

NAMA : TEGUH FIRMANSYAH  
NIM : 23.51.0031  
JURUSAN : SISTEM INFORMASI

## UAS Pemrograman Berorientasi Objek Kelas B

Form Barang

controllerBarang.java

### FORM BARANG

KODE

NAMA

JUMLAH

HARGA

MEREK

Simpan

Ubah

Bersih

Hapus

Keluar

Kode Barang	Nama Barang	Jumlah	Harga	Merek

```
5 package controller;
6
7 import dao.daoBarang;
8 import java.util.List;
9 import javax.swing.JOptionPane;
10 import model.*;
11 import view.FormBarang;
12
13 /**
14  *
15  * @author USER
16  */
17 public class controllerBarang {
18     FormBarang frame;
19     List<Barang> listBrg;
20     daoBarang daoBrg = new daoBarang();
21     Barang brg = new Barang();
22
23     public controllerBarang(FormBarang frame) {
24         this.frame = frame;
25         listBrg = daoBrg.getData();
26     }
27
28     public void tampil_tabel() {
29         TableModelBarang tabelBrg = new TableModelBarang(listBrg);
30         frame.getTblBarang().setModel(dataModel: tabelBrg);
31     }
```

```

33 public void tambahData() {
34     if (frame.getTxtKode().getText().equals(anObject: "")) {
35         JOptionPane.showMessageDialog(parentComponent: null, message: "Kode barang belum diisi");
36     } else {
37         brg.setKode(kode: frame.getTxtKode().getText());
38         brg.setNama(nama: frame.getTxtNama().getText());
39         brg.setJumlah(jumlah: Integer.parseInt(s: frame.getTxtJumlah().getText()));
40         brg.setHarga(harga: Integer.parseInt(s: frame.getTxtHarga().getText()));
41         brg.setMerek(merek: frame.getTxtMerek().getText());
42         daoBrg.tambah(brg);
43         JOptionPane.showMessageDialog(parentComponent: frame, message: "Data berhasil di simpan");
44     }
45 }
46
47 public void ubahData() {
48     if (frame.getTxtKode().getText().equals(anObject: "")) {
49         JOptionPane.showMessageDialog(parentComponent: null, message: "Kode barang belum diisi");
50     } else {
51         brg.setKode(kode: frame.getTxtKode().getText());
52         brg.setNama(nama: frame.getTxtNama().getText());
53         brg.setJumlah(jumlah: Integer.parseInt(s: frame.getTxtJumlah().getText()));
54         brg.setHarga(harga: Integer.parseInt(s: frame.getTxtHarga().getText()));
55         brg.setMerek(merek: frame.getTxtMerek().getText());
56         daoBrg.ubah(brg);
57         JOptionPane.showMessageDialog(parentComponent: frame, message: "Data berhasil di ubah");
58     }
59 }
60
61 public void hapusData() {
62     brg.setKode(kode: frame.getTxtKode().getText());
63     daoBrg.hapus(brg);
64     JOptionPane.showMessageDialog(parentComponent: frame, message: "Data berhasil di hapus");
65 }
66
67 public void bersih() {
68     frame.setTxtKode(txtKode: "");
69     frame.setTxtNama(txtNama: "");
70     frame.setTxtJumlah(txtJumlah: 0);
71     frame.setTxtHarga(txtHarga: 0);
72     frame.setTxtMerek(txtMerek: "");
73 }
74
75 public void keluar() {
76     frame.dispose();
77 }
78 }

```

## koneksi.java

```
5  package controller;
6
7  import com.mysql.cj.jdbc.MysqlDataSource;
8  import java.sql.Connection;
9
10 /**
11  *
12  * @author USER
13  */
14 public class koneksi {
15     static Connection kon;
16
17     public static Connection connection() {
18         if (kon == null) {
19             MysqlDataSource data = new MysqlDataSource();
20             data.setDatabaseName(dbName: "toko");
21             data.setUser(userID: "root");
22             data.setPassword(pass: "");
23             try {
24                 kon = data.getConnection();
25                 System.out.println(x: "sudah konek");
26             } catch (Exception e) {
27                 System.out.println(x: "tidak konek");
28             }
29         }
30     }
31     return kon;
32 }
```

## daoBarang.java

```
32 public daoBarang(){
33     connection = koneksi.connection();
34 }
35
36 public void tambah(Barang brg){
37     PreparedStatement statement = null;
38     try {
39         statement = connection.prepareStatement(string: insert);
40         statement.setString(i: 1, string: brg.getKode());
41         statement.setString(i: 2, string: brg.getNama());
42         statement.setInt(i: 3, i1: brg.getJumlah());
43         statement.setInt(i: 4, i1: brg.getHarga());
44         statement.setString(i: 5, string: brg.getMerek());
45         statement.executeUpdate();
46     } catch (SQLException ex) {
47
48     }
49 }
50
51 public void ubah(Barang brg){
52     PreparedStatement statement = null;
53     try {
54         statement = connection.prepareStatement(string: update);
55         statement.setString(i: 1, string: brg.getNama());
56         statement.setInt(i: 2, i1: brg.getJumlah());
57         statement.setInt(i: 3, i1: brg.getHarga());
58         statement.setString(i: 4, string: brg.getMerek());
59         statement.setString(i: 5, string: brg.getKode());
```

```

57         statement.setInt(i: 3, i1: brg.getHarga());
58         statement.setString(i: 4, string: brg.getMerek());
59         statement.setString(i: 5, string: brg.getKode());
60         statement.executeUpdate();
61     } catch (SQLException ex) {
62     }
63 }
64
65
66 public void hapus(Barang brg) {
67     PreparedStatement statement = null;
68     try {
69         statement = connection.prepareStatement(string: delete);
70         statement.setString(i: 1, string: brg.getKode());
71         statement.executeUpdate();
72     } catch (SQLException ex) {
73     }
74 }
75
76
77 public void tampil(Barang brg) {
78     PreparedStatement statement = null;
79     try {
80         statement = connection.prepareStatement(string: selectData);
81         statement.setString(i: 1, string: brg.getKode());
82         statement.executeUpdate();

```



```

82         statement.executeUpdate();
83     }catch (SQLException ex){
84         JOptionPane.showMessageDialog(parentComponent:null, message:ex);
85     }
86 }
87
88 public List<Barang> getData(){
89     List<Barang> listBrg = null;
90     try {
91         listBrg = new ArrayList<>();
92         Statement st = connection.createStatement();
93         ResultSet rs = st.executeQuery(string: select);
94         while (rs.next()){
95             Barang brg = new Barang();
96             brg.setKode(kode: rs.getString(string: "kode"));
97             brg.setNama(nama: rs.getString(string: "nama"));
98             brg.setJumlah(jumlah: rs.getInt(string: "jumlah"));
99             brg.setHarga(harga: rs.getInt(string: "harga"));
100            brg.setMerek(merek: rs.getString(string: "merek"));
101            listBrg.add(=: brg);
102        }
103    }catch (SQLException ex){
104        JOptionPane.showMessageDialog(parentComponent:null, message:ex);
105    }
106    return listBrg;
107 }
108
109 public int cekKode(String kode){
110     PreparedStatement statement = null;
111     int ketemu = 0;
112     try {
113         statement = connection.prepareStatement(string: selectData);
114         statement.setString(i: 1, string: kode);
115         ResultSet rs = statement.executeQuery();
116         while(rs.next()) {
117             ketemu++;
118         }
119     }catch (SQLException ex){
120
121     }
122     return ketemu;
123 }
124

```

## Barang.java

```
11 public class Barang {
12     private String kode, nama, merek;
13     private int jumlah, harga;
14
15     public String getKode(){
16         return kode;
17     }
18
19     public void setKode(String kode){
20         this.kode = kode;
21     }
22
23     public String getNama(){
24         return nama;
25     }
26
27     public void setNama(String nama){
28         this.nama = nama;
29     }
30
31     public String getMerek() {
32         return merek;
33     }
34
35     public void setMerek(String merek) {
36         this.merek = merek;
37     }
```



```
28         this.nama = nama;
29     }
30
31     public String getMerek() {
32         return merek;
33     }
34
35     public void setMerek(String merek) {
36         this.merek = merek;
37     }
38
39     public int getJumlah() {
40         return jumlah;
41     }
42
43     public void setJumlah(int jumlah) {
44         this.jumlah = jumlah;
45     }
46
47     public int getHarga() {
48         return harga;
49     }
50
51     public void setHarga(int harga) {
52         this.harga = harga;
53     }
54
55 }
```

## TabelModelBarang.java

```
7  import java.util.List;
8  import javax.swing.table.AbstractTableModel;
9
10 /**
11  *
12  * @author USER
13  */
14 public class TabelModelBarang extends AbstractTableModel{
15
16     List<Barang> listBrg;
17
18     public TabelModelBarang(List<Barang> listBrg){
19         this.listBrg = listBrg;
20     }
21
22     @Override
23     public int getRowCount() {
24         return listBrg.size();
25     }
26
27     @Override
28     public int getColumnCount() {
29         return 5;
30     }
```

```

32      @Override
33      public Object getValueAt(int row, int column) {
34          return switch (column) {
35              case 0 -> listBrg.get(index: row).getKode();
36              case 1 -> listBrg.get(index: row).getNama();
37              case 2 -> listBrg.get(index: row).getJumlah();
38              case 3 -> listBrg.get(index: row).getHarga();
39              case 4 -> listBrg.get(index: row).getMerek();
40              default -> null;
41          };
42      }
43
44      @Override
45      public String getColumnName(int column) {
46          return switch (column) {
47              case 0 -> "Kode Barang";
48              case 1 -> "Nama Barang";
49              case 2 -> "Jumlah";
50              case 3 -> "Harga";
51              case 4 -> "Merek";
52              default -> null;
53          };
54      }
55  }
56

```

## FormBarang.java

```
7  import controller.*;
8  import javax.swing.JTable;
9  import javax.swing.JTextField;
10
11  /**
12   *
13   * @author User
14   */
15  public class FormBarang extends javax.swing.JFrame {
16      controllerBarang cBarang;
17      /**
18       * Creates new form FormBarang
19       */
20      public FormBarang() {
21          initComponents();
22          cBarang = new controllerBarang(frame: this);
23          cBarang.tampil_tabel();
24      }
25
26      public void tampilkantabel(){
27          cBarang = new controllerBarang(frame: this);
28          cBarang.tampil_tabel();
29      }
30  }
```

```
32 public JTable getTblBarang() {
33     return tblBarang;
34 }
35
36 public void setTblBarang(JTable tblBarang) {
37     this.tblBarang = tblBarang;
38 }
39
40 public JTextField getTxtHarga() {
41     return txtHarga;
42 }
43
44 public void setTxtHarga(int txtHarga) {
45     this.txtHarga.setText(Integer.toString(txtHarga));
46 }
47
48 public JTextField getTxtJumlah() {
49     return txtJumlah;
50 }
51
52 public void setTxtJumlah(int txtJumlah) {
53     this.txtJumlah.setText(Integer.toString(txtJumlah));
54 }
55
56 public JTextField getTxtKode() {
57     return txtKode;
58 }
```

```
56 public JTextField getTxtKode() {
57     return txtKode;
58 }
59
60 public void setTxtKode(String txtKode) {
61     this.txtKode.setText(txtKode);
62 }
63
64 public JTextField getTxtMerek() {
65     return txtMerek;
66 }
67
68 public void setTxtMerek(String txtMerek) {
69     this.txtMerek.setText(txtMerek);
70 }
71
72 public JTextField getTxtNama() {
73     return txtNama;
74 }
75
76 public void setTxtNama(String txtNama) {
77     this.txtNama.setText(txtNama);
78 }
```



```

322     private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
323         cBarang.tambahData();
324         tampilkantabel();
325         cBarang.bersih();
326         txtKode.requestFocus();
327     }

329     private void btnUbahActionPerformed(java.awt.event.ActionEvent evt) {
330         cBarang.ubahData();
331         tampilkantabel();
332         cBarang.bersih();
333         txtKode.requestFocus();
334     }

336     private void btnBatalActionPerformed(java.awt.event.ActionEvent evt) {
337         cBarang.bersih();
338     }

340     private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
341         cBarang.hapusData();
342         tampilkantabel();
343         cBarang.bersih();
344         txtKode.requestFocus();
345     }

336         cBarang.bersih();
337     }
338

340     private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
341         cBarang.hapusData();
342         tampilkantabel();
343         cBarang.bersih();
344         txtKode.requestFocus();
345     }

347     private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt) {
348         cBarang.keluar();
349     }

351     private void tblBarangMouseClicked(java.awt.event.MouseEvent evt) {
352         // TODO add your handling code here:
353         int pilih = tblBarang.getSelectedRow();
354         txtKode.setText(tblBarang.getModel().getValueAt(rowIndex:pilih, columnIndex:0).toString());
355         txtNama.setText(tblBarang.getModel().getValueAt(rowIndex:pilih, columnIndex:1).toString());
356         txtJumlah.setText(tblBarang.getModel().getValueAt(rowIndex:pilih, columnIndex:2).toString());
357         txtHarga.setText(tblBarang.getModel().getValueAt(rowIndex:pilih, columnIndex:3).toString());
358         txtMerek.setText(tblBarang.getModel().getValueAt(rowIndex:pilih, columnIndex:4).toString());
359     }

```

## Form Cari Barang

FORM CARI BARANG				
Input Nama Barang : <input type="text"/>				
<input type="button" value="Cari"/> <input type="button" value="Keluar"/>				
Kode Barang	Nama Barang	Jumlah	Harga	Merek

## ControllerCariBarang.java

```
7 //import dao.daoCariBarang;
8 import dao.daoCariBarang;
9 import java.util.List;
10 import javax.swing.JOptionPane;
11 import model.*;
12 //import view.FormBarang;
13 import view.FormCariBarang;
14 /**
15  *
16  * @author User
17  */
18 public class controllerCariBarang {
19     FormCariBarang frame;
20     List<Barang> listBrg;
21     List<Barang> listCariBrg;
22     daoCariBarang daoCariBrg = new daoCariBarang();
23     // Barang brg = new Barang();
24
25     public controllerCariBarang(FormCariBarang frame){
26         this.frame = frame;
27         listBrg = daoCariBrg.getData();
28         listCariBrg = daoCariBrg.getCariData(NamaBrg:frame.getTxtNamaBrg().getText());
29     }
30
31     public void tampil_tabel(){
32         TabelModelCariBarang tabelBrg = new TabelModelCariBarang(listBrg);
33         frame.getTblBarang().setModel(dataModel: tabelBrg);
34     }
35
36     public void tampilFilter_tabel(){
37         TabelModelCariBarang tabelBrg = new TabelModelCariBarang(listBrg:listCariBrg);
38         frame.getTblBarang().setModel(dataModel: tabelBrg);
39     }
40
41     public void cekNamaBrg() {
42         if (frame.getTxtNamaBrg().getText().equals(anObject: "")) {
43             JOptionPane.showMessageDialog(parentComponent:frame, message:"Nama Barang belum diisi");
44         } else {
45             daoCariBrg.getCariData(NamaBrg:frame.getTxtNamaBrg().getText());
46             // daoCariBrg.cekNama(frame.getTxtNamaBrg().getText());
47             tampilFilter_tabel();
48             //int row = daoCariBrg.cekNama(frame.getTxtNamaBrg().getText());
49             // JOptionPane.showMessageDialog(frame, "berhasil");
50             //daoCariBrg.getCariData();
51         }
52     }
53
54     public void keluar(){
55         frame.dispose();
56     }
57 }
```

## daoCariBarang.java

```
7  import controller.koneksi;
8  import java.sql.Connection;
9  import java.sql.PreparedStatement;
10 import java.sql.ResultSet;
11 import java.sql.SQLException;
12 import java.sql.Statement;
13 import java.util.ArrayList;
14 import java.util.List;
15 import javax.swing.JOptionPane;
16 import model.*;
17 /**
18  *
19  * @author User
20  */
21 public class daoCariBarang {
22     Connection connection;
23     final String select = "SELECT * FROM barang ORDER BY nama ASC;";
24     final String selectData = "SELECT * FROM barang WHERE nama LIKE %?%";
25     final String cari = "SELECT * FROM barang WHERE nama LIKE ?";
26
27     public daoCariBarang(){
28         connection = koneksi.connection();
29     }
30
31     public void tampil(Barang brg){
32         PreparedStatement statement = null;
33         try {
34             statement = connection.prepareStatement(string: select);
```

```

34         statement = connection.prepareStatement(string: select);
35         statement.setString(i: 1, string: brg.getKode());
36         statement.executeUpdate();
37     } catch (SQLException ex) {
38         JOptionPane.showMessageDialog(parentComponent: null, message: ex);
39     }
40 }
41
42 public List<Barang> getData() {
43     List<Barang> listBrg = null;
44     try {
45         listBrg = new ArrayList<>();
46         Statement st = connection.createStatement();
47         ResultSet rs = st.executeQuery(string: select);
48         while (rs.next()) {
49             Barang brg = new Barang();
50             brg.setKode(kode: rs.getString(string: "kode"));
51             brg.setNama(nama: rs.getString(string: "nama"));
52             brg.setJumlah(jumlah: rs.getInt(string: "jumlah"));
53             brg.setHarga(harga: rs.getInt(string: "harga"));
54             brg.setMerek(merek: rs.getString(string: "merek"));
55             listBrg.add(e: brg);
56         }
57     } catch (SQLException ex) {
58         JOptionPane.showMessageDialog(parentComponent: null, message: ex);
59     }
60     return listBrg;
61 }

```

```

62
63 public List<Barang> getCariData(String NamaBrg){
64     PreparedStatement statement = null;
65     List<Barang> listBrg = null;
66     try {
67         listBrg = new ArrayList<>();
68         // Statement st = connection.createStatement();
69         // ResultSet rs = st.executeQuery(selectData);
70         statement = connection.prepareStatement(string: cari);
71         statement.setString(1, "%" + NamaBrg + "%");
72         ResultSet rs = statement.executeQuery();
73         while (rs.next()){
74             Barang brg = new Barang();
75             brg.setKode(kode: rs.getString(string: "kode"));
76             brg.setNama(nama: rs.getString(string: "nama"));
77             brg.setJumlah(jumlah: rs.getInt(string: "jumlah"));
78             brg.setHarga(harga: rs.getInt(string: "harga"));
79             brg.setMerek(merek: rs.getString(string: "merek"));
80             listBrg.add(brg);
81         }
82     } catch (SQLException ex){
83         JOptionPane.showMessageDialog(parentComponent:null, message:ex);
84     }
85     return listBrg;
86 }

```



## TabelModelCariBarang.java

```
7  import java.util.List;
8  import javax.swing.table.AbstractTableModel;
9  /**
10   *
11   * @author User
12   */
13  public class TabelModelCariBarang extends AbstractTableModel {
14      List<Barang> listBrg;
15
16      public TabelModelCariBarang(List<Barang> listBrg) {
17          this.listBrg = listBrg;
18      }
19
20      @Override
21      public int getRowCount() {
22          return listBrg.size();
23      }
24
25      @Override
26      public int getColumnCount() {
27          return 5;
28      }
29
30      @Override
31      public Object getValueAt(int row, int column) {
32          return switch (column) {
33              case 0 -> listBrg.get(index: row).getKode();
```

```

30      @Override
31      public Object getValueAt(int row, int column){
32          return switch (column){
33              case 0 -> listBrg.get(index: row).getKode();
34              case 1 -> listBrg.get(index: row).getNama();
35              case 2 -> listBrg.get(index: row).getJumlah();
36              case 3 -> listBrg.get(index: row).getHarga();
37              case 4 -> listBrg.get(index: row).getMerek();
38              default -> null;
39          };
40      }
41
42      @Override
43      public String getColumnName(int column){
44          return switch (column){
45              case 0 -> "Kode Barang";
46              case 1 -> "Nama Barang";
47              case 2 -> "Jumlah";
48              case 3 -> "Harga";
49              case 4 -> "Merek";
50              default -> null;
51          };
52      }
53  }

```

**FormCariBarang.java**

```

7  import controller.controllerCariBarang;
8  import javax.swing.JTable;
9  import javax.swing.JTextField;
10 import model.*;
11  /**
12   *
13   * @author USER
14   */
15  public class FormCariBarang extends javax.swing.JFrame {
16      controllerCariBarang cBarang;
17      /**
18       * Creates new form FormCariBarang
19       */
20      public FormCariBarang() {
21          initComponents();
22          cBarang = new controllerCariBarang(frame: this);
23          cBarang.tampil_tabel();
24      }
25
26      public void tampilkantabel(){
27          cBarang = new controllerCariBarang(frame: this);
28          cBarang.tampil_tabel();
29      }
30
31
32      public JTable getTblBarang() {
33          return tblBarang;
34      }

```

```

36 public void setTblBarang(JTable tblBarang) {
37     this.tblBarang = tblBarang;
38 }
39
40 public JTextField getTxtNamaBrg() {
41     return txtCariNama;
42 }
43
44 public void setTxtNamaBrg(String txtNamaBrg) {
45     this.txtCariNama.setText(txtNamaBrg);
46 }
47
48 /**
49  * This method is called from within the constructor to initialize the form.
50  * WARNING: Do NOT modify this code. The content of this method is always
51  * regenerated by the Form Editor.
52  */
53 @SuppressWarnings("unchecked")
54 Generated Code
194
196 private void tblBarangMouseClicked(java.awt.event.MouseEvent evt) {
197     // TODO add your handling code here:
198 }
199
200 private void btnCariActionPerformed(java.awt.event.ActionEvent evt) {
201     // TODO add your handling code here:
202     cBarang.cekNamaBrg();
203 }
204
205 private void btnCariActionPerformed(java.awt.event.ActionEvent evt) {
206     // TODO add your handling code here:
207     cBarang.cekNamaBrg();
208     tampilkantabel();
209 }
210
211 private void btnKeluarActionPerformed(java.awt.event.ActionEvent evt) {
212     // TODO add your handling code here:
213     cBarang.keluar();
214 }
215
216 /**
217  * @param args the command line arguments
218  */
219 public static void main(String args[]) {
220     /* Set the Nimbus look and feel */
221     Look and feel setting code (optional)
222
223     /* Create and display the form */
224     java.awt.EventQueue.invokeLater(new Runnable() {
225         public void run() {
226             new FormCariBarang().setVisible(true);
227         }
228     });
229 }

```