

Interview and conversation with Client Ms. Vinsen (Ms)

Ms: Kid, do you think your friends will remember all the musical terms I mentioned earlier?

Me: If you ask if I can remember the term miss given earlier, I can miss. But, for the other friends, I don't know about them.

Ms: I think I need a dictionary that can be used to get them to remember music terms.

Me: But in my opinion it is a less efficient miss. It will be heavy if carried in a bag including our notebooks.

Ms: Or maybe the dictionary can be digital. Like an application that they can access using their macbook.

Me: Sounds great, Miss. It will be more helpful for us.

Ms: But, the thing is who will make it? I mean, it will be more costly if I hire a developer right.

Me: I can help you, Miss, for sure.

Ms: Can you? How?

Me: I learned Java in computer science and I know how to create an application. Indeed, I have never made a dictionary, but I can try. It can improve my skills.

Ms: Awesome. I will send you the term and vocabulary that you must add to the dictionary.

Me: Okay, Miss. Do you need other things that must be added in the dictionary? Maybe a bunch of music scores or another?

Ms: Music scores, it sounds too complicated. Maybe you can add a dictionary to explain music instruments. Don't forget to make it as creative as possible.

Me: Alright, Miss. Maybe I need quite a long time, I will try to finish it as soon as possible.

Ms: No problem. As long as you can finish it without disturbing your other subject, it will be better. Thanks, kid.

Me: Yups, you're welcome, Miss.

```
package kamusia;
```

```
public class Kamusla {
```

```
    public static void main(String[] args) {
```

```
        Register rg = new Register();
```

```
        rg.setVisible(true);
```

```
        rg.pack();
```

```
        rg.setLocationRelativeTo(null);
```

```
        rg.setDefaultCloseOperation(Register.EXIT_ON_CLOSE);
```

```
    }
```

```
}
```

```
package kamusia;

import javax.swing.JFrame;

public class Register extends JFrame {

    public Register() {
        initComponents();
        setLocationRelativeTo(null);
    }

    private void jBtnLoginMouseClicked(java.awt.event.MouseEvent evt) {
        login uy = new login();
        uy.setVisible(true);
        uy.pack();
        uy.setLocationRelativeTo(null);
        this.dispose();
    }

    private void jBtnRegisterMouseClicked(java.awt.event.MouseEvent evt) {
        RegisPagw wy = new RegisPagw();
        wy.setVisible(true);
        wy.pack();
        wy.setLocationRelativeTo(null);
        this.dispose();
    }

    private void AdminMouseClicked(java.awt.event.MouseEvent evt) {
        AdminPage er = new AdminPage();
        er.setVisible(true);
        er.pack();
        er.setLocationRelativeTo(null);
        this.dispose();
    }
}
```

```

package kamusia;

import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Connection;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JFrame;

public class RegisPagw extends javax.swing.JFrame {
    public RegisPagw() {
        initComponents();
        setLocationRelativeTo(null);
    }

    Connection con;
    PreparedStatement pst;
    ResultSet rs;

    public void Connect() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            try {
                con = DriverManager.getConnection("jdbc:mysql://localhost/atha", "root", "");
            }
            catch (SQLException ex) {
                Logger.getLogger(RegisPagw.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        catch (ClassNotFoundException ex) {
            Logger.getLogger(RegisPagw.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    private void jBtnRegisActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost/atha", "root", "");
            String id = jTxtID.getText();
            String name = jTxtName.getText();
            SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");

```

```

String birth = dateFormat.format(jDateBirth.getDate());
String address = jTxtAddress.getText();
String phone = jTxtPhone.getText();
String password = String.valueOf(jPasswordField2.getPassword());

    pst = con.prepareStatement("INSERT INTO `daftar`(`id`, `name`, `birth`, `address`,
`phone`, `password`) VALUES (?,?,?,?,?,?)");
    pst.setString(1, id);
    pst.setString(2, name);
    pst.setString(3, birth);
    pst.setString(4, address);
    pst.setString(5, phone);
    pst.setString(6, password);

    int k = pst.executeUpdate();
    if (k == 1) {
        JOptionPane.showMessageDialog(null, "Successfully Added");
    }
    else {
        JOptionPane.showMessageDialog(null, "Failed Proccess");
    }
}
catch (SQLException ex) {
    Logger.getLogger(RegisPagw.class.getName()).log(Level.SEVERE, null, ex);
}
catch (ClassNotFoundException ex) {
    Logger.getLogger(RegisPagw.class.getName()).log(Level.SEVERE, null, ex);
}

login sr = new login();
sr.setVisible(true);
sr.pack();
sr.setLocationRelativeTo(null);
this.dispose();

private void jLblInfoMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    login sf = new login();
    sf.setVisible(true);
    sf.pack();
    sf.setLocationRelativeTo(null);
    this.dispose();
}

```

```

package kamusia;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

public class login extends javax.swing.JFrame {
    public login() {
        initComponents();
        this.setLocationRelativeTo(null);
        Connect();
    }

    Connection con;
    PreparedStatement pst;
    ResultSet rs;

    public void Connect(){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            try{
                con = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
            }
            catch (SQLException ex){
                Logger.getLogger(login.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        catch (ClassNotFoundException ex){
            Logger.getLogger(login.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    private void jBtnLoginActionPerformed(java.awt.event.ActionEvent evt) {
        try{
            String id = jTxtID.getText();
            String password = String.valueOf(jPssField.getPassword());
            pst = con.prepareStatement("select * from daftar WHERE id=? AND password=?");
            pst.setString(1, id);
            pst.setString(2, password);
        }
    }
}

```

```

rs = pst.executeQuery();

if (rs.next() == true){
    Dictionary mh = new Dictionary();
    mh.setVisible(true);
    mh.pack();
    mh.setLocationRelativeTo(null);
    mh.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.dispose();
}
else { JOptionPane.showMessageDialog(null, "ID or Password is incorrect");
jPssField.requestFocus();}

}
catch(SQLException ex){
    Logger.getLogger(login.class.getName()).log(Level.SEVERE, null, ex);
}
}

private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {
    RegisPagw sr = new RegisPagw();
    sr.setVisible(true);
    sr.pack();
    sr.setLocale(null);
    this.dispose();
}

```

```

package kamusia;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;

public class Dictionary extends javax.swing.JFrame {
    public Dictionary(){
        initComponents();
        this.setLocationRelativeTo(null);
        //connect();

    }

    Connection conn;
    PreparedStatement stmt;
    ResultSet rs;

    public void Connect (){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            try{
                conn = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
            }
            catch (SQLException ex){
                Logger.getLogger(Dictionary.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        catch (ClassNotFoundException ex){
            Logger.getLogger(Dictionary.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    private void jBtnMagDictActionPerformed(java.awt.event.ActionEvent evt) {
        String keyy = jTxtSearchDict.getText();
    }

```



```

try{

    Class.forName("com.mysql.cj.jdbc.Driver");
    conn = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
    stmt = conn.prepareStatement("select * from masuk WHERE keyword= ?");
    stmt.setString(1, keyy);
    rs = stmt.executeQuery();

    if (rs.next()){
        String key = rs.getString("keyword");
        String spell = rs.getString("spell");
        String meaning = rs.getString("meaning");
        jTextAreaDict.append("Keyword: "+key+"\n"+"Spell: "+spell+"\n"+"Meaning: "+meaning);
    }
    else{

        JOptionPane.showMessageDialog(null, "Data did not found");
        jTextAreaDict.setText("");
        jTxtSearchDict.setText("");
    }

}
catch (SQLException ex) {
    ex.printStackTrace();
}
catch (ClassNotFoundException ex){
}
}

```

```

private void jBtnMagInstActionPerformed(java.awt.event.ActionEvent evt) {
    String keyy = jTxtSearchInst.getText();
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        conn = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
        stmt = conn.prepareStatement("select * from met WHERE keyword= ?");
        stmt.setString(1, keyy);
        rs = stmt.executeQuery();
        if (rs.next()){
            String key = rs.getString("keyword");
            String spell = rs.getString("part");
            String meaning = rs.getString("step");
            jTextAreaDict.append("Keyword: "+key+"\n"+"Part: "+spell+"\n"+"Step: "+meaning);
        }
        else{
            JOptionPane.showMessageDialog(null, "Data did not found");
        }
    }
}

```

```

        jTxtAreaInst.setText("");
        jTxtSearchInst.setText("");
    }
}
catch (SQLException ex) {
    ex.printStackTrace();
}
catch (ClassNotFoundException ex){
}
}

```

```

private void JBtnResetDictActionPerformed(java.awt.event.ActionEvent evt) {
    jTxtAreaDict.setText("");
    jTxtSearchDict.setText("");
}

```

```

private void JBtnResetInstMouseClicked(java.awt.event.MouseEvent evt) {
    jTxtAreaInst.setText("");
    jTxtSearchInst.setText("");
}

```

```

private void jTxtSearchDictMouseClicked(java.awt.event.MouseEvent evt) {
    jTxtAreaDict.setText("");
    jTxtSearchDict.setText(""); // TODO add your handling code here:
}

```

```

private void jTxtSearchInstMouseClicked(java.awt.event.MouseEvent evt) {
    jTxtAreaInst.setText("");
    jTxtSearchInst.setText(""); // TODO add your handling code here:
}

```

```
package kamusia;

import javax.swing.JFrame;
import javax.swing.JOptionPane;

public class AdminPage extends javax.swing.JFrame {
    public AdminPage() {
        initComponents();
    }

    private void jBtnLoginActionPerformed(java.awt.event.ActionEvent evt) {
        if (jLblUser.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null, "Please insert your username");
            jLblUser.requestFocus();
        }
        else if (jPssField.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null, "Please insert your password");
            jPssField.requestFocus();
        }

        else if (jTxtID.getText().contains("admin") && jPssField.getText().contains("bacbeat"))
        {
            new AdminStep().show();
            this.dispose();
        }

        else {
            JOptionPane.showMessageDialog(null, "Username and Password incorrect!");
        }
    }
}
```

```

package kamusia;

import javax.swing.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.Driver;
import java.sql.ResultSet;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import com.mysql.cj.jdbc.result.ResultSetMetaData;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Vector;
import javax.swing.JFrame;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableRowSorter;

public class AdminStep extends javax.swing.JFrame {
    public AdminStep() {
        initComponents();
        setLocationRelativeTo(null);
    }

    Connection con1;
    PreparedStatement insert;
    ResultSet rs;

    public void Connect () {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            try {
                con1 = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
            }
            catch (SQLException ex) {
                Logger.getLogger(AdminStep.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        catch (ClassNotFoundException ex) {
            Logger.getLogger(AdminStep.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

```

private void table_update(){
    int CC;
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        con1 = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
        insert = con1.prepareStatement("SELECT * FROM masuk");
        ResultSet Rs = insert.executeQuery();
        ResultSetMetaData RSMD = (ResultSetMetaData) Rs.getMetaData();
        CC = RSMD.getColumnCount();
        DefaultTableModel DFT = (DefaultTableModel) jTable3Dict.getModel();
        DFT.setRowCount(0);

        while (Rs.next()) {
            Vector v2 = new Vector();
            for (int ii = 1; ii <= CC; ii++) {
                v2.add(Rs.getString("id"));
                v2.add(Rs.getString("keyword"));
                v2.add(Rs.getString("spell"));
                v2.add(Rs.getString("meaning"));
            }
            DFT.addRow(v2);
        }
    }
    catch (Exception e) {
    }

    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        con1 = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
        insert = con1.prepareStatement("SELECT * FROM met");
        ResultSet Rs = insert.executeQuery();
        ResultSetMetaData RSMD = (ResultSetMetaData) Rs.getMetaData();
        CC = RSMD.getColumnCount();
        DefaultTableModel DFT = (DefaultTableModel) jTable2Inst.getModel();
        DFT.setRowCount(0);

        while (Rs.next()) {
            Vector v2 = new Vector();
            for (int ii = 1; ii <= CC; ii++) {
                v2.add(Rs.getString("id"));
                v2.add(Rs.getString("keyword"));
                v2.add(Rs.getString("part"));
                v2.add(Rs.getString("step"));
            }
            DFT.addRow(v2);
        }
    }
}

```

```

    }
} catch (Exception e) {
}
}

```

```

private void jBtnAddInstActionPerformed(java.awt.event.ActionEvent evt) {
    String keyword = jTxtKey.getText();
    String part = jTxtPart.getText();
    String step = jTxtStep.getText();
    DefaultTableModel model = (DefaultTableModel) jTable2Inst.getModel();
    int selectedIndex = jTable2Inst.getSelectedRow();
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        con1 = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
        insert = con1.prepareStatement("insert into met (keyword,part,step)values(?,?,?)");
        insert.setString(1, keyword);
        insert.setString(2, part);
        insert.setString(3, step);
        insert.executeUpdate();
        JOptionPane.showMessageDialog(this, "Successfully Added");
        jTxtKey.setText("");
        jTxtPart.setText("");
        jTxtStep.setText("");
        table_update();
    }
    catch (ClassNotFoundException ex){
        Logger.getLogger(AdminStep.class.getName()).log(Level.SEVERE,null,ex);
    }
    catch(SQLException ex){
        Logger.getLogger(AdminStep.class.getName()).log(Level.SEVERE,null,ex);
    }
}

```

```

private void jBtnDeleteInstActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = (DefaultTableModel) jTable2Inst.getModel();
    int selectedIndex = jTable2Inst.getSelectedRow();
    try {
        int id = Integer.parseInt(model.getValueAt(selectedIndex, 0).toString());
        int dialogResult = JOptionPane.showConfirmDialog (null, "Do you want to delete the
instruction?", "Warning", JOptionPane.YES_NO_OPTION);
        if(dialogResult == JOptionPane.YES_OPTION){
            Class.forName("com.mysql.cj.jdbc.Driver");
            con1 = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
            insert = con1.prepareStatement("delete from met where id= ?");
            insert.setInt(1,id);

```

```

        insert.executeUpdate();
        JOptionPane.showMessageDialog(this, "Instruction Deleted");
        jTxtKey.setText("");
        jTxtPart.setText("");
        jTxtStep.setText("");
        table_update();
    }
} catch (ClassNotFoundException ex) {
}
catch (SQLException ex){
}
}
}

```

```

private void jBtnEditInstMouseClicked(java.awt.event.MouseEvent evt) {
    DefaultTableModel model = (DefaultTableModel) jTable2Inst.getModel();
    int selectedIndex = jTable2Inst.getSelectedRow();
    try {
        int nid = Integer.parseInt(model.getValueAt(selectedIndex, 0).toString());
        String nkeyword = jTxtKey.getText();
        String npart = jTxtPart.getText();
        String nstep = jTxtStep.getText();
        Class.forName("com.mysql.cj.jdbc.Driver");
        con1 = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
        insert = con1.prepareStatement("update met set keyword= ?, part= ?, step= ? where id=
?");
        insert.setString(1, nkeyword);
        insert.setString(2, npart);
        insert.setString(3, nstep);
        insert.setInt(4, nid);
        insert.executeUpdate();
        JOptionPane.showMessageDialog(this, "Dictionary Updated");
        jTxtKey.setText("");
        jTxtPart.setText("");
        jTxtStep.setText("");
        table_update();
    } catch (ClassNotFoundException ex){
    }
    catch (SQLException ex){
    }
}
}

```

```

private void jTable2InstMouseClicked(java.awt.event.MouseEvent evt) {
    DefaultTableModel model = (DefaultTableModel) jTable2Inst.getModel();
    int selectedIndex = jTable2Inst.getSelectedRow();
    jTxtKey.setText(model.getValueAt(selectedIndex, 1).toString());
}

```

```

jTxtPart.setText(model.getValueAt(selectedIndex, 2).toString());
jTxtStep.setText(model.getValueAt(selectedIndex, 3).toString());
}

```

```

private void jBtnSortInsMouseClicked(java.awt.event.MouseEvent evt) {
    DefaultTableModel model = (DefaultTableModel) jTable2Inst.getModel();
    int selectedIndex = jTable2Inst.getSelectedRow();
    if(selectedIndex >=0){
        jTxtKey.setText(model.getValueAt(selectedIndex, 1).toString());
        jTxtPart.setText(model.getValueAt(selectedIndex, 2).toString());
        jTxtStep.setText(model.getValueAt(selectedIndex, 3).toString());
    }
    jTable2Inst.setModel(model);
    TableRowSorter<DefaultTableModel> sorter = new
TableRowSorter<DefaultTableModel>(model);
    jTable2Inst.setRowSorter(sorter);
    List<RowSorter.SortKey> sortKeys = new ArrayList<>(0);
    sortKeys.add(new RowSorter.SortKey(1, SortOrder.ASCENDING));
    sorter.setSortKeys(sortKeys);
    sorter.sort();
    List<RowSorter.SortKey> keys = (List<RowSorter.SortKey>) sorter.getSortKeys();
    if (!keys.isEmpty()) {
        RowSorter.SortKey key = keys.get(0);
        int columnIndex = key.getColumn();
        if (key.getSortOrder() == SortOrder.ASCENDING) {
            if (jBtnSortIns.isSelected()) {
                jBtnAddInst.setEnabled(true);
                jBtnEditInst.setEnabled(true);
                jBtnDeleteInst.setEnabled(true);
            } else
            {
                jBtnAddInst.setEnabled(false);
                jBtnEditInst.setEnabled(false);
                jBtnDeleteInst.setEnabled(false);
                if (key.getSortOrder() == SortOrder.UNSORTED){
                    jTable2Inst.getRowSorter().setSortKeys(Collections.emptyList());
                }
            }
        }
    }
}
}
}

```

```

private void jBtnResetInstActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = (DefaultTableModel) jTable2Inst.getModel();
    int rowCount = model.getRowCount();
    for (int i= 0; i < rowCount; i++){

```



```

        model.setValueAt("", i, 0);
        model.setValueAt("", i, 1);
        model.setValueAt("", i, 2);
        model.setValueAt("", i, 3);
    }

    int CC;
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        con1 = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
        insert = con1.prepareStatement("SELECT * FROM met");
        ResultSet Rs = insert.executeQuery();

        ResultSetMetaData RSMD = (ResultSetMetaData) Rs.getMetaData();
        CC = RSMD.getColumnCount();
        DefaultTableModel DFT = (DefaultTableModel) jTable2Inst.getModel();
        DFT.setRowCount(0);
        while (Rs.next()) {
            Vector v2 = new Vector();
            for (int ii = 1; ii <= CC; ii++) {
                v2.add(Rs.getString("id"));
                v2.add(Rs.getString("keyword"));
                v2.add(Rs.getString("part"));
                v2.add(Rs.getString("step"));
            }
            DFT.addRow(v2);
        }
    } catch (Exception e) {
    }
    jTxtKey.setText("");
    jTxtPart.setText("");
    jTxtStep.setText("");
    table_update();
}

```

```

private void jBtnAddDictActionPerformed(java.awt.event.ActionEvent evt) {
    String keyword = jTxtKeyDic.getText();
    String spell = jTxtSpellDic.getText();
    String meaning = jTxtMeaningDic.getText();
    DefaultTableModel model = (DefaultTableModel) jTable3Dict.getModel();
    int selectedIndex1 = jTable3Dict.getSelectedRow();
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        con1 = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
    }
}

```

```

        insert = con1.prepareStatement("insert into masuk
(keyword,spell,meaning)values(?,?,?)");
        insert.setString(1, keyword);
        insert.setString(2, spell);
        insert.setString(3, meaning);
        insert.executeUpdate();
        JOptionPane.showMessageDialog(this, "Succesfully Added");
        jTxtKey.setText("");
        jTxtSpellDic.setText("");
        jTxtMeaningDic.setText("");
        table_update();
    }
    catch (ClassNotFoundException ex){
        Logger.getLogger(AdminStep.class.getName()).log(Level.SEVERE,null,ex);
    }
    catch (SQLException ex){
        Logger.getLogger(AdminStep.class.getName()).log(Level.SEVERE,null,ex);
    }
}

private void jBtnEditDictActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = (DefaultTableModel) jTable3Dict.getModel();
    int selectedIndex1 = jTable3Dict.getSelectedRow();
    try {
        int nid = Integer.parseInt(model.getValueAt(selectedIndex1, 0).toString());
        String nkeyword = jTxtKeyDic.getText();
        String nspell = jTxtSpellDic.getText();
        String nmeaning = jTxtMeaningDic.getText();
        Class.forName("com.mysql.cj.jdbc.Driver");
        con1 = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
        insert = con1.prepareStatement("update masuk set keyword= ?, spell= ?, meaning= ?
where id= ?");
        insert.setString(1, nkeyword);
        insert.setString(2, nspell);
        insert.setString(3, nmeaning);
        insert.setInt(4, nid);
        insert.executeUpdate();
        JOptionPane.showMessageDialog(this, "Dictionary Updated");
        jTxtKeyDic.setText("");
        jTxtSpellDic.setText("");
        jTxtMeaningDic.setText("");
        table_update();
    } catch (ClassNotFoundException ex){
    }
    catch (SQLException ex){

```

```
}  
}
```

```
private void jBtnDeleteDictActionPerformed(java.awt.event.ActionEvent evt) {  
    DefaultTableModel model = (DefaultTableModel) jTable3Dict.getModel();  
    int selectedIndex1= jTable3Dict.getSelectedRow();  
    try {  
        int id = Integer.parseInt(model.getValueAt(selectedIndex1, 0).toString());  
        int dialogResult = JOptionPane.showConfirmDialog (null, "Do you want to delete the  
dictionary?", "Warning", JOptionPane.YES_NO_OPTION);  
        if(dialogResult == JOptionPane.YES_OPTION){  
            Class.forName("com.mysql.cj.jdbc.Driver");  
            con1 = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");  
            insert = con1.prepareStatement("delete from masuk where id= ?");  
            insert.setInt(1,id);  
            insert.executeUpdate();  
            JOptionPane.showMessageDialog(this, "Dictionary Deleted");  
            jTxtKeyDic.setText("");  
            jTxtSpellDic.setText("");  
            jTxtMeaningDic.setText("");  
            table_update();  
        }  
    } catch (ClassNotFoundException ex) {  
    }  
    catch (SQLException ex){  
    }  
}
```

```
private void jTable3DictMouseClicked(java.awt.event.MouseEvent evt) {  
    DefaultTableModel model = (DefaultTableModel) jTable3Dict.getModel();  
    int selectedIndex = jTable3Dict.getSelectedRow();  
    jTxtKeyDic.setText(model.getValueAt(selectedIndex, 1).toString());  
    jTxtSpellDic.setText(model.getValueAt(selectedIndex, 2).toString());  
    jTxtMeaningDic.setText(model.getValueAt(selectedIndex, 3).toString());  
}
```

```
private void jBtnSortDictMouseClicked(java.awt.event.MouseEvent evt) {  
    DefaultTableModel model = (DefaultTableModel) jTable3Dict.getModel();  
    int selectedIndex = jTable3Dict.getSelectedRow();  
    if(selectedIndex >=0){  
        jTxtKeyDic.setText(model.getValueAt(selectedIndex, 1).toString());  
        jTxtSpellDic.setText(model.getValueAt(selectedIndex, 2).toString());  
        jTxtMeaningDic.setText(model.getValueAt(selectedIndex, 3).toString());  
    }  
}
```

```

jTable3Dict.setModel(model);
TableRowSorter<DefaultTableModel> sorter = new
TableRowSorter<DefaultTableModel>(model);
jTable3Dict.setRowSorter(sorter);
List<RowSorter.SortKey> sortKeys = new ArrayList<>(0);
sortKeys.add(new RowSorter.SortKey(1, SortOrder.ASCENDING));
sorter.setSortKeys(sortKeys);
sorter.sort();
List<RowSorter.SortKey> keys = (List<RowSorter.SortKey>) sorter.getSortKeys();
if (!keys.isEmpty()) {
    RowSorter.SortKey key = keys.get(0);
    int columnIndex = key.getColumn();
    if (key.getSortOrder() == SortOrder.ASCENDING) {
        if (jBtnSortDict.isSelected()){
            jBtnAddDict.setEnabled(true);
            jBtnEditDict.setEnabled(true);
            jBtnDeleteDict.setEnabled(true);
        }
        else {
            jBtnAddDict.setEnabled(false);
            jBtnEditDict.setEnabled(false);
            jBtnDeleteDict.setEnabled(false);
            if (key.getSortOrder() == SortOrder.UNSORTED){
                jTable3Dict.getRowSorter().setSortKeys(Collections.emptyList());
            }
        }
    }
}
}
}
}

```

```

private void jBtnResetDictActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel model = (DefaultTableModel) jTable3Dict.getModel();
    int rowCount = model.getRowCount();
    for (int i= 0; i < rowCount; i++){
        model.setValueAt("", i, 0);
        model.setValueAt("", i, 1);
        model.setValueAt("", i, 2);
        model.setValueAt("", i, 3);
    }
    int CC;
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        con1 = DriverManager.getConnection("jdbc:mysql://localhost/atha","root","");
        insert = con1.prepareStatement("SELECT * FROM masuk");
        ResultSet Rs = insert.executeQuery();
    }
}

```

```

ResultSetMetaData RSMD = (ResultSetMetaData) Rs.getMetaData();
CC = RSMD.getColumnCount();
DefaultTableModel DFT = (DefaultTableModel) jTable3Dict.getModel();
DFT.setRowCount(0);
while (Rs.next()) {
    Vector v2 = new Vector();
    for (int ii = 1; ii <= CC; ii++) {
        v2.add(Rs.getString("id"));
        v2.add(Rs.getString("keyword"));
        v2.add(Rs.getString("spell"));
        v2.add(Rs.getString("meaning"));
    }
    DFT.addRow(v2);
}
} catch (Exception e) {
}
jTxtKeyDic.setText("");
jTxtSpellDic.setText("");
jTxtMeaningDic.setText("");
table_update();
}

```

