

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJECT**

PERIODE: SEMESTER GENAP 2024/2025



NAMA : NESSA NOVIANTI

NIM : 24104410007

PERIODE : SEMESTER GENAP 2024/2025

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM BALITAR
2024/2025**

1. Pembuatan Database dengan nama db_toko dan table product

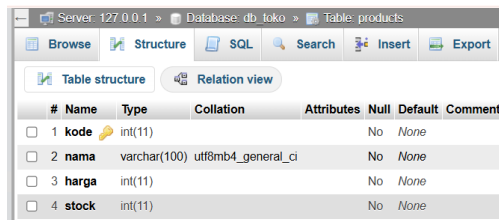
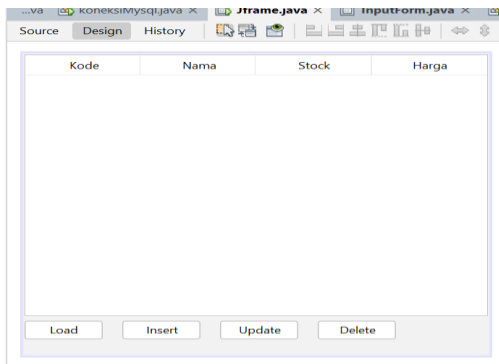


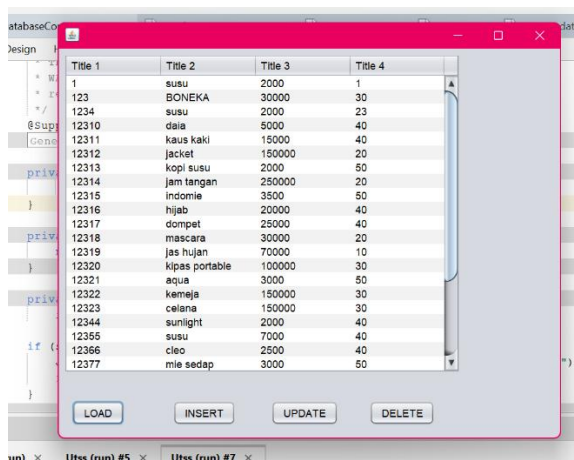
Table structure for 'products' table:

#	Name	Type	Collation	Attributes	Null	Default	Comment
1	kode	int(11)			No	None	
2	nama	varchar(100)	utf8mb4_general_ci		No	None	
3	harga	int(11)			No	None	
4	stock	int(11)			No	None	

2. Pembuatan JFrame yang dilengkapi dengan 4 JButton dan 1 JTable



- Load: seleksi semua data di tabel produk

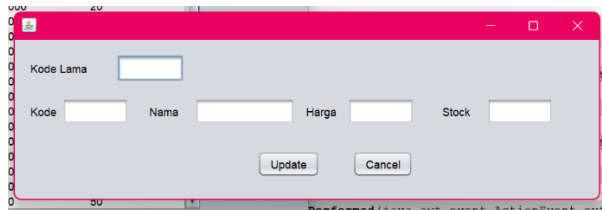


- Insert: memasukkan satu produk baru

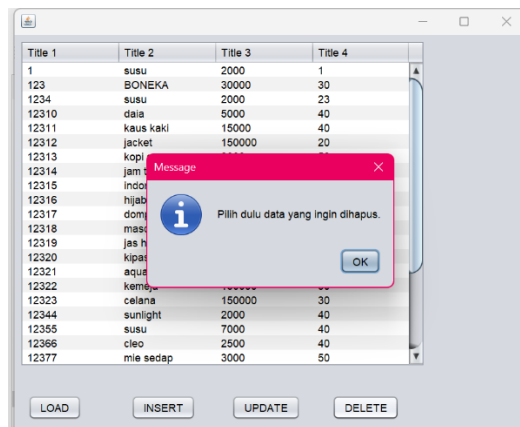


Form fields: kode, Nama, Harga, Stock. Buttons: Tambah, Cancel.

- Update: menyeleksi salah satu isi tabel dan melakukan update



- Delete: menyeleksi salah satu isi tabel dan melakukan delete



3. Penerapan enkapsulasi, Inheritance, try-catch SQLException, GUI Swing project

A. Enkapsulasi

```
private void LoadActionPerformed(java.awt.event.ActionEvent evt) {
try {
    Connection conn = DatabaseConnection.DatabaseConnection();
    Statement stat = conn.createStatement();
    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
    model.setRowCount(0);
    while (rs.next()) {
        Vector<Object> row = new Vector<>();
        row.add(rs.getInt("kode"));
        row.add(rs.getString("nama"));
        row.add(rs.getInt("harga"));
        row.add(rs.getInt("stock"));
        model.addRow(row);
    }
} catch (SQLException ex) {
    JOptionPane.showMessageDialog(this, "Error saat memuat data: " + ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    ex.printStackTrace();
}
}
```

a. Inheritance

```
public class Kasir extends javax.swing.JFrame {

    /**
     * Creates new form Kasir
     */
    public Kasir() {
        initComponents();
    }
}
```

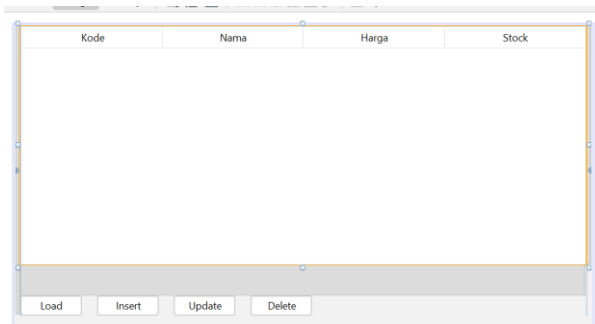
b. try-catch SQLException

```

7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

c. GUI Swing



4. Penjelasan codingan dan output

A. DATA BASE CONNECTION

```
package projectpbo;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.SQLException;
```

```
/**
```

```
*
```

```
* @author ASUS
```

```
*/
```

```
public class DatabaseConnection {
```

```
    private static final String URL = "jdbc:mysql://localhost:3306/db_toko";
```

```
    private static final String USER = "root"; // Ganti dengan username MySQL Anda
```

```
    private static final String PASSWORD = ""; // Ganti dengan password MySQL Anda
```

```
    private static Connection connection = null;
```

```

// Metode untuk mendapatkan koneksi

public static Connection DatabaseConnection() throws SQLException {
    if (connection == null || connection.isClosed()) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(URL, USER, PASSWORD);
            System.out.println("Koneksi ke database berhasil!");
        } catch (ClassNotFoundException e) {
            System.err.println("JDBC Driver tidak ditemukan.");
            throw new SQLException("Driver tidak ditemukan", e);
        }
    }
    return connection;
}

// Metode untuk menutup koneksi

public static void closeConnection() {
    if (connection != null) {
        try {
            connection.close();
            System.out.println("Koneksi database ditutup.");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

```

static Connection getConnection() throws SQLException, ClassNotFoundException {
    Class.forName("com.mysql.cj.jdbc.Driver");
    return DriverManager.getConnection("jdbc:mysql://localhost:3306/db_toko", "root", "");
}

```

- ANALISA

Kelas DatabaseConnection adalah kelas yang digunakan untuk mengelola koneksi ke database MySQL db_toko dengan dua metode: DatabaseConnection() dan getConnection(). Kode ini efektif untuk aplikasi sederhana, menggunakan driver com.mysql.cj.jdbc.Driver dan kredensial root.

B. KASIR

```

package projectpbo;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
import java.sql.Statement;
import java.sql.ResultSet;

public class Kasir extends javax.swing.JFrame {

    /**
     * Creates new form Kasir
     */
    public Kasir() {
        initComponents();
    }
}

```

```

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {
    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    Load = new javax.swing.JButton();
    Insert = new javax.swing.JButton();
    Update = new javax.swing.JButton();
    Delete = new javax.swing.JButton();
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null},
            {null, null, null, null}
        },
        new String [] {
            "Kode", "Nama", "Harga", "Stock"
        }
    ));

```

```

jScrollPane1.setViewportViewView(jTable1);

Load.setText("Load");

Load.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        LoadActionPerformed(evt);

    }

});

Insert.setText("Insert");

Insert.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        InsertActionPerformed(evt);

    }

});

Update.setText("Update");

Update.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        UpdateActionPerformed(evt);

    }

});

Delete.setText("Delete");

Delete.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        DeleteActionPerformed(evt);

    }

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

```



```

getContentPane().setLayout(layout);

layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 664,
Short.MAX_VALUE)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(Load)
                    .addGap(18, 18, 18)
                    .addComponent(Insert)
                    .addGap(18, 18, 18)
                    .addComponent(Update)
                    .addGap(18, 18, 18)
                    .addComponent>Delete)
                    .addGap(0, 0, Short.MAX_VALUE)))
            .addContainerGap())
        );

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 280,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(36, 36, 36)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(Load)

```

```

        .addComponent(Insert)

        .addComponent(Update)

        .addComponent>Delete))

        .addGap(0, 10, Short.MAX_VALUE)

    );

    pack();

} // </editor-fold>

private void LoadActionPerformed(java.awt.event.ActionEvent evt) {
try (Connection conn = DatabaseConnection.DatabaseConnection(); Statement stmt =
conn.createStatement(); ResultSet rs = stmt.executeQuery("SELECT * FROM products")) {

    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

    model.setRowCount(0);

    while (rs.next()) {

        Vector<Object> row = new Vector<>();

        row.add(rs.getInt("kode"));

        row.add(rs.getString("nama"));

        row.add(rs.getInt("harga"));

        row.add(rs.getInt("stock"));

        model.addRow(row);

    }

} catch (SQLException ex) {

    JOptionPane.showMessageDialog(this, "Error saat memuat data: " + ex.getMessage(),
"Database Error", JOptionPane.ERROR_MESSAGE);

    ex.printStackTrace();

}

}

private void InsertActionPerformed(java.awt.event.ActionEvent evt) {

```

```

        new FormInput().setVisible(true);
    }

    private void UpdateActionPerformed(java.awt.event.ActionEvent evt) {
        new FormUpdate().setVisible(true);
    }

    private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
        int selectedRow = jTable1.getSelectedRow();
        if (selectedRow == -1) {
            JOptionPane.showMessageDialog(this, "Pilih dulu data yang ingin dihapus.");
            return;
        }

        int confirm = JOptionPane.showConfirmDialog(this, "Yakin ingin menghapus data ini?",
            "Konfirmasi Hapus", JOptionPane.YES_NO_OPTION);
        if (confirm == JOptionPane.YES_OPTION) {
            try {
                int id = Integer.parseInt(jTable1.getValueAt(selectedRow, 0).toString()); // Ambil ID dari kolom
                pertama (index 0)

                Connection conn = DatabaseConnection.DatabaseConnection();
                String sql = "DELETE FROM products WHERE kode = ?";
                PreparedStatement stmt = conn.prepareStatement(sql);
                stmt.setInt(1, id);
                stmt.executeUpdate();

                // Hapus dari JTable juga
                DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
                model.removeRow(selectedRow);

                JOptionPane.showMessageDialog(this, "Data berhasil dihapus!");
            } catch (SQLException e) {

```

```

        JOptionPane.showMessageDialog(this, "Gagal hapus data: " + e.getMessage());
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Kasir.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Kasir.class.getName()).log(java.util.logging.Level.SEVERE,
            null, ex);
    }
}

```

```

    } catch (IllegalAccessException ex) {

        java.util.logging.Logger.getLogger(Kasir.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

        java.util.logging.Logger.getLogger(Kasir.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

    }

//</editor-fold>

/* Create and display the form */

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new Kasir().setVisible(true);

    }

});

}

// Variables declaration- do not modify

private javax.swing.JButton Delete;

private javax.swing.JButton Insert;

private javax.swing.JButton Load;

private javax.swing.JButton Update;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

// End of variables declaration

}

```

- ANALISA

Kelas Kasir ini digunakan untuk mengelola data produk dengan fitur **LOADING, INSERT, UPDATE, DELETE** data dari tabel products. Kode ini menggunakan Swing dan JDBC untuk integrasi database, dengan logika utama di Load dan Delete, serta delegasi ke FormInput dan FormUpdate untuk Insert dan Update.

C. FORM INPUT

```
package projectpbo;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import java.util.Vector;

import javax.swing.table.DefaultTableModel;

import java.sql.Statement;

import java.sql.ResultSet;


/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */


/**
 *
 * @author ASUS
 */
```

```

public class Kasir extends javax.swing.JFrame {

    /**
     * Creates new form Kasir
     */
    public Kasir() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        Load = new javax.swing.JButton();
        Insert = new javax.swing.JButton();
        Update = new javax.swing.JButton();
        Delete = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```

```

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Kode", "Nama", "Harga", "Stock"
    }
));
jScrollPane1.setViewportViewView(jTable1);

Load.setText("Load");
Load.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        LoadActionPerformed(evt);
    }
});

Insert.setText("Insert");
Insert.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        InsertActionPerformed(evt);
    }
}

```



```
});
```

```
Update.setText("Update");
```

```
Update.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        UpdateActionPerformed(evt);  
    }  
});
```

```
Delete.setText("Delete");
```

```
Delete.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        DeleteActionPerformed(evt);  
    }  
});
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());  
getContentPane().setLayout(layout);  
layout.setHorizontalGroup(  
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(layout.createSequentialGroup()  
            .addGap(10, 10, 10)  
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 664,  
Short.MAX_VALUE)  
                .addGroup(layout.createSequentialGroup()  
                    .addGap(10, 10, 10)  
                    .addComponent(updateButton, javax.swing.GroupLayout.DEFAULT_SIZE, 100, false)  
                    .addGap(10, 10, 10)  
                    .addComponent(deleteButton, javax.swing.GroupLayout.DEFAULT_SIZE, 100, false)  
                )  
            )  
        )  
);
```

```

        .addComponent(Load)
        .addGap(18, 18, 18)
        .addComponent(Insert)
        .addGap(18, 18, 18)
        .addComponent(Update)
        .addGap(18, 18, 18)
        .addComponent>Delete)
        .addGap(0, 0, Short.MAX_VALUE)))
    .addContainerGap()
);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 280,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(36, 36, 36)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(Load)
                .addComponent(Insert)
                .addComponent(Update)
                .addComponent>Delete))
            .addGap(0, 10, Short.MAX_VALUE))
        );

pack();
} // </editor-fold>

```

```

private void LoadActionPerformed(java.awt.event.ActionEvent evt) {

try (Connection conn = DatabaseConnection.DatabaseConnection(); Statement stmt =
conn.createStatement(); ResultSet rs = stmt.executeQuery("SELECT * FROM products")) {

    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

    model.setRowCount(0);

    while (rs.next()) {

        Vector<Object> row = new Vector<>();

        row.add(rs.getInt("kode"));

        row.add(rs.getString("nama"));

        row.add(rs.getInt("harga"));

        row.add(rs.getInt("stock"));

        model.addRow(row);

    }

} catch (SQLException ex) {

    JOptionPane.showMessageDialog(this, "Error saat memuat data: " + ex.getMessage(),
"Database Error", JOptionPane.ERROR_MESSAGE);

    ex.printStackTrace();

}

}

private void InsertActionPerformed(java.awt.event.ActionEvent evt) {

    new FormInput().setVisible(true);

}

```

```

private void UpdateActionPerformed(java.awt.event.ActionEvent evt) {
    new FormUpdate().setVisible(true);
}

private void DeleteActionPerformed(java.awt.event.ActionEvent evt) {
    int selectedRow = jTable1.getSelectedRow();

    if (selectedRow == -1) {
        JOptionPane.showMessageDialog(this, "Pilih dulu data yang ingin dihapus.");
        return;
    }

    int confirm = JOptionPane.showConfirmDialog(this, "Yakin ingin menghapus data ini?",
        "Konfirmasi Hapus", JOptionPane.YES_NO_OPTION);

    if (confirm == JOptionPane.YES_OPTION) {
        try {
            int id = Integer.parseInt(jTable1.getValueAt(selectedRow, 0).toString()); // Ambil ID dari
            kolom pertama (index 0)

            Connection conn = DatabaseConnection.DatabaseConnection();

            String sql = "DELETE FROM products WHERE kode = ?";

            PreparedStatement stmt = conn.prepareStatement(sql);

            stmt.setInt(1, id);

            stmt.executeUpdate();

            // Hapus dari jTable juga

```

```

        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

        model.removeRow(selectedRow);

        JOptionPane.showMessageDialog(this, "Data berhasil dihapus!");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Gagal hapus data: " + e.getMessage());
    }
}

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see
     * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    }
}

```

```

        }
    }
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Kasir.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Kasir.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Kasir.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Kasir.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Kasir().setVisible(true);
    }
});
}

```

```
// Variables declaration- do not modify

private javax.swing.JButton Delete;

private javax.swing.JButton Insert;

private javax.swing.JButton Load;

private javax.swing.JButton Update;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

// End of variables declaration
}
```

- ANALISA

Kelas FormInput adalah digunakan untuk menambah data produk ke database dengan mudah. Yang terdiri dari empat kolom input: **Kode**, **Nama**, **Harga**, dan **Stock**, yang masing-masing diwakili oleh label dan text field. Pengguna mengisi data pada kolom tersebut, lalu menekan tombol **INSERT** untuk menyimpan data ke database. Saat tombol INSERT ditekan, aplikasi akan mengambil data dari setiap text field, mengonversi ke tipe data yang sesuai (angka atau teks), lalu menjalankan perintah SQL untuk memasukkan data ke tabel products menggunakan PreparedStatement. Jika proses berhasil, muncul pesan sukses dan form akan tertutup otomatis; jika gagal, akan muncul pesan error yang menjelaskan masalahnya, misalnya salah input angka atau masalah koneksi database. Semua komponen form diatur menggunakan GroupLayout agar tampil rapi. Koneksi ke database dilakukan dengan memanggil method dari kelas DatabaseConnection. Validasi input hanya sebatas memastikan input angka bisa dikonversi, tanpa pemeriksaan lebih lanjut seperti data kosong atau nilai negatif. Secara singkat, kelas ini berfungsi sebagai form sederhana untuk menambah data produk ke database, dengan proses input yang langsung, notifikasi hasil, dan desain antarmuka yang mudah digunakan.

D. FORM UPDATE

```
/*
```

```
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
license
```

```
* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
```

```

*/

package projectpbo;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import java.util.Vector;

import javax.swing.table.DefaultTableModel;

import java.sql.Statement;

import java.sql.ResultSet;


/**
 *
 * @author ASUS
 */

public class FormUpdate extends javax.swing.JFrame {


    /**
     * Creates new form FormUpdate
     */

    public FormUpdate() {

        initComponents();

    }


    /**

```



```

* This method is called from within the constructor to initialize the form.
* WARNING: Do NOT modify this code. The content of this method is always
* regenerated by the Form Editor.
*/

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel4 = new javax.swing.JLabel();
    Stock = new javax.swing.JTextField();
    INSERT = new javax.swing.JButton();
    CANCEL = new javax.swing.JButton();
    jLabel2 = new javax.swing.JLabel();
    tfKodeBaru = new javax.swing.JTextField();
    jLabel1 = new javax.swing.JLabel();
    tfNama = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    Harga = new javax.swing.JTextField();
    jLabel5 = new javax.swing.JLabel();
    tfKode = new javax.swing.JTextField();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    jLabel4.setText("Stock");
    Stock.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            StockActionPerformed(evt);
        }
    });

```

```

    }
});
INSERT.setText("INSERT");
INSERT.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        INSERTActionPerformed(evt);
    }
});
CANCEL.setText("CANCEL");
jLabel2.setText("Kode Baru");
tfKodeBaru.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tfKodeBaruActionPerformed(evt);
    }
});
jLabel1.setText("Nama");
tfNama.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tfNamaActionPerformed(evt);
    }
});
jLabel3.setText("Harga");
jLabel5.setText("Kode Lama");
tfKode.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        tfKodeActionPerformed(evt);
    }
});

```



```

        .addGap(40, 40, 40)

        .addComponent(jLabel1)

        .addGap(18, 18, 18)

        .addComponent(tfNama, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(55, 55, 55)

        .addComponent(jLabel3)

        .addGap(18, 18, 18)

        .addComponent(Harga, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(68, 68, 68)

        .addComponent(jLabel4)

        .addGap(18, 18, 18)

        .addComponent(Stock, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(14, Short.MAX_VALUE))
    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup())

                .addGap(63, 63, 63)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(jLabel1)

                .addComponent(tfNama, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(jLabel3)

        .addComponent(Harga, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel4)

        .addComponent(Stock, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())

        .addContainerGap()

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel5)

            .addComponent(tfKode, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(26, 26, 26)))

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel2)

            .addComponent(tfKodeBaru, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(INSERT)

                .addComponent(CANCEL))

            .addContainerGap(24, Short.MAX_VALUE))

    );

    pack();
} // </editor-fold>

```

```

private void StockActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void tfKodeBaruActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void tfNamaActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void tfKodeActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

private void INSERTActionPerformed(java.awt.event.ActionEvent evt) {

int kodeLama = Integer.parseInt(tfKode.getText());

int kodeBaru = Integer.parseInt(tfKodeBaru.getText());

String nama = tfNama.getText();

int harga = Integer.parseInt(Harga.getText());

int stok = Integer.parseInt(Stock.getText());

try {

    Connection conn = DatabaseConnection.DatabaseConnection(); // pastikan class koneksi sudah
    sesuai

    String sql = "UPDATE products SET kode = ?, nama = ?, harga = ?, stock = ? WHERE kode = ?";

    PreparedStatement stmt = conn.prepareStatement(sql);

    stmt.setInt(1, kodeBaru);

    stmt.setString(2, nama);

    stmt.setInt(3, harga);

```

```

stmt.setInt(4, stok);
stmt.setInt(5, kodeLama);

int rowsUpdated = stmt.executeUpdate();
if (rowsUpdated > 0)
{
    JOptionPane.showMessageDialog(this, "Data berhasil diupdate.");
    this.dispose();
} else {
    JOptionPane.showMessageDialog(this, "Data gagal diupdate. Kode lama tidak ditemukan.");
}
stmt.close();
conn.close();
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(this, "Harga dan Stock harus berupa angka.");
} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Kesalahan database: " + e.getMessage());
}
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

```

```

    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(FormUpdate.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(FormUpdate.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(FormUpdate.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(FormUpdate.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }

    //</editor-fold>

    /* Create and display the form */

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new FormUpdate().setVisible(true);

```



```

    }
    });
}

// Variables declaration- do not modify

private javax.swing.JButton CANCEL;

private javax.swing.JTextField Harga;

private javax.swing.JButton INSERT;

private javax.swing.JTextField Stock;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JTextField tfKode;

private javax.swing.JTextField tfKodeBaru;

private javax.swing.JTextField tfNama;

// End of variables declaration
}

```

- **ANALISIS**

Form ini merupakan jendela GUI berbasis Swing, yang dirancang untuk memperbarui data produk di tabel products pada database MySQL db_toko yang menyediakan formulir dengan teks field untuk Kode Lama, Kode Baru, Nama, Harga, dan Stock, serta tombol INSERT dan CANCEL. Method INSERTActionPerformed mengelola pembaruan data menggunakan PreparedStatement dengan query UPDATE, mengambil input, dan menampilkan pesan sukses atau error via JOptionPane, meskipun penutupan koneksi dilakukan secara manual