52, 52)
547. .addComponent(tfNIP, javax.swing.GroupLayout.PREFERRED\_SIZE, 119, javax.swing.GroupLayout.PREFERRED\_SIZE)))
548. .addGap(31, 31, 31)
549. .addComponent(lblAgama)
550. .addGap(10, 10, 10)
551. .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)
552. .addGap(18, 18, 18)
553. .addComponent(lblNoTelp)
554. .addGap(18, 18, 18)
555. .addComponent(tfNoTlp, javax.swing.GroupLayout.PREFERRED\_SIZE, 142, javax.swing.GroupLayout.PREFERRED\_SIZE))
556. .addGroup(layout.createSequentialGroup()
557. .addComponent(btnAdd, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE)
558. .addGap(18, 18, 18)
559. .addComponent(btnEdit, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE)
560. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
561. .addComponent(btnDelete, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE)
562. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
563. .addComponent(btnReset, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE)))
564. .addGap(0, 22, Short.MAX\_VALUE))
565. );
566. layout.setVerticalGroup(
567. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
568. .addGroup(layout.createSequentialGroup()
569. .addGap(20, 20, 20)
570. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
571. .addGroup(layout.createSequentialGroup()
572. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
573. .addComponent(lblNIP)
574. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
575. .addComponent(tfNIP, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)
576. .addComponent(lblAgama))
577. .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
578. .addComponent(lblNoTelp)
579. .addComponent(tfNoTlp, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))
580. .addGap(10, 10, 10)
581. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
582. .addComponent(lblNama)
583. .addComponent(tfNama, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))
584. .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))
585. .addGap(28, 28, 28)
586. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
587. .addComponent(btnAdd)
588. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
589. .addComponent(btnEdit)
590. .addComponent(btnDelete)
591. .addComponent(btnReset)))
592. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
593. .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 190, javax.swing.GroupLayout.PREFERRED\_SIZE)
594. .addContainerGap(64, Short.MAX\_VALUE))
595. );
596. pack();
597. }// </editor-fold>//GEN-END:initComponents
598. private void tblGuruMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_tblGuruMouseClicked
599. DefaultTableModel model = (DefaultTableModel)tblGuru.getModel();
600. int i = tblGuru.getSelectedRow();
601. String a = model.getValueAt(i, 0).toString();
602. String b = model.getValueAt(i, 1).toString();
603. String c = model.getValueAt(i, 2).toString();
604. String d = model.getValueAt(i, 3).toString();
606. tfNIP.setText(a);
607. tfNama.setText(b);
608. tfAgama.setText(c);
609. tfNoTlp.setSelectedItem(d);
611. // tfNPM.setText(model.getValueAt(i, 0).toString());
612. // tfNama.setText(model.getValueAt(i, 1).toString());
613. // cbAngkatan.getModel().setSelectedItem(model.getValueAt(i, 2).toString());
614. }//GEN-LAST:event\_tblGuruMouseClicked
615. private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnAddActionPerformed
616. // TODO add your handling code here:
617. if(checkField() == 1){
618. try {
619. Statement st = con.createStatement();
620. String sql = "INSERT into tbl\_kriteria (id\_kriteria, nama\_kriteria, bobot,flag) values ('"+tfNIP.getText()+"', '"+Utility.capitalizeWord(tfNama.getText())+"', '"+Utility.capitalizeWord(tfAgama.getText())+"', '"+tfNoTlp.getSelectedItem().toString()+"')";
621. st.executeUpdate(sql);
622. RefreshPage();
623. } catch (SQLException e) {
624. JOptionPane.showMessageDialog(null, e);
625. }
627. } else{
628. JOptionPane.showMessageDialog(this, "Isi Semua Field!");
629. }
630. }//GEN-LAST:event\_btnAddActionPerformed
631. private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnEditActionPerformed
632. // TODO add your handling code here:
633. if(tfNIP.equals("")){
634. JOptionPane.showMessageDialog(this, "Pilih Data Yang Ingin Di Edit!");
635. } else{
636. int conf;
637. conf = JOptionPane.showConfirmDialog(null, "Apakah Anda Yakin Ingin Mengedit Data Ini?",
638. "Warning",JOptionPane.WARNING\_MESSAGE);
639. if(conf == 0){
640. try {
641. Statement st = con.createStatement();
642. String sql = "UPDATE `tbl\_kriteria` SET `nama\_kriteria` = '" +Utility.capitalizeWord(tfNama.getText())+"', "+
643. "`bobot` = '"+Utility.capitalizeWord(tfAgama.getText())+"', "+
644. "`flag` = '"+tfNoTlp.getSelectedItem().toString()+"' "+
645. "WHERE `id\_kriteria` ='"+tfNIP.getText()+"'";
646. st.executeUpdate(sql);
647. RefreshPage();
648. } catch (SQLException e) {
649. JOptionPane.showMessageDialog(null, e);
650. }
652. }
653. }
654. }//GEN-LAST:event\_btnEditActionPerformed
655. private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnDeleteActionPerformed
656. // TODO add your handling code here:
657. if(tfNIP.equals(""))
658. {
659. JOptionPane.showMessageDialog(this, "Pilih Data Yang Ingin Di Hapus!");
660. }
661. else{
662. int conf;
663. conf = JOptionPane.showConfirmDialog(null, "Apakah Anda Yakin Ingin Menghapus Data Ini?",
664. "Warning",JOptionPane.WARNING\_MESSAGE);
665. if(conf == 0){
666. try {
667. Statement st = con.createStatement();
668. String sql = "DELETE from tbl\_kriteria where id\_kriteria ='"+tfNIP.getText()+"'";
669. st.executeUpdate(sql);
670. RefreshPage();
671. } catch (SQLException e) {
672. JOptionPane.showMessageDialog(null, e);
673. }
674. }
675. }
676. }//GEN-LAST:event\_btnDeleteActionPerformed
677. public void bersih(){
678. tfNIP.setText("");
679. tfNama.setText("");
680. tfAgama.setText("");
681. tfNoTlp.setSelectedIndex(0);
682. }
684. private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnResetActionPerformed
685. // TODO add your handling code here:
686. tfNIP.setText("");
687. tfNama.setText("");
688. tfAgama.setText("");
689. tfNoTlp.setSelectedIndex(0);
690. }//GEN-LAST:event\_btnResetActionPerformed
692. public int checkField(){
693. if(
694. tfNIP.getText().trim().isEmpty() ||
695. tfNama.getText().trim().isEmpty() ||
696. tfAgama.getText().trim().isEmpty()
697. ) {
698. return 0;
699. } else{
700. return 1;
701. }
702. }
703. // Variables declaration - do not modify//GEN-BEGIN:variables
704. private javax.swing.JButton btnAdd;
705. private javax.swing.JButton btnDelete;
706. private javax.swing.JButton btnEdit;
707. private javax.swing.JButton btnReset;
708. private javax.swing.JScrollPane jScrollPane1;
709. private javax.swing.JScrollPane jScrollPane2;
710. private javax.swing.JLabel lblAgama;
711. private javax.swing.JLabel lblNIP;
712. private javax.swing.JLabel lblNama;
713. private javax.swing.JLabel lblNoTelp;
714. private javax.swing.JTable tblGuru;
715. private javax.swing.JTextArea tfAgama;
716. private javax.swing.JTextField tfNIP;
717. private javax.swing.JTextField tfNama;
718. private javax.swing.JComboBox<String> tfNoTlp;
719. // End of variables declaration//GEN-END:variables
720. }
721. Data Penilaian
722. /\*
723. \* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
724. \* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JInternalFrame.java to edit this template
725. \*/
726. package Form;
727. import Koneksi.KoneksiDb;
728. import Process.PrintPdf;
729. import Process.ReadWrite;
730. import Process.ResultData;
731. import Process.ResultOrder;
732. import com.itextpdf.text.DocumentException;
733. import java.awt.Dimension;
734. import java.awt.event.KeyEvent;
735. import java.io.FileNotFoundException;
736. import java.sql.Connection;
737. import java.sql.ResultSet;
738. import java.sql.Statement;
739. import java.text.DateFormat;
740. import java.text.SimpleDateFormat;
741. import java.sql.Date;
742. import java.sql.PreparedStatement;
743. import java.sql.SQLException;
744. import java.util.LinkedList;
745. import javax.swing.JOptionPane;
746. import javax.swing.plaf.basic.BasicInternalFrameUI;
747. import javax.swing.table.DefaultTableModel;
748. /\*\*
749. \*
750. \* @author resar
751. \*/
752. public class Penilaian extends javax.swing.JInternalFrame {
753. /\*\*
754. \* Creates new form Profile
755. \*/
756. private PrintPdf printPdf = new PrintPdf();
757. private Penilaian tesData;
758. private LinkedList<Double> listResult = new LinkedList<>();
759. private LinkedList<Double> listResult1 = new LinkedList<>();
760. private LinkedList<ResultData> listResultData = new LinkedList<>();
761. private ResultData resultData;
762. private LinkedList<ResultData> listResultData1 = new LinkedList<>();
763. private ResultData resultData1;
764. private ResultOrder resultOrder = new ResultOrder();
765. private ResultOrder resultOrder1 = new ResultOrder();
767. public double[][] data;
768. private double[][] dataBagi;
769. private double[][] dataPersen;
771. private int alternatif = 0;
772. private int criteria = 0;
774. Connection con;
775. Statement st;
776. ResultSet rs;
777. PreparedStatement ps;
778. public LinkedList<Object> listKriteriaId = new LinkedList<>();
779. public LinkedList<Object> listKriteriaNama = new LinkedList<>();
780. public LinkedList<String> listHeader = new LinkedList<>();
781. public LinkedList<Object> listKriteriaBobot = new LinkedList<>();
782. public LinkedList<Object> listKriteriaFlag = new LinkedList<>();
783. public LinkedList<Object> listnasabahId = new LinkedList<>();
784. public LinkedList<Object> listnasabahNama = new LinkedList<>();
785. private ReadWrite readWrite = new ReadWrite();
786. public Penilaian() throws SQLException {
787. initComponents();
788. con = new KoneksiDb().getConnect();
789. tampilData();
790. }
791. /\*\*
792. \* This method is called from within the constructor to initialize the form.
793. \* WARNING: Do NOT modify this code. The content of this method is always
794. \* regenerated by the Form Editor.
795. \*/
796. @SuppressWarnings("unchecked")
797. // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
798. private void initComponents() {
799. jDialog1 = new javax.swing.JDialog();
800. jScrollPane2 = new javax.swing.JScrollPane();
801. jTable1 = new javax.swing.JTable();
802. jLabel1 = new javax.swing.JLabel();
803. jScrollPane3 = new javax.swing.JScrollPane();
804. jTable2 = new javax.swing.JTable();
805. jLabel2 = new javax.swing.JLabel();
806. jButton1 = new javax.swing.JButton();
807. jButton2 = new javax.swing.JButton();
808. jScrollPane1 = new javax.swing.JScrollPane();
809. tblProsesHitung = new javax.swing.JTable();
810. btnAdd = new javax.swing.JButton();
811. btnEdit = new javax.swing.JButton();
812. btnReset = new javax.swing.JButton();
813. jTable1.setModel(new javax.swing.table.DefaultTableModel(
814. new Object [][] {
815. {null, null, null, null},
816. {null, null, null, null},
817. {null, null, null, null},
818. {null, null, null, null}
819. },
820. new String [] {
821. "Title 1", "Title 2", "Title 3", "Title 4"
822. }
823. ));
824. jScrollPane2.setViewportView(jTable1);
825. jLabel1.setText("HASIL PERHITUNGAN SAW");
826. jTable2.setModel(new javax.swing.table.DefaultTableModel(
827. new Object [][] {
828. {null, null, null, null},
829. {null, null, null, null},
830. {null, null, null, null},
831. {null, null, null, null}
832. },
833. new String [] {
834. "Title 1", "Title 2", "Title 3", "Title 4"
835. }
836. ));
837. jScrollPane3.setViewportView(jTable2);
838. jLabel2.setText("NORMALISASI");
839. jButton1.setText("Hitung");
840. jButton1.addActionListener(new java.awt.event.ActionListener() {
841. public void actionPerformed(java.awt.event.ActionEvent evt) {
842. jButton1ActionPerformed(evt);
843. }
844. });
845. jButton2.setText("Clear");
846. jButton2.addActionListener(new java.awt.event.ActionListener() {
847. public void actionPerformed(java.awt.event.ActionEvent evt) {
848. jButton2ActionPerformed(evt);
849. }
850. });
851. javax.swing.GroupLayout jDialog1Layout = new javax.swing.GroupLayout(jDialog1.getContentPane());
852. jDialog1.getContentPane().setLayout(jDialog1Layout);
853. jDialog1Layout.setHorizontalGroup(
854. jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
855. .addGroup(jDialog1Layout.createSequentialGroup()
856. .addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
857. .addGroup(jDialog1Layout.createSequentialGroup()
858. .addContainerGap()
859. .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT\_SIZE, 770, Short.MAX\_VALUE))
860. .addGroup(jDialog1Layout.createSequentialGroup()
861. .addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
862. .addGroup(jDialog1Layout.createSequentialGroup()
863. .addGap(344, 344, 344)
864. .addComponent(jLabel1))
865. .addGroup(jDialog1Layout.createSequentialGroup()
866. .addGap(368, 368, 368)
867. .addComponent(jLabel2))
868. .addGroup(jDialog1Layout.createSequentialGroup()
869. .addContainerGap()
870. .addComponent(jButton1)
871. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
872. .addComponent(jButton2)))
873. .addGap(0, 0, Short.MAX\_VALUE)))
874. .addContainerGap())
875. .addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
876. .addGroup(jDialog1Layout.createSequentialGroup()
877. .addContainerGap()
878. .addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT\_SIZE, 770, Short.MAX\_VALUE)
879. .addContainerGap()))
880. );
881. jDialog1Layout.setVerticalGroup(
882. jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
883. .addGroup(jDialog1Layout.createSequentialGroup()
884. .addGap(10, 10, 10)
885. .addComponent(jLabel1)
886. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 360, Short.MAX\_VALUE)
887. .addComponent(jLabel2)
888. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
889. .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 328, javax.swing.GroupLayout.PREFERRED\_SIZE)
890. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
891. .addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
892. .addComponent(jButton1)
893. .addComponent(jButton2))
894. .addGap(8, 8, 8))
895. .addGroup(jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
896. .addGroup(jDialog1Layout.createSequentialGroup()
897. .addGap(49, 49, 49)
898. .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 328, javax.swing.GroupLayout.PREFERRED\_SIZE)
899. .addContainerGap(423, Short.MAX\_VALUE)))
900. );
901. setFocusCycleRoot(false);
902. setFocusable(false);
903. setPreferredSize(new java.awt.Dimension(1200, 580));
904. setRequestFocusEnabled(false);
905. jScrollPane1.setAutoscrolls(true);
906. tblProsesHitung.setAutoCreateRowSorter(true);
907. tblProsesHitung.setBorder(javax.swing.BorderFactory.createCompoundBorder());
908. tblProsesHitung.setModel(new javax.swing.table.DefaultTableModel(
909. new Object [][] {
910. {}
911. },
912. new String [] {
913. }
914. ));
915. tblProsesHitung.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
916. jScrollPane1.setViewportView(tblProsesHitung);
917. btnAdd.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
918. btnAdd.setText("ADD");
919. btnAdd.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
920. btnAdd.addActionListener(new java.awt.event.ActionListener() {
921. public void actionPerformed(java.awt.event.ActionEvent evt) {
922. btnAddActionPerformed(evt);
923. }
924. });
925. btnEdit.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
926. btnEdit.setText("EDIT");
927. btnEdit.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
928. btnEdit.addActionListener(new java.awt.event.ActionListener() {
929. public void actionPerformed(java.awt.event.ActionEvent evt) {
930. btnEditActionPerformed(evt);
931. }
932. });
933. btnReset.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
934. btnReset.setText("RESET");
935. btnReset.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));
936. btnReset.addActionListener(new java.awt.event.ActionListener() {
937. public void actionPerformed(java.awt.event.ActionEvent evt) {
938. btnResetActionPerformed(evt);
939. }
940. });
941. javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
942. getContentPane().setLayout(layout);
943. layout.setHorizontalGroup(
944. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
945. .addGroup(layout.createSequentialGroup()
946. .addGap(30, 30, 30)
947. .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 578, javax.swing.GroupLayout.PREFERRED\_SIZE)
948. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
949. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
950. .addComponent(btnAdd, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE)
951. .addComponent(btnEdit, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE)
952. .addComponent(btnReset, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE))
953. .addGap(0, 0, Short.MAX\_VALUE))
954. );
955. layout.setVerticalGroup(
956. layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
957. .addGroup(layout.createSequentialGroup()
958. .addContainerGap()
959. .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
960. .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 359, javax.swing.GroupLayout.PREFERRED\_SIZE)
961. .addGroup(layout.createSequentialGroup()
962. .addComponent(btnAdd)
963. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
964. .addComponent(btnEdit)
965. .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
966. .addComponent(btnReset)))
967. .addContainerGap(180, Short.MAX\_VALUE))
968. );
969. pack();
970. }// </editor-fold>//GEN-END:initComponents
971. private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnAddActionPerformed
972. // TODO add your handling code here:
973. boolean check = true;
974. for (int i = 0; i < tblProsesHitung.getRowCount()-2; i++) {
975. for (int j = 2; j < tblProsesHitung.getColumnCount(); j++) {
976. String value = (String) tblProsesHitung.getValueAt(i, j);
977. //System.out.print("Value("+i+","+j+") : "+isNumeric(value)+" || ");
978. if (value != null) {
979. if(isNumeric(value) == false){
980. check = false;
981. }
982. }else if(value == null){
983. check = false;
984. }
985. }
986. System.out.println();
987. }
988. if (check) {
989. tblProsesHitung.editCellAt(-1, -1);
990. tblProsesHitung.setRowSelectionInterval(0, 0);
991. tblProsesHitung.setEnabled(false);
992. JOptionPane.showMessageDialog(null, "Data sudah disimpan");
993. jDialog1.setVisible(true);
994. jDialog1.setSize(new Dimension(800, 800));
995. jDialog1.setResizable(false);
996. jDialog1.setLocationRelativeTo(null);
998. } else {
999. JOptionPane.showMessageDialog(null, "Mohon cek kembali data yang anda inputkan");
1000. }
1001. }//GEN-LAST:event\_btnAddActionPerformed
1002. private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnEditActionPerformed
1003. // TODO add your handling code here:
1004. tblProsesHitung.setEnabled(true);
1005. }//GEN-LAST:event\_btnEditActionPerformed
1006. private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnResetActionPerformed
1007. // TODO add your handling code here:
1008. try {
1009. tblProsesHitung.setEnabled(true);
1010. refresh();
1011. tampilData();
1012. } catch (SQLException ex) {
1013. }
1014. }//GEN-LAST:event\_btnResetActionPerformed
1015. private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed
1017. // TODO add your handling code here:
1018. DefaultTableModel mdlClear = (DefaultTableModel) jTable1.getModel();
1019. mdlClear.setRowCount(0);
1020. listResult.clear();
1021. listResultData.clear();
1023. DefaultTableModel mdlClear1 = (DefaultTableModel) jTable2.getModel();
1024. mdlClear1.setRowCount(0);
1025. listResult1.clear();
1026. listResultData1.clear();
1027. }//GEN-LAST:event\_jButton2ActionPerformed
1028. private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed
1029. // TODO add your handling code here:
1030. DefaultTableModel mdlClear = (DefaultTableModel) jTable2.getModel();
1031. mdlClear.setRowCount(0);
1032. listResult.clear();
1034. DefaultTableModel mdlClear1 = (DefaultTableModel) jTable1.getModel();
1035. mdlClear1.setRowCount(0);
1036. listResult1.clear();
1038. //proses menghapus semua isi di folder laporan
1039. printPdf.removeDir("laporan");


1043. alternatif = tblProsesHitung.getRowCount()-2;
1044. criteria = tblProsesHitung.getColumnCount()-2;
1045. data = new double[alternatif][criteria];
1046. dataBagi = new double[alternatif][criteria];
1047. dataPersen = new double[alternatif][criteria];
1048. //Proses input data dari table ke dalam array 2D
1049. for (int i = 0; i < alternatif; i++) {
1050. for (int j = 0; j < criteria; j++) {
1051. data[i][j] = Double.parseDouble(tblProsesHitung.getValueAt(i, j+2).toString());
1052. }
1053. }
1054. System.out.println("\n============== DATA ASLI ================");
1055. for (int i = 0; i < alternatif; i++) {
1056. for (int j = 0; j < criteria; j++) {
1057. System.out.print(data[i][j] + "\t");
1058. }
1059. System.out.println("");
1060. }
1062. //PROSES PRINT DATA
1063. try {
1064. printPdf.print("DATA NILAI KRITERIA", "data nilai kriteria", listKriteriaId, listnasabahId, listnasabahNama, data);
1065. } catch (FileNotFoundException ex) {
1066. } catch (DocumentException ex) {
1067. }
1068. //Proses Membagi Nilai
1069. for (int i = 0; i < criteria; i++) {
1070. String value = tblProsesHitung.getValueAt(tblProsesHitung.getRowCount() - 1, i + 2).toString();
1071. if (value.equals("benefit")) {
1072. double max = 0;
1073. for (int j = 0; j < alternatif; j++) {
1074. double nilai = data[j][i];
1075. if (nilai > max) {
1076. max = nilai;
1077. }
1078. }
1079. for (int j = 0; j < alternatif; j++) {
1080. double val = data[j][i] / max;
1081. dataBagi[j][i] = Math.round(val\*1000.0)/1000.0;
1082. }
1083. } else {
1084. double min = 1000000000;
1085. for (int j = 0; j < alternatif; j++) {
1086. double nilai = data[j][i];
1087. if (nilai < min) {
1088. min = nilai;
1089. }
1090. }
1091. for (int j = 0; j < alternatif; j++) {
1092. double val = min / data[j][i];
1093. dataBagi[j][i] = Math.round(val\*1000.0)/1000.0;
1094. }
1095. }
1096. }
1097. System.out.println("\n\n=================DATA SETELAH DIBAGI====================");
1098. for (int i = 0; i < alternatif; i++) {
1099. for (int j = 0; j < criteria; j++) {
1100. System.out.print(dataBagi[i][j] + "\t");
1101. }
1102. System.out.println();
1103. }
1104. //PROSES PRINT DATA
1105. try {
1106. printPdf.print("DATA NILAI TERNORMALISASI", "data nilai ternormalisasi", listKriteriaId, listnasabahId, listnasabahNama, dataBagi);
1107. } catch (FileNotFoundException ex) {
1108. } catch (DocumentException ex) {
1109. }
1110. //Proses Menghitung Dengan Persentase
1111. for (int i = 0; i < criteria; i++) {
1112. double persentase = Float.parseFloat(tblProsesHitung.getValueAt(tblProsesHitung.getRowCount() - 2, i + 2).toString());
1113. System.out.print(persentase + "%\t");
1114. for (int j = 0; j < alternatif; j++) {
1115. double val = (dataBagi[j][i] \* persentase) / 100f;
1116. dataPersen[j][i] = Math.round(val\*1000.0)/1000.0;
1117. }
1118. }
1119. System.out.println("\n\n=================DATA PERSENTASE====================");
1121. DefaultTableModel mdl1 = new DefaultTableModel();
1122. mdl1.addColumn("C1");
1123. mdl1.addColumn("C2");
1124. mdl1.addColumn("C3");
1125. mdl1.addColumn("C4");
1126. for (int i = 0; i < alternatif; i++) {
1127. for (int j = 0; j < criteria; j++) {
1128. System.out.print(dataPersen[i][j] + "\t");
1129. mdl1.addRow(new Object[]{
1130. dataPersen[i][0],
1131. dataPersen[i][1],
1132. dataPersen[i][2],
1133. dataPersen[i][3]
1134. });
1135. }
1136. System.out.println();
1137. }
1138. jTable1.setModel(mdl1);
1139. //Proses menjumlahkan setiap kriteria
1140. for (int i = 0; i < alternatif; i++) {
1141. double sum = 0;
1142. for (int j = 0; j < criteria; j++) {
1143. sum += dataPersen[i][j];
1144. }
1145. listResult.add(Math.round(sum\*1000.0)/1000.0);
1146. }
1148. //PROSES PRINT DATA
1149. // try {
1150. // printPdf.print("DATA HASIL PERBANDINGAN", "data hasil perbandingan", listKriteriaId, listnasabahId, listnasabahNama, dataPersen);
1151. // } catch (FileNotFoundException ex) {
1152. // Logger.getLogger(M\_ProsesHitung.class.getName()).log(Level.SEVERE, null, ex);
1153. // } catch (DocumentException ex) {
1154. // Logger.getLogger(M\_ProsesHitung.class.getName()).log(Level.SEVERE, null, ex);
1155. // }
1156. for(int i=0; i<listResult.size(); i++){
1157. LinkedList<Double> listValue = new LinkedList<>();
1158. int cols = tblProsesHitung.getColumnCount();
1159. for(int j=2; j<cols; j++){
1160. double val = Float.parseFloat(tblProsesHitung.getValueAt(i, j).toString());
1161. listValue.add(val);
1162. }
1163. resultData = new ResultData();
1164. resultData.setAlternatif("Alternatif "+(i+1));
1165. resultData.setNamaAlternatif(tblProsesHitung.getValueAt(i, 1).toString());
1166. resultData.setListValue(listValue);
1167. resultData.setResult(listResult.get(i));
1169. listResultData.add(resultData);
1170. }
1172. ResultData[] rd = new ResultData[listResultData.size()];
1173. for(int i=0; i<rd.length; i++){
1174. rd[i] = listResultData.get(i);
1175. }
1177. LinkedList<ResultData> listFinalResult = resultOrder.getResultData(rd);
1178. String sqln = "TRUNCATE table tbl\_hasil";
1179. try {
1180. ps = con.prepareStatement(sqln);
1181. ps.execute();
1182. } catch (SQLException ex) {
1183. }
1184. String sql = "insert into tbl\_hasil(alternatif, nama, c1, c2,c3,c4,hasil) values(?,?,?,?,?,?,?)";
1186. try {
1187. ps = con.prepareStatement(sql);
1188. } catch (SQLException ex) {
1189. }

1192. for(int i=0; i<listFinalResult.size(); i++){
1193. try {
1194. ps.setString(1, listFinalResult.get(i).getAlternatif());
1195. ps.setString(2, listFinalResult.get(i).getNamaAlternatif());
1196. System.out.print(listFinalResult.get(i).getAlternatif()+"||");
1197. System.out.print(listFinalResult.get(i).getNamaAlternatif()+"||");
1198. int c = 3;
1199. for(int j=0; j<listFinalResult.get(i).getListValue().size(); j++){
1200. System.out.print(listFinalResult.get(i).getListValue().get(j)+"||");
1201. ps.setDouble(c, listFinalResult.get(i).getListValue().get(j));
1202. c++;
1203. }
1204. System.out.println(listFinalResult.get(i).getResult());
1205. ps.setDouble(c, listFinalResult.get(i).getResult());
1206. ps.execute();
1207. } catch (SQLException ex) {
1208. }



1213. }
1215. //Proses menampilkan di tabel
1216. DefaultTableModel mdl = new DefaultTableModel();
1217. mdl.addColumn("Alternatif");
1218. mdl.addColumn("Nama");
1219. mdl.addColumn("Hasil");
1220. for (int i = 0; i < listResult.size(); i++) {
1221. mdl.addRow(new Object[]{
1222. listFinalResult.get(i).getAlternatif(),
1223. listFinalResult.get(i).getNamaAlternatif(),
1224. listFinalResult.get(i).getResult()
1225. });
1226. }
1227. jTable2.setModel(mdl);
1229. //PROSES PRINT DATA
1230. // try {
1231. // printPdf.print("DATA NILAI HASIL PERANKINGAN", "data nilai hasil perankingan", listFinalResult);
1232. // } catch (FileNotFoundException ex) {
1233. // Logger.getLogger(M\_ProsesHitung.class.getName()).log(Level.SEVERE, null, ex);
1234. // } catch (DocumentException ex) {
1235. // Logger.getLogger(M\_ProsesHitung.class.getName()).log(Level.SEVERE, null, ex);
1236. // }
1237. //PROSES PRINT DATA
1238. try {
1239. printPdf.print("DATA NILAI HASIL PERANKINGAN", "data nilai hasil perankingan", listKriteriaId, listFinalResult);
1240. } catch (FileNotFoundException ex) {
1241. } catch (DocumentException ex) {
1242. }
1244. }//GEN-LAST:event\_jButton1ActionPerformed
1246. void tampilData() throws SQLException {
1247. DefaultTableModel mdl = new DefaultTableModel();
1248. tblProsesHitung.setModel(mdl);
1249. //proses menentukan header
1250. mdl.addColumn("<html><b>Alternatif</b><html>");
1251. mdl.addColumn("<html><b>Nama</b><html>");
1252. try {
1253. st = con.createStatement();
1254. rs = st.executeQuery("select \* from tbl\_kriteria");
1255. while (rs.next()) {
1256. listKriteriaId.add(rs.getString("id\_kriteria"));
1257. listKriteriaNama.add(rs.getString("nama\_kriteria"));
1258. listKriteriaBobot.add(rs.getString("bobot"));
1259. listKriteriaFlag.add(rs.getString("flag"));
1260. }
1261. } catch (Exception e) {
1262. System.out.println("#ERROR " + e.getMessage());
1263. }
1264. for (int i = 0; i < listKriteriaId.size(); i++) {
1265. listHeader.add("<html><b>" + (String) listKriteriaId.get(i) + " (" + (String) listKriteriaNama.get(i) + ")</b><html>");
1266. }
1267. for (String i : listHeader) {
1268. mdl.addColumn(i);
1269. }
1270. //Proses Menentukan Isi
1271. try {
1272. st = con.createStatement();
1273. rs = st.executeQuery("select \* from tbl\_alternatif ORDER BY LENGTH(id\_alternatif), id\_alternatif");
1274. while (rs.next()) {
1275. listnasabahId.add(rs.getString("id\_alternatif"));
1276. listnasabahNama.add(rs.getString("nama"));
1277. mdl.addRow(new Object[]{
1278. rs.getString("id\_alternatif"),
1279. rs.getString("nama")
1280. });
1281. tblProsesHitung.setModel(mdl);
1282. }
1283. } catch (Exception e) {
1284. System.out.println("#ERROR " + e.getMessage());
1285. }
1286. //Proses Menentukan Bobot dan Flag
1287. mdl.addRow(new Object[]{
1288. "<html><b>Bobot (%)</b></html>",
1289. " - "
1290. });
1291. mdl.addRow(new Object[]{
1292. "<html><b>Flag</b><html>",
1293. " - "
1294. });
1295. //mdl.isCellEditable(0, 0);
1296. //proses agar tabel ada scroll
1297. tblProsesHitung.setModel(mdl);
1298. for (int i = 0; i < mdl.getColumnCount(); i++) {
1299. tblProsesHitung.getColumnModel().getColumn(i).setPreferredWidth(150);
1300. }
1301. int rows = tblProsesHitung.getRowCount() - 1;
1302. int cols = tblProsesHitung.getColumnCount();
1303. for (int i = 2; i < cols; i++) {
1304. tblProsesHitung.setValueAt(listKriteriaBobot.get(i - 2), rows - 1, i);
1305. tblProsesHitung.setValueAt(listKriteriaFlag.get(i - 2), rows, i);
1306. }
1307. tblProsesHitung.setShowGrid(true);
1308. }
1309. public void refresh() {
1310. listHeader.clear();
1311. listKriteriaBobot.clear();
1312. listKriteriaFlag.clear();
1313. listKriteriaId.clear();
1314. listKriteriaNama.clear();
1315. listnasabahId.clear();
1316. listnasabahNama.clear();
1317. }
1318. public boolean isNumeric(String str) {
1319. try {
1320. double d = Double.parseDouble(str);
1321. } catch (NumberFormatException nfe) {
1322. return false;
1323. }
1324. return true;
1325. }
1326. // Variables declaration - do not modify//GEN-BEGIN:variables
1327. private javax.swing.JButton btnAdd;
1328. private javax.swing.JButton btnEdit;
1329. private javax.swing.JButton btnReset;
1330. private javax.swing.JButton jButton1;
1331. private javax.swing.JButton jButton2;
1332. private javax.swing.JDialog jDialog1;
1333. private javax.swing.JLabel jLabel1;
1334. private javax.swing.JLabel jLabel2;
1335. private javax.swing.JScrollPane jScrollPane1;
1336. private javax.swing.JScrollPane jScrollPane2;
1337. private javax.swing.JScrollPane jScrollPane3;
1338. private javax.swing.JTable jTable1;
1339. private javax.swing.JTable jTable2;
1340. public javax.swing.JTable tblProsesHitung;
1341. // End of variables declaration//GEN-END:variables
1342. }