

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Rekayasa Perangkat Lunak 2

Kelas : 4IA26

Praktikum ke- : 4

Tanggal : 15 November 2024

Materi : ORM

NPM : 50421002

Nama : Abdan Syakur

Ketua Asisten :

Nama Asisten :

Paraf Asisten :

Jumlah Lembar : 7



LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2023/2024

LISTING

```
1  import java.sql.Connection;
2  import java.sql.DriverManager;
3  import java.sql.ResultSet;
4  import java.sql.Statement;
5  import javax.swing.JOptionPane;
6  /*
7   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
8   * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
9   */
10
11  /**
12   *
13   * @author abdan
14   */
15  public class latihan extends javax.swing.JFrame {
16      private Statement St;
17      private Connection Con;
18      private ResultSet Rs;
19      private String sql = "";
20      /**
21       * Creates new form latihan
22       */
23      public latihan() {
24          initComponents();
25          KoneksiDatabase();
26      }
27
28      private void KoneksiDatabase() {
29          try {
30              // Memuat driver JDBC untuk MySQL
31              Class.forName("com.mysql.cj.jdbc.Driver");
32
33              // Membuat koneksi ke database
34              Con = DriverManager.getConnection("jdbc:mysql://localhost:3306/db_latihan", "root", "abdano14");
35              JOptionPane.showMessageDialog(null, "Koneksi Berhasil");
36
37              // Inisialisasi Statement
38              St = Con.createStatement();
39
40          } catch (Exception e) {
41              e.printStackTrace();
42              System.out.println("Koneksi Gagal: " + e.getMessage());
43          }
44          try {
45              Class.forName("com.mysql.cj.jdbc.Driver");
46
47          } catch (ClassNotFoundException e) {
48              System.out.println("JDBC Driver tidak ditemukan!");
49              e.printStackTrace();
50          }
51
52      }
53  }
```

```

55  /**
56  * This method is called from within the constructor to initialize the form.
57  * WARNING: Do NOT modify this code. The content of this method is always
58  * regenerated by the Form Editor.
59  */
60  @SuppressWarnings("unchecked")
61  Generated Code
79
80  /**
81  * @param args the command line arguments
82  */
83  public static void main(String args[]) {
84      /* Set the Nimbus look and feel */
85      Look and feel setting code (optional)
106
107      /* Create and display the form */
108      java.awt.EventQueue.invokeLater(new Runnable() {
109          public void run() {
110              new latihan().setVisible(true);
111          }
112      });
113  }
114
115  // Variables declaration - do not modify
116  // End of variables declaration
117  }
118

```

LOGIKA

```
1  import java.sql.Connection;
2  import java.sql.DriverManager;
3  import java.sql.ResultSet;
4  import java.sql.Statement;
5  import javax.swing.JOptionPane;
6  /*
7   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
8   * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
9   */
10
11  /**
12   *
13   * @author abdan
14   */
15  public class latihan extends javax.swing.JFrame {
16      private Statement St;
17      private Connection Con;
18      private ResultSet Rs;
19      private String sql = "";
20      /**
21       * Creates new form latihan
22       */
23      public latihan() {
24          initComponents();
25          KoneksiDatabase();
26      }
```

Kelas Java `latihan` adalah sebuah aplikasi GUI yang terhubung ke database menggunakan Java Swing. Kelas ini memiliki beberapa field, yaitu `Con` untuk mengelola koneksi ke database, `St` untuk menangani pernyataan SQL, dan `Rs` untuk menyimpan hasil query. Di dalam konstruktor `latihan()`, komponen GUI diinisialisasi melalui metode `initComponents()` yang secara otomatis dihasilkan oleh NetBeans. Setelah itu, metode `KoneksiDatabase()` dipanggil untuk menghubungkan aplikasi dengan database. Dengan struktur ini, aplikasi dapat menjalankan operasi pada database, menampilkan data, dan berinteraksi dengan pengguna melalui antarmuka grafis yang sederhana.

```

28 private void KoneksiDatabase() {
29     try {
30         // Memuat driver JDBC untuk MySQL
31         Class.forName("com.mysql.cj.jdbc.Driver");
32
33         // Membuat koneksi ke database
34         Con = DriverManager.getConnection("jdbc:mysql://localhost:3306/db_latihan", "root", "abdan014");
35         JOptionPane.showMessageDialog(null, "Koneksi Berhasil");
36
37         // Inisialisasi Statement
38         St = Con.createStatement();
39
40     } catch (Exception e) {
41         e.printStackTrace();
42         System.out.println("Koneksi Gagal: " + e.getMessage());
43     }
44     try {
45         Class.forName("com.mysql.cj.jdbc.Driver");
46
47     } catch (ClassNotFoundException e) {
48         System.out.println("JDBC Driver tidak ditemukan!");
49         e.printStackTrace();
50     }
51 }
52
53 }

```

KoneksiDatabase() berfungsi untuk membuat koneksi ke database MySQL dengan terlebih dahulu memuat driver JDBC MySQL menggunakan `Class.forName("com.mysql.cj.jdbc.Driver")`. Setelah driver siap, koneksi ke database `db_latihan` di `localhost` dibuat menggunakan `DriverManager.getConnection(...)` dengan kredensial username "root" dan password "abdan014". Jika koneksi berhasil, akan muncul pesan "Koneksi Berhasil", dan objek Statement (St) diinisialisasi untuk menjalankan perintah SQL pada database. Jika terjadi kegagalan koneksi atau driver tidak ditemukan, pesan error dicetak di konsol dan detail kesalahan akan ditampilkan.

```

55  /**
56  * This method is called from within the constructor to initialize the form.
57  * WARNING: Do NOT modify this code. The content of this method is always
58  * regenerated by the Form Editor.
59  */
60  @SuppressWarnings("unchecked")
61  Generated Code
79
80  /**
81  * @param args the command line arguments
82  */
83  public static void main(String args[]) {
84      /* Set the Nimbus look and feel */
85      Look and feel setting code (optional)
106
107      /* Create and display the form */
108      java.awt.EventQueue.invokeLater(new Runnable() {
109          public void run() {
110              new latihan().setVisible(true);
111          }
112      });
113  }
114
115  // Variables declaration - do not modify
116  // End of variables declaration
117  }
118

```

Metode `initComponents()` pada kode di atas berfungsi untuk mengatur komponen dasar dalam antarmuka GUI aplikasi. Metode ini dipanggil di dalam konstruktor dan dihasilkan secara otomatis oleh editor GUI, sehingga tidak disarankan untuk dimodifikasi secara manual. Metode ini mengatur `setDefaultCloseOperation` agar aplikasi tertutup ketika jendela ditutup, lalu menggunakan `GroupLayout` untuk mengatur tata letak (layout) antarmuka dengan dimensi horizontal 400 piksel dan vertikal 300 piksel. Setelah layout selesai diatur, metode `pack()` dipanggil untuk menyesuaikan ukuran jendela sesuai dengan tata letak komponen. Di metode `main`, tema Nimbus diaktifkan jika tersedia, dan blok kode `EventQueue.invokeLater` memastikan bahwa GUI (`latihan`) diluncurkan di event-dispatch thread, menjadikannya responsif dan aman untuk diakses dari antarmuka grafis.

OUTPUT

