LAPORAN TUGAS AKHIR
PEMROGRAMAN BERORIENTASI OBJEK
CHATBOT TELEGRAM DENGAN PEMROGRAMAN BERBASIS JAVA

Disusun Oleh:

NIM : A11.2021.13441

NAMA : Mohammad Wahyu Bagus Dwi Satya

KELOMPOK : A11.442U

FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO SEMARANG

2022/2023

DAFTAR ISI

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar belakang

Dalam era digital yang semakin berkembang pesat seperti saat ini, pemrograman menjadi salah satu keterampilan yang sangat berharga. Bahasa pemrograman Java, sebagai salah satu bahasa pemrograman yang populer dan luas digunakan, menjadi fokus utama banyak individu yang ingin mempelajari dunia pemrograman. Namun, memahami dan menguasai bahasa pemrograman Java tidak selalu mudah bagi pemula.

Pemahaman konsep dasar dan sintaks dalam bahasa pemrograman Java merupakan langkah awal yang penting dalam proses belajar. Selain itu, pemecahan masalah dan logika pemrograman juga menjadi tantangan yang harus dihadapi oleh pemula. Menghadapi kesulitan-kesulitan tersebut, seringkali pemula membutuhkan bantuan dan panduan yang tepat untuk dapat memahami dan mengatasi hambatan yang ada.

Dalam lingkungan pembelajaran, penggunaan chatbot sebagai alat bantu dapat memberikan solusi yang efektif dan inovatif. Chatbot dapat memberikan respons instan, memberikan penjelasan, serta memberikan contoh-contoh yang relevan terkait konsep-konsep dalam pemrograman Java. Keunggulan chatbot sebagai alat pembelajaran terletak pada interaktivitasnya yang dapat menyesuaikan diri dengan kebutuhan pengguna, memberikan bantuan yang personal, serta memberikan umpan balik secara langsung.

Telegram, sebagai salah satu platform komunikasi yang populer, menyediakan fitur bot yang memungkinkan pengembangan chatbot yang dapat diintegrasikan dengan aplikasi Telegram. Dengan menggabungkan potensi chatbot dan platform Telegram, dapat diciptakan sebuah chatbot berorientasi objek yang mengadopsi tema pembelajaran bahasa pemrograman Java.

Tujuan utama dari laporan tugas akhir ini adalah untuk merancang dan mengembangkan chatbot Telegram berorientasi objek yang bertema pembelajaran tentang bahasa pemrograman Java. Chatbot ini diharapkan dapat memberikan bantuan yang efektif dalam mempelajari bahasa pemrograman Java kepada pengguna, mulai dari penjelasan konsep dasar hingga pemecahan masalah yang lebih kompleks.

Dengan adanya chatbot ini, diharapkan pemula dalam dunia pemrograman Java dapat memperoleh pengalaman belajar yang lebih interaktif, mudah, dan efisien. Selain itu, chatbot ini juga dapat menjadi solusi bagi mereka yang membutuhkan bantuan cepat dan responsif dalam memahami konsep-konsep pemrograman Java.

BAB II

ANALISA DAN DESAIN

2.1 Analisa Kebutuhan

2.1.1 Kebutuhan Pengguna

Sistem yang akan dikembangkan pada chatbot meliputi :

1. Kelola data kata kunci atau keyword.
2. Kelola data history pesan masuk dan keluar.
3. Kelola member yang telah terdaftar di sistem.
4. Mengirim broadcast kepada member.
5. Login untuk masuk dalam sistem.
6. Kelola Dasboard Admin.

2.1.2 Kebutuhan Perangkat Keras

Konfigurasi perangkat keras (hardware) :

a. Laptop atau PC dengan Processor: Intel Core i7-7700HQ
b. RAM: 16 GB
c. SSD: NVMe M.2 512 GB
d. Kartu grafis: NVIDIA GeForce RTX 1050

Dari konfigurasi diatas dapat menggunakan laptop yang mampu beroperasi dalam jangka waktu yang lama, karena digunakan sebagai server untuk menjalankan program chatbot agar selalu online dan dapat menerima sekaligus mengirimkan response kepada member.

2.1.3 Kebutuhan Perangkat Lunak

Digunakan untuk mendukung chatbot secara komputerisasi, antara lain :

1. Bahasa pemrograman

   Menggunakan bahasa pemrograman Java 17 dengan pertimbangan sebagai berikut:

   a. Bahasa pemrograman java mempunyai fasilitas dalam penyediaan fungsi perintah dan kecepatan eksekusi.
   b. Bahasa pemrograman yang diwajibkan dalam mata kuliah Pemrograman Berorientasi Objek sebagai proyek akhir.

2. Text Editor (IDE)

   a. Netbeans 17
   b. Visual Studio Code versi 1.79.2.

3. Database

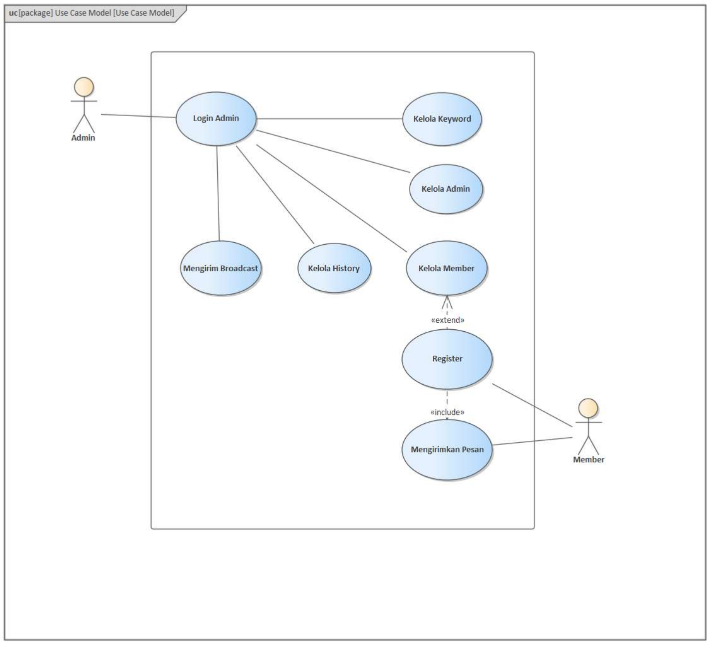
   a. MySQL

4. Software

   Telegram Desktop atau Aplikasi Telegram sebagai software testing dan validasi fitur baru pada chatbot.

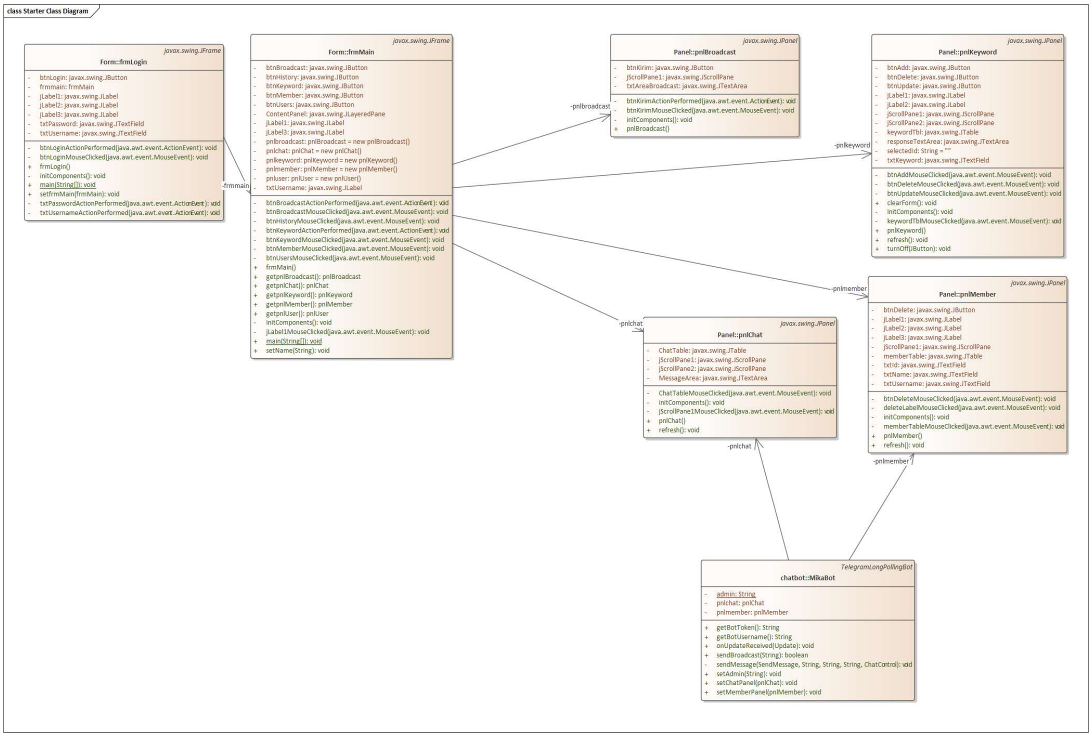
2.2 Perancangan

2.2.1 Use Case Diagram



Gambar 1 Use Case

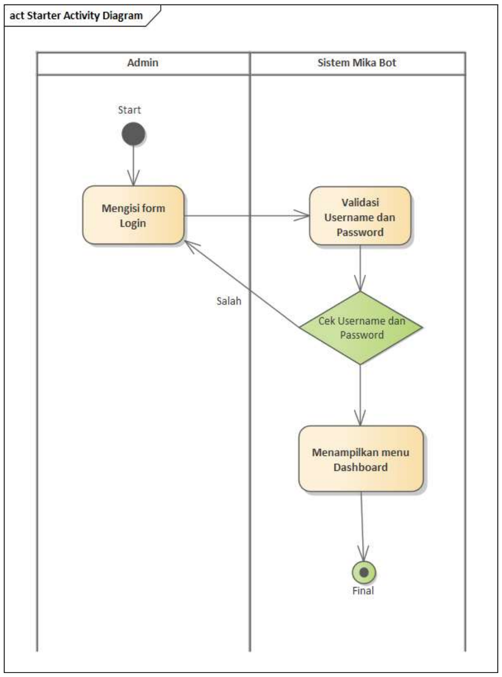
2.2.2 Class Diagram


class Starter Class Diagram
javax.swing.JFrame
Form::frmLogin
- btnLogin: javax.swing.JButton
- frmmain: frmMain
- jLabel1: javax.swing.JLabel
- jLabel2: javax.swing.JLabel
- jLabel3: javax.swing.JLabel
- txtPassword: javax.swing.JTextField
- txtUsername: javax.swing.JTextField
- btnLoginActionPerformed(java.awt.event.ActionEvent): void
- btnLoginMouseClicked(java.awt.event.MouseEvent): void
+ frmLogin()
- initComponents(): void
+ main(String[]): void
+ setfrmMain(frmMain): void
- txtPasswordActionPerformed(java.awt.event.ActionEvent): void
- txtUsernameActionPerformed(java.awt.event.ActionEvent): void
-frmmain
javax.swing.JFrame
Form::frmMain
- btnBroadcast: javax.swing.JButton
- btnHistory: javax.swing.JButton
- btnKeyword: javax.swing.JButton
- btnMember: javax.swing.JButton
- btnUsers: javax.swing.JButton
- ContentPanel: javax.swing.JLayeredPane
- jLabel1: javax.swing.JLabel
- jLabel3: javax.swing.JLabel
- pnlbroadcast: pnlBroadcast = new pnlBroadcast()
- pnlchat: pnlChat = new pnlChat()
- pnlkeyword: pnlKeyword = new pnlKeyword()
- pnlmember: pnlMember = new pnlMember()
- pnluser: pnlUser = new pnlUser()
- txtUsername: javax.swing.JLabel
- btnBroadcastActionPerformed(java.awt.event.ActionEvent): void
- btnBroadcastMouseClicked(java.awt.event.MouseEvent): void
- btnHistoryMouseClicked(java.awt.event.MouseEvent): void
- btnKeywordActionPerformed(java.awt.event.ActionEvent): void
- btnKeywordMouseClicked(java.awt.event.MouseEvent): void
- btnMemberMouseClicked(java.awt.event.MouseEvent): void
- btnUsersMouseClicked(java.awt.event.MouseEvent): void
+ frmMain()
+ getpnlBroadcast(): pnlBroadcast
+ getpnlChat(): pnlChat
+ getpnlKeyword(): pnlKeyword
+ getpnlMember(): pnlMember
+ getpnlUser(): pnlUser
- initComponents(): void
- jLabel1MouseClicked(java.awt.event.MouseEvent): void
+ main(String[]): void
+ setName(String): void
-pnlbroadcast
javax.swing.JPanel
Panel::pnlBroadcast
- btnKirim: javax.swing.JButton
- jScrollPane1: javax.swing.JScrollPane
- txtAreaBroadcast: javax.swing.JTextArea
- btnKirimActionPerformed(java.awt.event.ActionEvent): void
- btnKirimMouseClicked(java.awt.event.MouseEvent): void
- initComponents(): void
+ pnlBroadcast()
-pnlkeyword
javax.swing.JPanel
Panel::pnlKeyword
- btnAdd: javax.swing.JButton
- btnDelete: javax.swing.JButton
- btnUpdate: javax.swing.JButton
- jLabel1: javax.swing.JLabel
- jLabel2: javax.swing.JLabel
- jScrollPane1: javax.swing.JScrollPane
- jScrollPane2: javax.swing.JScrollPane
- keywordTbl: javax.swing.JTable
- responseTextArea: javax.swing.JTextArea
- selectedId: String = ""
- txtKeyword: javax.swing.JTextField
- btnAddMouseClicked(java.awt.event.MouseEvent): void
- btnDeleteMouseClicked(java.awt.event.MouseEvent): void
- btnUpdateMouseClicked(java.awt.event.MouseEvent): void
+ clearForm(): void
- initComponents(): void
- keywordTblMouseClicked(java.awt.event.MouseEvent): void
+ pnlKeyword()
+ refresh(): void
+ turnOff(JButton): void
-pnlmember
javax.swing.JPanel
Panel::pnlMember
- btnDelete: javax.swing.JButton
- jLabel1: javax.swing.JLabel
- jLabel2: javax.swing.JLabel
- jLabel3: javax.swing.JLabel
- jScrollPane1: javax.swing.JScrollPane
- memberTable: javax.swing.JTable
- txtId: javax.swing.JTextField
- txtName: javax.swing.JTextField
- txtUsername: javax.swing.JTextField
- btnDeleteMouseClicked(java.awt.event.MouseEvent): void
- deleteLabelMouseClicked(java.awt.event.MouseEvent): void
- initComponents(): void
- memberTableMouseClicked(java.awt.event.MouseEvent): void
+ pnlMember()
+ refresh(): void
-pnlchat
javax.swing.JPanel
Panel::pnlChat
- ChatTable: javax.swing.JTable
- jScrollPane1: javax.swing.JScrollPane
- jScrollPane2: javax.swing.JScrollPane
- MessageArea: javax.swing.JTextArea
- ChatTableMouseClicked(java.awt.event.MouseEvent): void
- initComponents(): void
- jScrollPane1MouseClicked(java.awt.event.MouseEvent): void
+ pnlChat()
+ refresh(): void
-pnlchat
-pnlmember
TelegramLongPollingBot
chatbot::MikaBot
- admin: String
- pnlchat: pnlChat
- pnlmember: pnlMember
+ getBotToken(): String
+ getBotUsername(): String
+ onUpdateReceived(Update): void
+ sendBroadcast(String): boolean
- sendMessage(SendMessage, String, String, String, ChatControl): void
+ setAdmin(String): void
+ setChatPanel(pnlChat): void
+ setMemberPanel(pnlMember): void

Gambar 2 Class Diagram

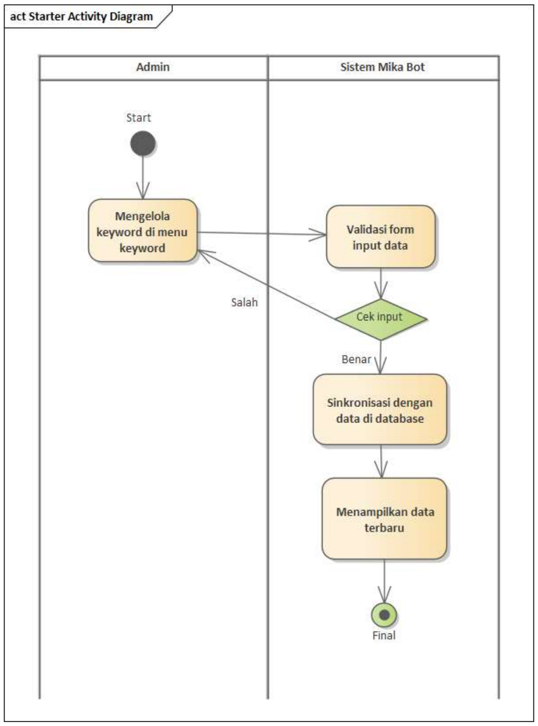
2.2.3 Activity Diagram

2.2.3.1. Activity Diagram Login Admin


act Starter Activity Diagram
Admin
Sistem Mika Bot
Start
Mengisi form Login
Validasi Username dan Password
Salah
Cek Username dan Password
Menampilkan menu Dashboard
Final

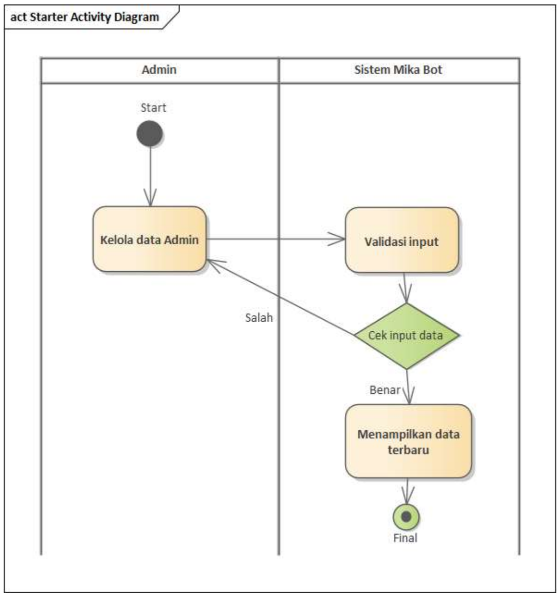
Gambar 3 Activity Diagram Login Admin

2.2.3.2. Activity Diagram Kelola Keyword


act Starter Activity Diagram
Admin
Sistem Mika Bot
Start
Mengelola keyword di menu keyword
Validasi form input data
Salah
Cek input
Benar
Sinkronisasi dengan data di database
Menampilkan data terbaru
Final

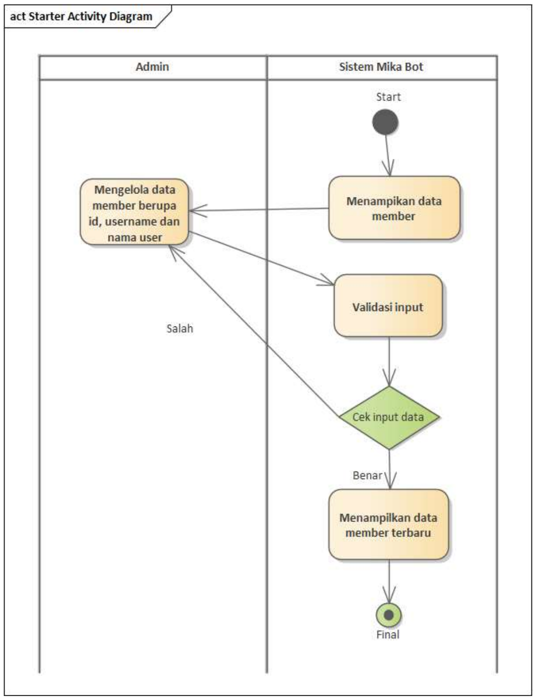
Gambar 4 Activity Diagram Kelola Keyword

2.2.3.3. *Activity Diagram Kelola Admin*



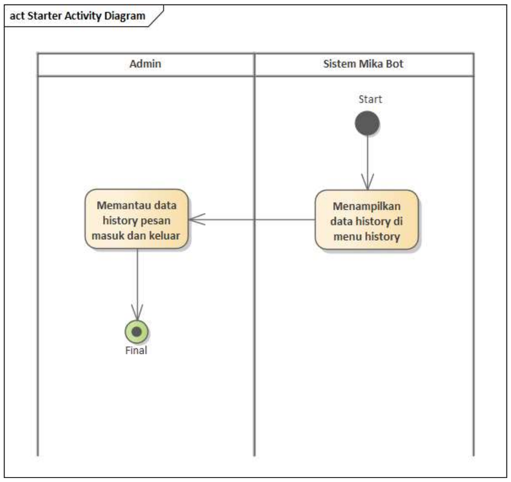
Gambar 5 Activity Diagram Kelola Admin

2.2.3.4. *Activity Diagram Kelola Member*


act Starter Activity Diagram
Admin
Sistem Mika Bot
Start
Menampikan data member
Mengelola data member berupa id, username dan nama user
Validasi input
Salah
Cek input data
Benar
Menampilkan data member terbaru
Final

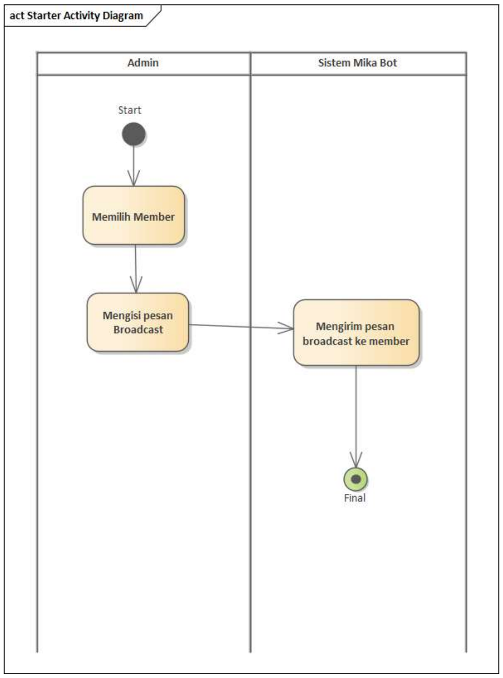
Gambar 6 Activity Diagram Kelola Member

2.2.3.5. Activity Diagram Kelola History


act Starter Activity Diagram
Admin
Sistem Mika Bot
Start
Menampilkan data history di menu history
Memantau data history pesan masuk dan keluar
Final

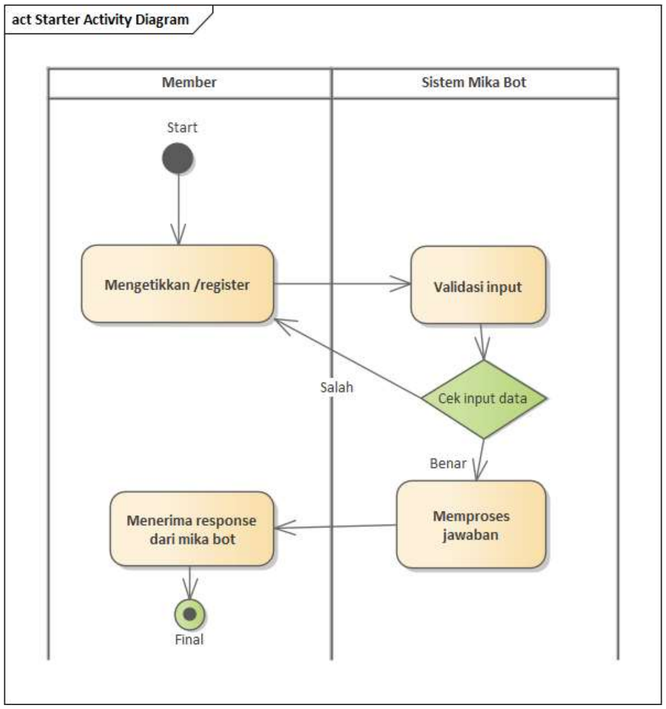
Gambar 7 Activity Diagram Kelola History

2.2.3.6. Activity Diagram Mengirim Broadcast



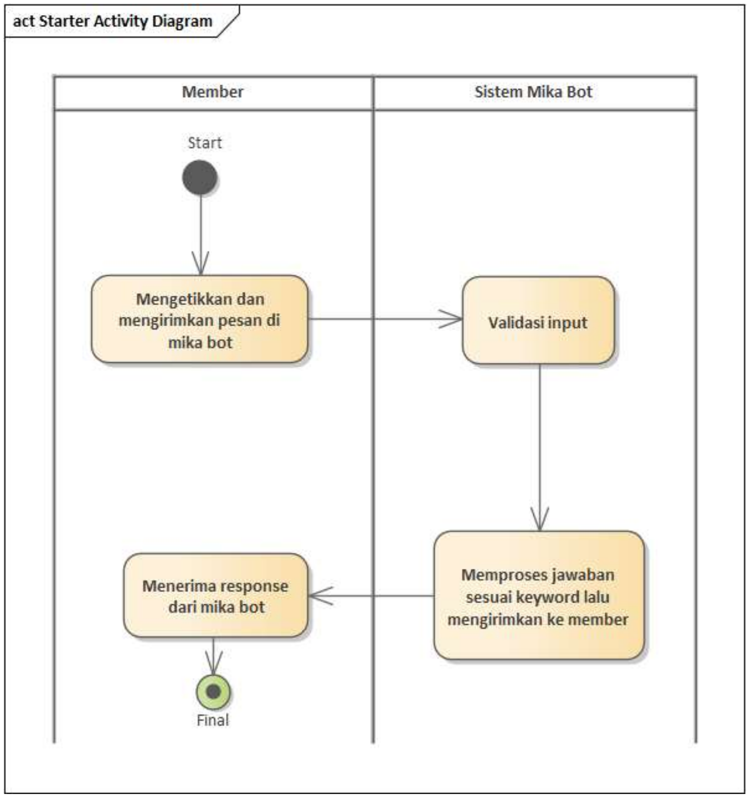
Gambar 8 Activity Diagram Mengirim Broadcast

2.2.3.7. *Activity Diagram Register*



Gambar 9 Activity Diagram Register

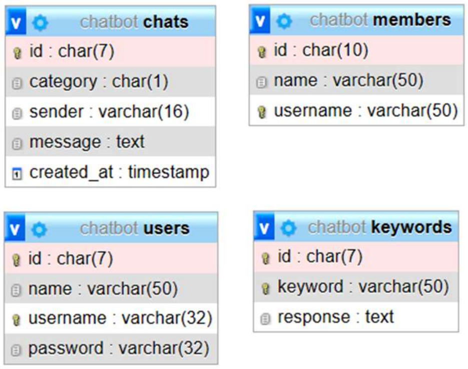
2.2.3.8. *Activity Diagram Mengirimkan Pesan*


act Starter Activity Diagram
Member
Sistem Mika Bot
Start
Mengetikkan dan mengirimkan pesan di mika bot
Validasi input
Memproses jawaban sesuai keyword lalu mengirimkan ke member
Menerima response dari mika bot
Final

Gambar 10 Activity Diagram Mengirimkan Pesan

2.2.4 Desain Database dan Tabel

Desain database bisa dilihat digambar berikut :



chatbot **chats**
- id : char(7)
- category : char(1)
- sender : varchar(16)
- message : text
- created_at : timestamp

chatbot **members**
- id : char(10)
- name : varchar(50)
- username : varchar(50)

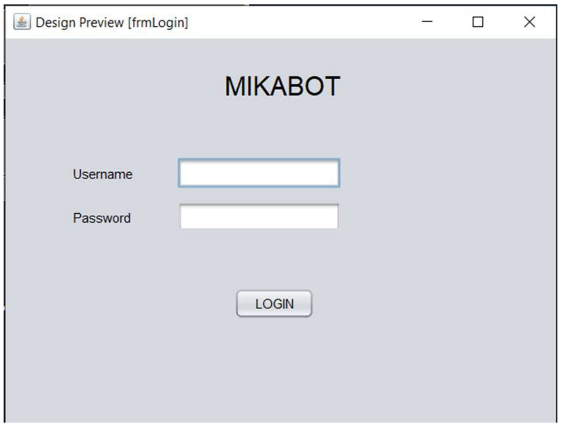
chatbot **users**
- id : char(7)
- name : varchar(50)
- username : varchar(32)
- password : varchar(32)

chatbot **keywords**
- id : char(7)
- keyword : varchar(50)
- response : text

Gambar 11 Desain Database dan Tabel

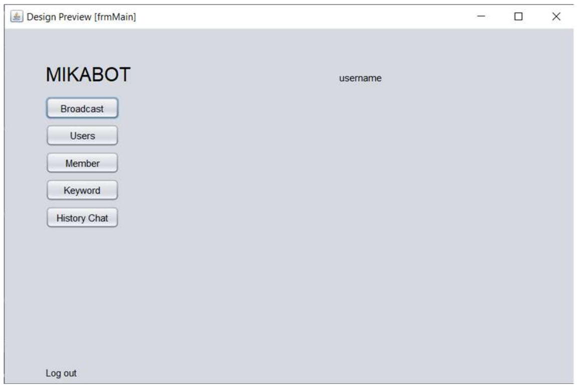
2.2.5 Desain Input dan Output

2.2.5.1. Form Login



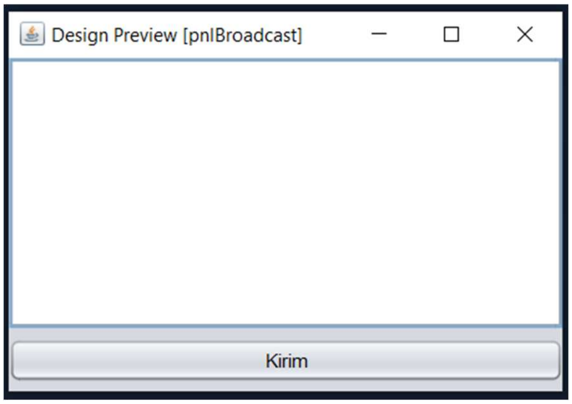
Gambar 12 Form Login

2.2.5.2. Form Main



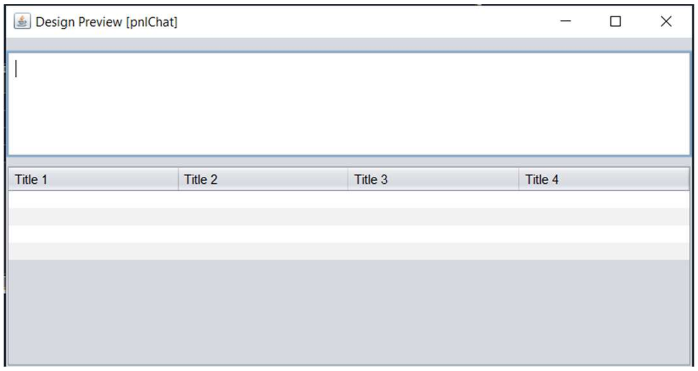
Gambar 13Form Main

2.2.5.3. Panel Broadcast



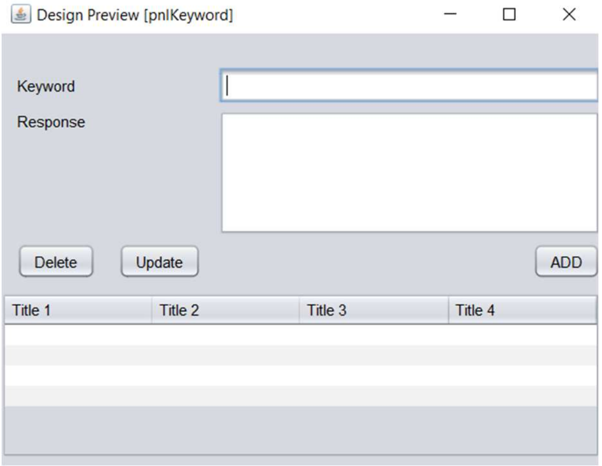
Gambar 14 Panel BroadCast

2.2.5.4. Panel Chat



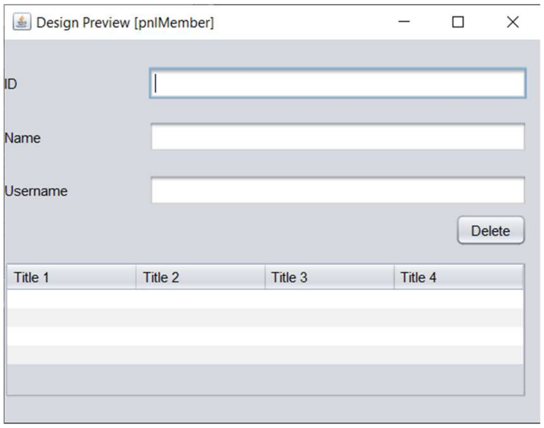
Gambar 15 Panel Chat

2.2.5.5. Panel Keyword



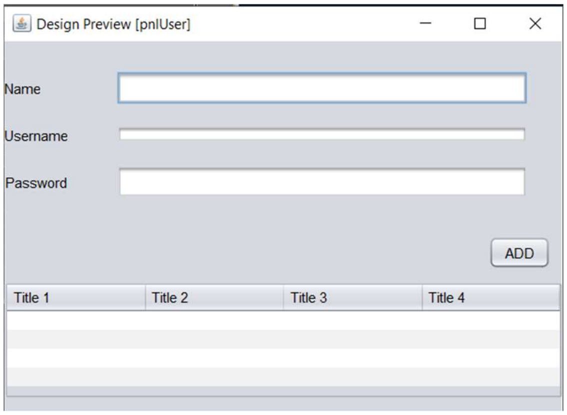
Gambar 16 Panel Keyword

2.2.5.6. *Panel Member*



Gambar 17 Panel Member

2.2.5.7. *Panel User*



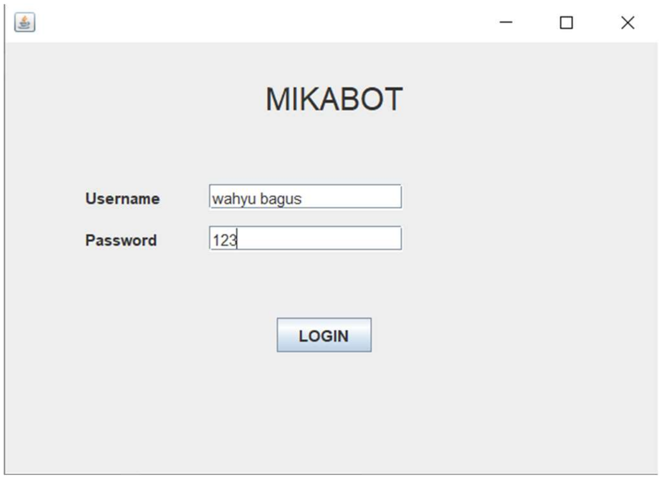
Gambar 18 Panel User

BAB III

IMPLEMENTASI

3.1 Implementasi Antar Muka

3.1.1 Form Login



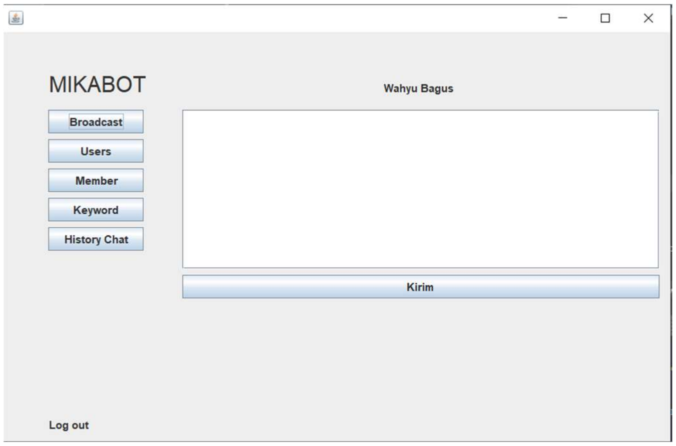
MIKABOT

Username: wahyu bagus

Password: 123

Gambar 19 Form Login

3.1.2 Form Panel Broadcast

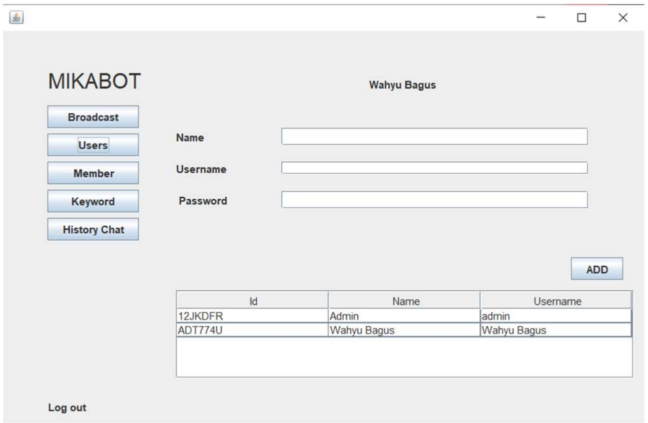


MIKABOT

Wahyu Bagus

Gambar 20 Form Panel Broadcast

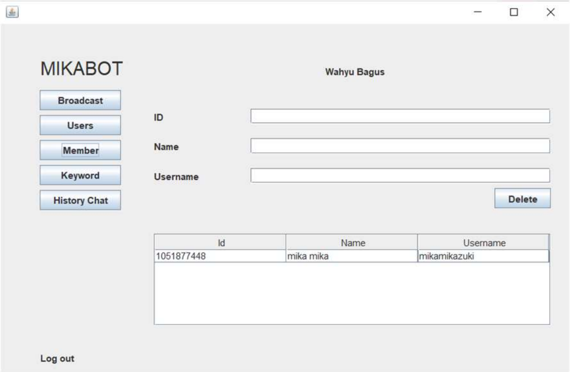
3.1.3 Form Panel Users



Id	Name	Username
12JKDFR	Admin	admin
ADT774U	Wahyu Bagus	Wahyu Bagus

Gambar 21 Form Panel Users

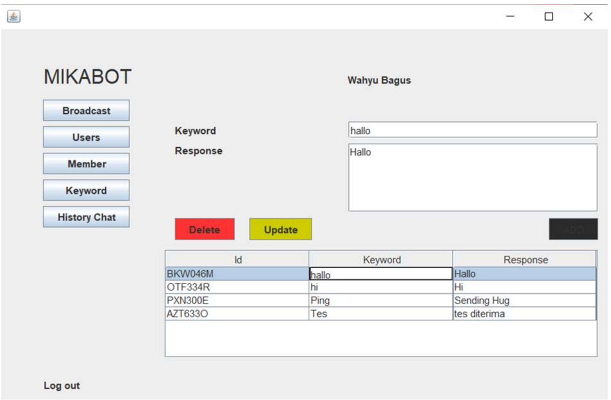
3.1.4 Form Panel Member



Id	Name	Username
1051877448	mika mika	mikamikazuki

Gambar 22 Form Panel Member

3.1.5 Form Panel Keyword



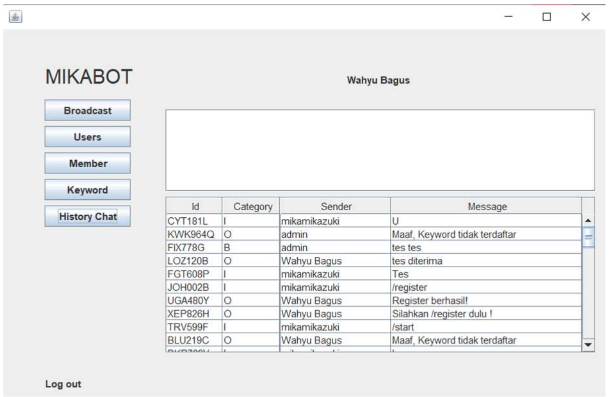
Keyword: hallo

Response: Hallo

Id	Keyword	Response
BKW046M	hallo	Hallo
OTF334R	hi	Hi
PXN300E	Ping	Sending Hug
AZT633O	Tes	tes diterima

Gambar 23 Form Panel Keyword

3.1.6 Form Panel History Chat



Wahyu Bagus

Id	Category	Sender	Message
CYT181L	I	mikamikazuki	U
KWK964Q	O	admin	Maaf, Keyword tidak terdaftar
FIX778G	B	admin	tes tes
LOZ120B	O	Wahyu Bagus	tes diterima
FGT608P	I	mikamikazuki	Tes
JOH002B	I	mikamikazuki	/register
UGA480Y	O	Wahyu Bagus	Register berhasil!
XEP826H	O	Wahyu Bagus	Silahkan /register dulu !
TRV599F	I	mikamikazuki	/start
BLU219C	O	Wahyu Bagus	Maaf, Keyword tidak terdaftar

Gambar 24 Form Panel History Chat

3.2 Implementasi Sistem

Dalam implementasi sistem chatbot Telegram bertema pembelajaran bahasa pemrograman Java, beberapa komponen dan teknologi telah digunakan. Berikut adalah implementasi sistem yang dapat dijelaskan secara umum:

1. Lingkungan Pengembangan: NetBeans IDE
    - NetBeans IDE merupakan lingkungan pengembangan terintegrasi yang digunakan untuk mengembangkan proyek chatbot Telegram.
    - NetBeans menyediakan fitur-fitur yang mendukung pengembangan aplikasi Java, termasuk editor kode, kompilasi, dan manajemen proyek.
2. Platform: Telegram Bot API
    - Telegram Bot API digunakan untuk berkomunikasi dengan Telegram dan membuat chatbot.
    - Melalui Telegram Bot API, chatbot dapat menerima dan mengirim pesan, serta menangani berbagai perintah dan interaksi dengan pengguna.
3. Bahasa Pemrograman: Java
    - Java dipilih sebagai bahasa pemrograman utama dalam mengimplementasikan logika chatbot.
    - Dalam bahasa Java, penulis dapat mengelola alur program, mengelola data, dan mengimplementasikan fungsi-fungsi yang diperlukan dalam chatbot.
4. Libraries dan Frameworks:
    - TelegramBots Library: Digunakan untuk mempermudah pengembangan chatbot dengan Telegram Bot API. Library ini menyediakan kelas-kelas dan metode-metode yang diperlukan untuk berinteraksi dengan API Telegram.
    - NetBeans Plugin: Plugin khusus untuk NetBeans yang dapat membantu pengembangan aplikasi chatbot dengan fitur-fitur yang lebih nyaman dan terintegrasi.
5. Pengembangan Fitur:

   - Penulis mengembangkan fitur-fitur seperti memberikan informasi tentang dasar-dasar pemrograman Java, menjawab pertanyaan seputar Java, memberikan contoh kode program, dan memberikan umpan balik kepada pengguna.
   - Fitur-fitur ini diimplementasikan dalam kode Java menggunakan library TelegramBots dan API Telegram.
6. Uji Coba dan Debugging:
   - Setelah implementasi selesai, chatbot diuji coba untuk memastikan fungsionalitas yang diharapkan berjalan dengan baik.
   - Jika ditemukan kesalahan atau masalah, proses debugging dilakukan untuk mengidentifikasi dan memperbaiki kesalahan tersebut.
7. Penyimpanan Data (Mysql):
   - Jika diperlukan, chatbot dapat menggunakan sistem penyimpanan data (misalnya, database) untuk menyimpan informasi pengguna, riwayat percakapan, atau data lainnya yang relevan.

BAB IV
PENUTUP

4.1 Kesimpulan

Berdasarkan analisis, perancangan dan implementasi pada sistem yang telah dibuat dapat diambil kesimpulan sebagai berikut:

1. Mika Bot dapat melakukan registrasi otomatis saat member melakukan chatting.
2. Mika Bot memiliki fitur seperti kelola admin, kelola kata kunci, kelola member, kelola pesan masuk dan keluar, dan mengirimkan broadcast kepada member yang telah dipilih.

4.2 Saran

Semoga Mika Bot dapat dikembangkan lebih lanjut untuk menambahkan kecerdasan buatan didalammya untuk pengolahan bahasa dan menambahkan Aplication Program Iterface (API) yang bermanfaat bagi pembelajaran di Mika Bot

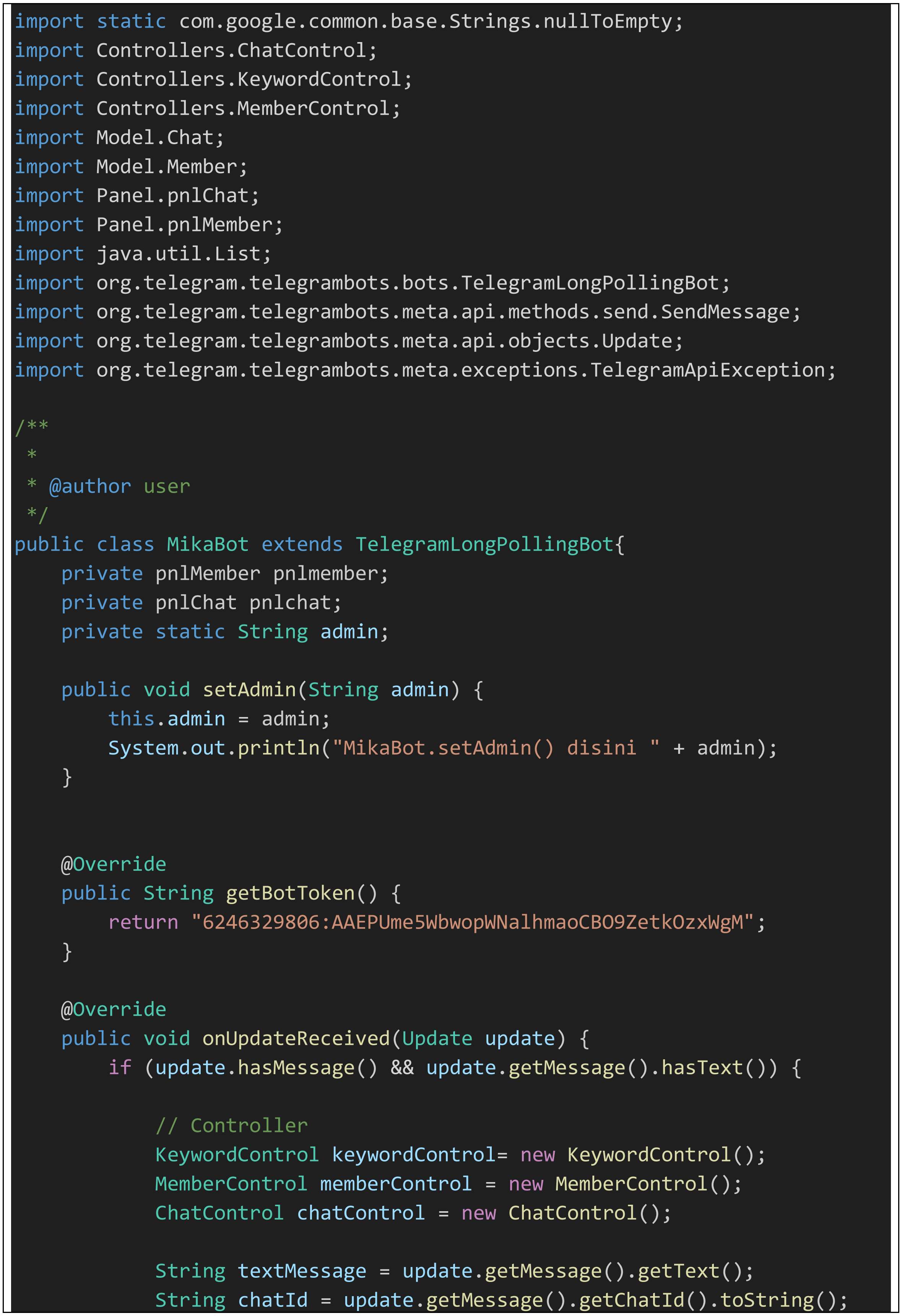
Lampiran Code Program

- Main.java

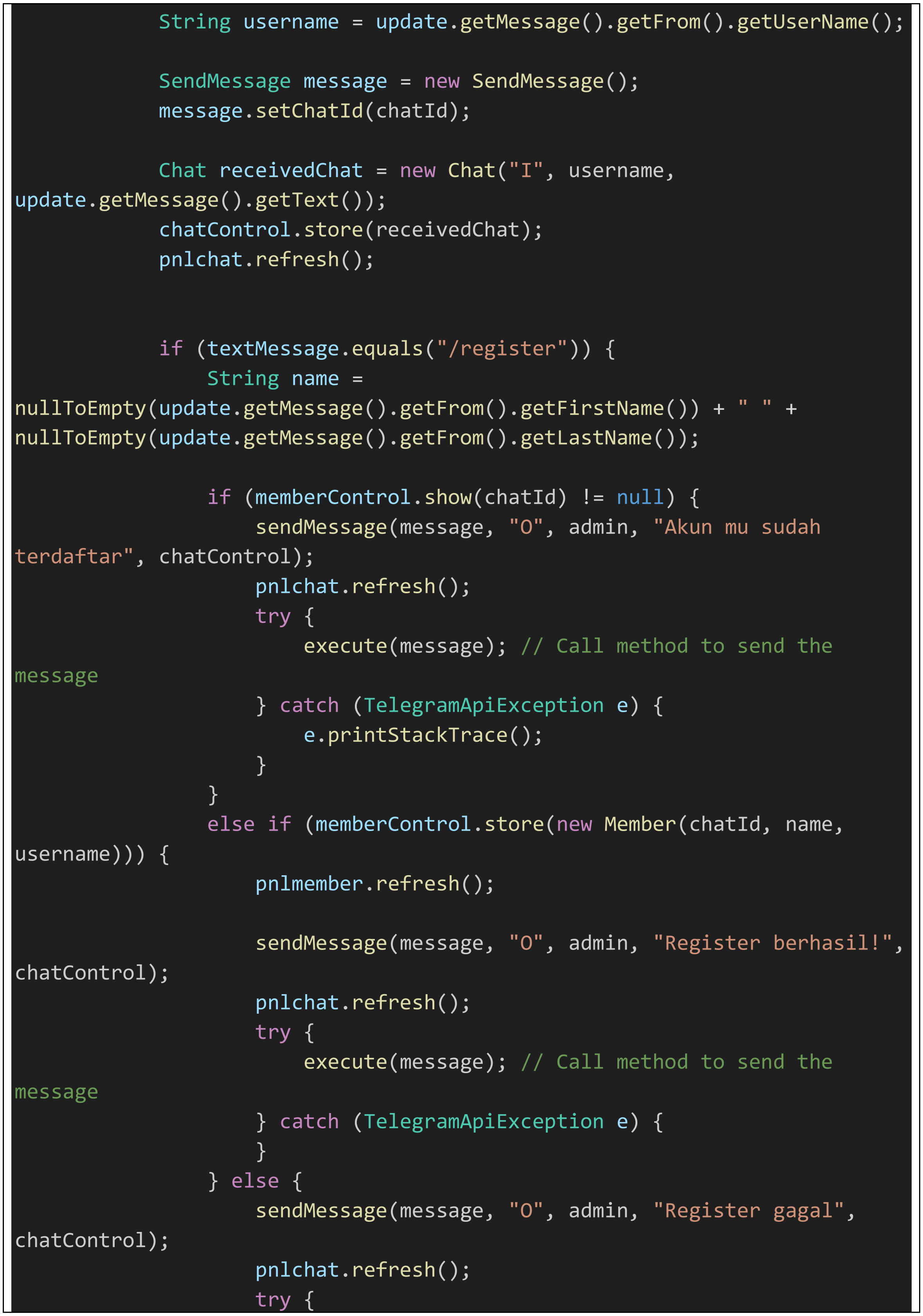
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package com.wahyubagus.project.chatbot;

import Form.frmLogin;
import Form.frmMain;

/**
 *
 * @author user
 */
public class Main {
    public static void main(String[] args) {

        frmMain frmmain = new frmMain();
        frmmain.setVisible(false);
        frmLogin frmlogin = new frmLogin();
        frmlogin.setfrmMain(frmmain);
        frmlogin.setVisible(true);


        System.out.println("Siap dimulai");
    }
}
```

- MikaBot.java

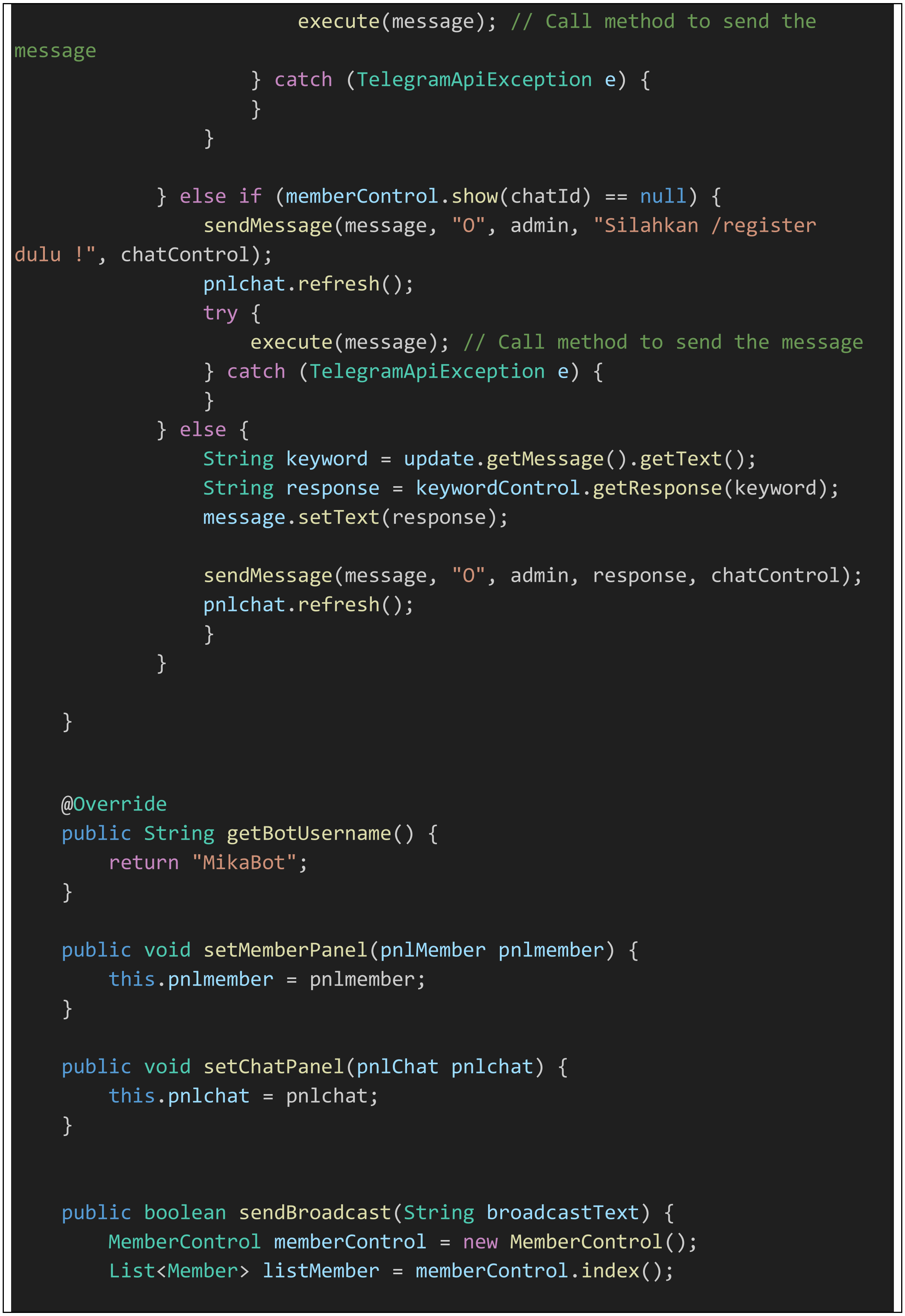
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package com.wahyubagus.project.chatbot;
```



```java
import static com.google.common.base.Strings.nullToEmpty;
import Controllers.ChatControl;
import Controllers.KeywordControl;
import Controllers.MemberControl;
import Model.Chat;
import Model.Member;
import Panel.pnlChat;
import Panel.pnlMember;
import java.util.List;
import org.telegram.telegrambots.bots.TelegramLongPollingBot;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Update;
import org.telegram.telegrambots.meta.exceptions.TelegramApiException;

/**
 *
 * @author user
 */
public class MikaBot extends TelegramLongPollingBot{
    private pnlMember pnlmember;
    private pnlChat pnlchat;
    private static String admin;

    public void setAdmin(String admin) {
        this.admin = admin;
        System.out.println("MikaBot.setAdmin() disini " + admin);
    }


    @Override
    public String getBotToken() {
        return "6246329806:AAEPUme5WbwopWNalhmaoCBO9ZetkOzxWgM";
    }

    @Override
    public void onUpdateReceived(Update update) {
        if (update.hasMessage() && update.getMessage().hasText()) {

            // Controller
            KeywordControl keywordControl= new KeywordControl();
            MemberControl memberControl = new MemberControl();
            ChatControl chatControl = new ChatControl();

            String textMessage = update.getMessage().getText();
            String chatId = update.getMessage().getChatId().toString();
```



```
            String username = update.getMessage().getFrom().getUserName();

            SendMessage message = new SendMessage();
            message.setChatId(chatId);

            Chat receivedChat = new Chat("I", username,
update.getMessage().getText());
            chatControl.store(receivedChat);
            pnlchat.refresh();


            if (textMessage.equals("/register")) {
                String name =
nullToEmpty(update.getMessage().getFrom().getFirstName()) + " " +
nullToEmpty(update.getMessage().getFrom().getLastName());

                if (memberControl.show(chatId) != null) {
                    sendMessage(message, "O", admin, "Akun mu sudah
terdaftar", chatControl);
                    pnlchat.refresh();
                    try {
                        execute(message); // Call method to send the
message
                    } catch (TelegramApiException e) {
                        e.printStackTrace();
                    }
                }
                else if (memberControl.store(new Member(chatId, name,
username))) {
                    pnlmember.refresh();

                    sendMessage(message, "O", admin, "Register berhasil!",
chatControl);
                    pnlchat.refresh();
                    try {
                        execute(message); // Call method to send the
message
                    } catch (TelegramApiException e) {
                    }
                } else {
                    sendMessage(message, "O", admin, "Register gagal",
chatControl);
                    pnlchat.refresh();
                    try {
```

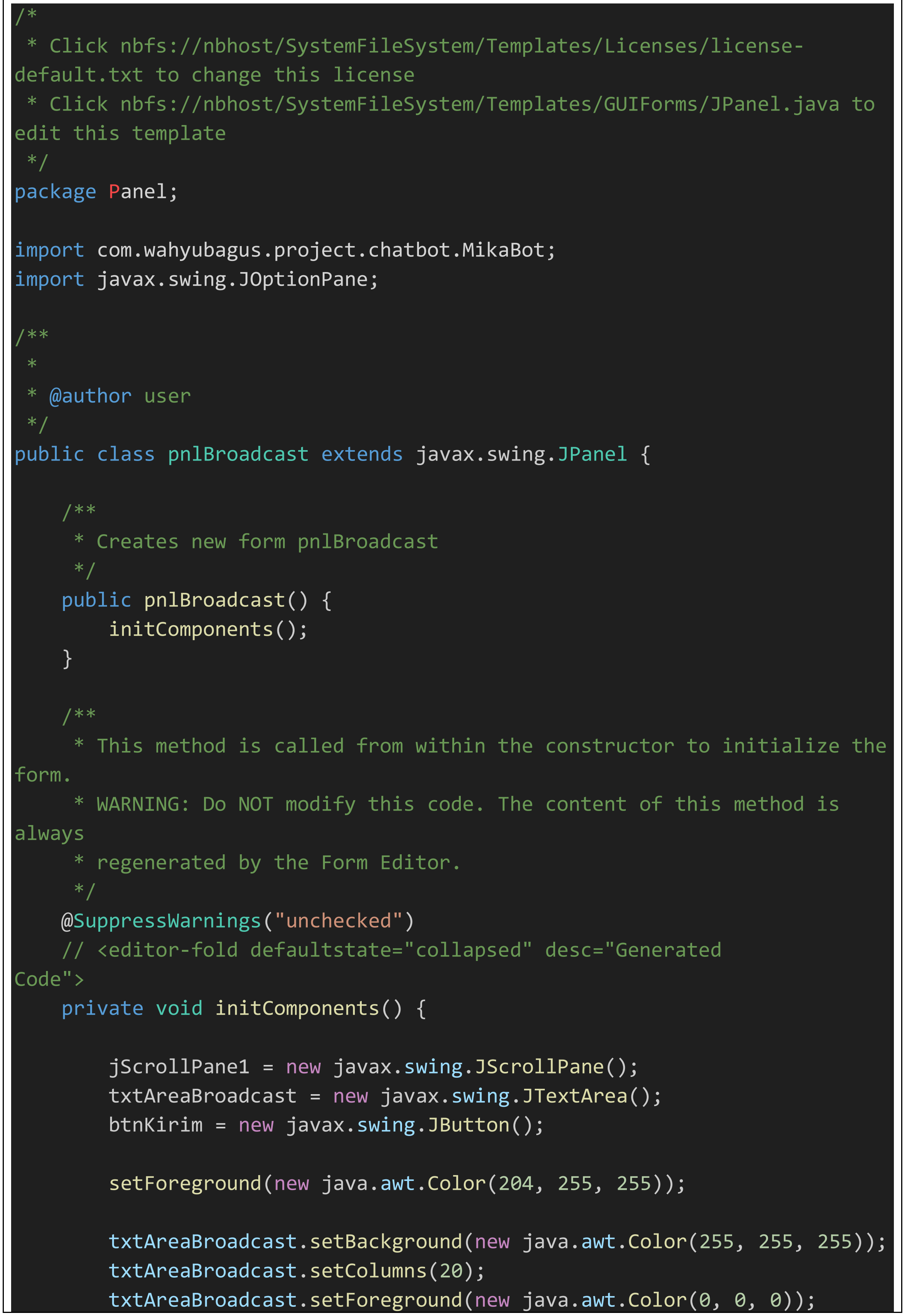


```
                        execute(message); // Call method to send the
message
                    } catch (TelegramApiException e) {
                    }
                }

            } else if (memberControl.show(chatId) == null) {
                sendMessage(message, "O", admin, "Silahkan /register
dulu !", chatControl);
                pnlchat.refresh();
                try {
                    execute(message); // Call method to send the message
                } catch (TelegramApiException e) {
                }
            } else {
                String keyword = update.getMessage().getText();
                String response = keywordControl.getResponse(keyword);
                message.setText(response);

                sendMessage(message, "O", admin, response, chatControl);
                pnlchat.refresh();
                }
            }

    }


    @Override
    public String getBotUsername() {
        return "MikaBot";
    }

    public void setMemberPanel(pnlMember pnlmember) {
        this.pnlmember = pnlmember;
    }

    public void setChatPanel(pnlChat pnlchat) {
        this.pnlchat = pnlchat;
    }


    public boolean sendBroadcast(String broadcastText) {
        MemberControl memberControl = new MemberControl();
        List<Member> listMember = memberControl.index();
        SendMessage message = new SendMessage();
        message.setText(broadcastText);

        try {
            for (int i = 0; i < listMember.size(); i++) {
                message.setChatId(listMember.get(i).getId());
                execute(message); // Call method to send the message
            }
            System.out.println("chatbot.MikaBot.sendBroadcast()" + admin);
            Chat chat = new Chat("B", admin, broadcastText);
            ChatControl chatControl = new ChatControl();
            chatControl.store(chat);
            System.out.println("chatbot.MikaBot.sendBroadcast()");
            return true;
        } catch (TelegramApiException e) {
            System.out.println("chatbot.MikaBot.sendBroadcast() broadcast
gagal");
            return false;
        }
    }

    private void sendMessage(SendMessage message, String category, String
sender, String text, ChatControl chatControl) {
        message.setText(text);
        try {
            execute(message); // Call method to send the message
            Chat newChat = new Chat(category, sender, text);
            chatControl.store(newChat);
        } catch (TelegramApiException e) {
        }
    }

}
```

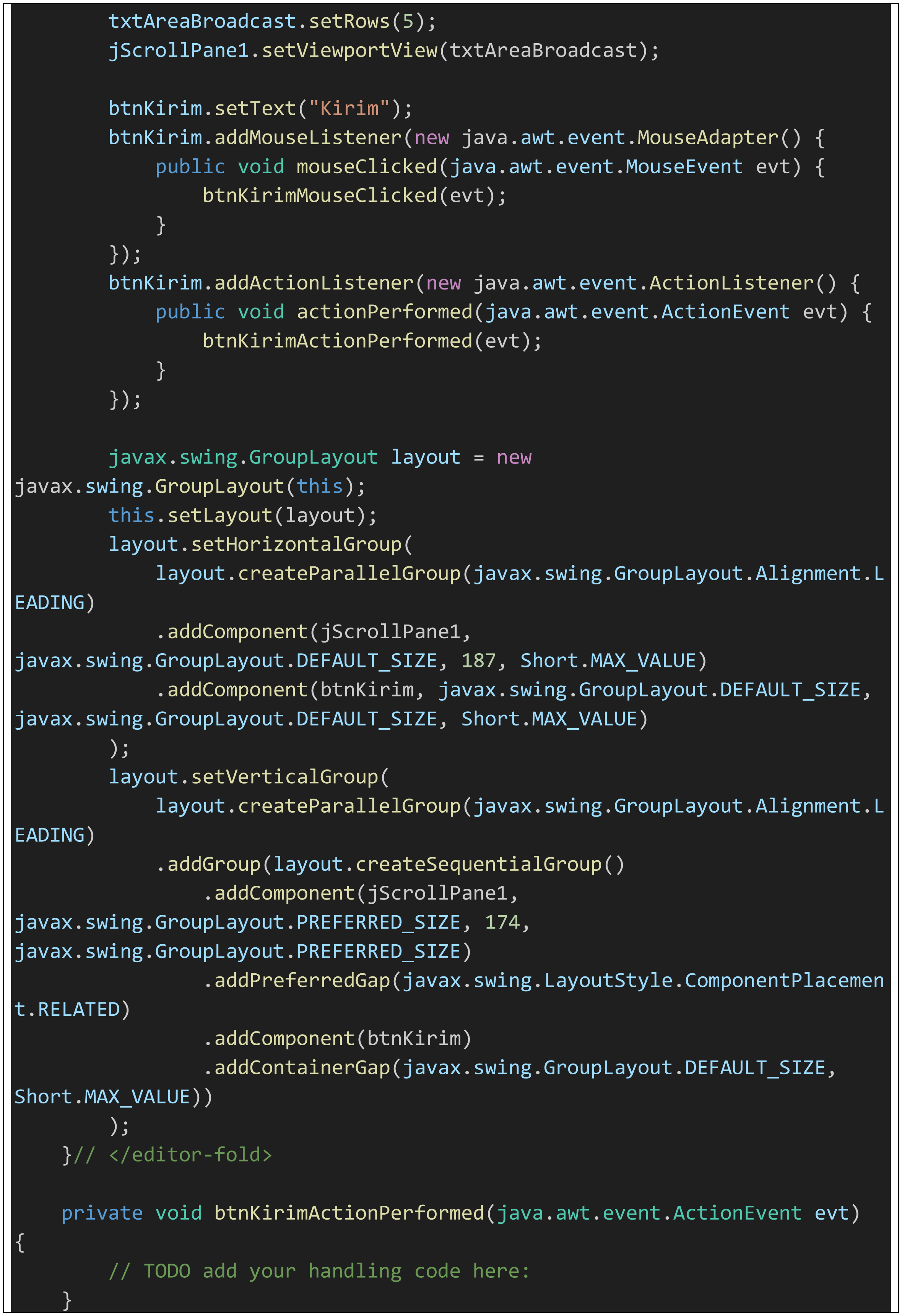
- ConnectMysql.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package com.wahyubagus.project.chatbot;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

/**
 *
 * @author user
 */
public class ConnectMysql {
    String url, usr, pwd, dbn;
    public ConnectMysql(String dbn){
        this.url = "jdbc:mysql://localhost/"+ dbn;
        this.usr = "root";
        this.pwd = "";
    }

    public ConnectMysql (String host, String user, String pass, String
dbn) {
        this.url = "jdbc:mysql://" + host + "/" + dbn;
        this.usr = user;
        this.pwd = pass;
    }

    public Connection getConnection() {
        Connection con = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection(this.url,
this.usr,this.pwd);
            //System.out.println("Berhasil ");
        }

        catch (ClassNotFoundException e) {
            System.out.println ("Error #1 : " + e.getMessage());
            System.exit(0);
        }
        catch (SQLException e) {
            System.out.println ("Error #2 : " + e.getMessage());
            System.exit(0);
        }
        return con;
    }

}
```

- pnlBroadcast.java



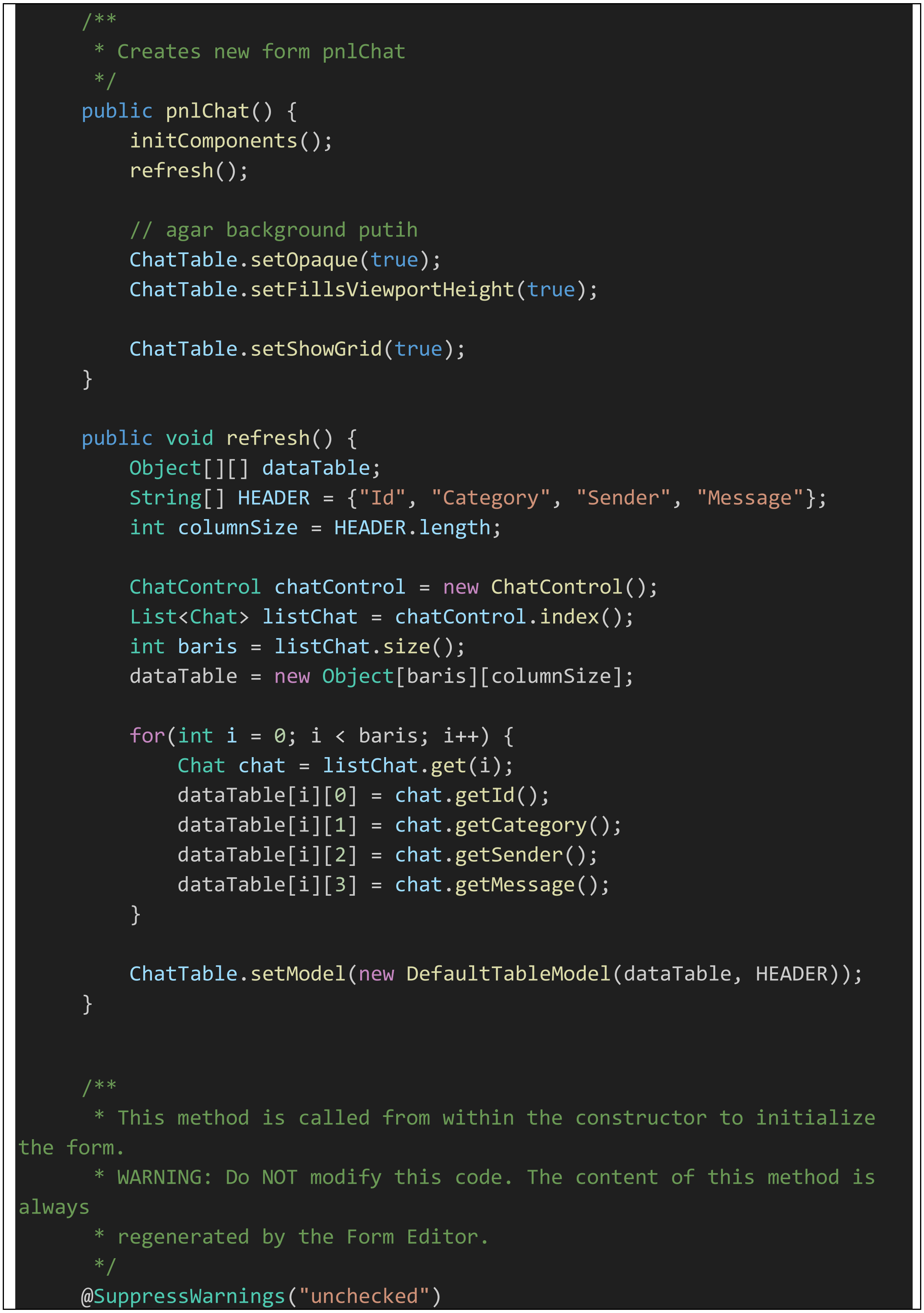
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to
edit this template
 */
package Panel;

import com.wahyubagus.project.chatbot.MikaBot;
import javax.swing.JOptionPane;

/**
 *
 * @author user
 */
public class pnlBroadcast extends javax.swing.JPanel {

    /**
     * Creates new form pnlBroadcast
     */
    public pnlBroadcast() {
        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();
        txtAreaBroadcast = new javax.swing.JTextArea();
        btnKirim = new javax.swing.JButton();

        setForeground(new java.awt.Color(204, 255, 255));

        txtAreaBroadcast.setBackground(new java.awt.Color(255, 255, 255));
        txtAreaBroadcast.setColumns(20);
        txtAreaBroadcast.setForeground(new java.awt.Color(0, 0, 0));
```



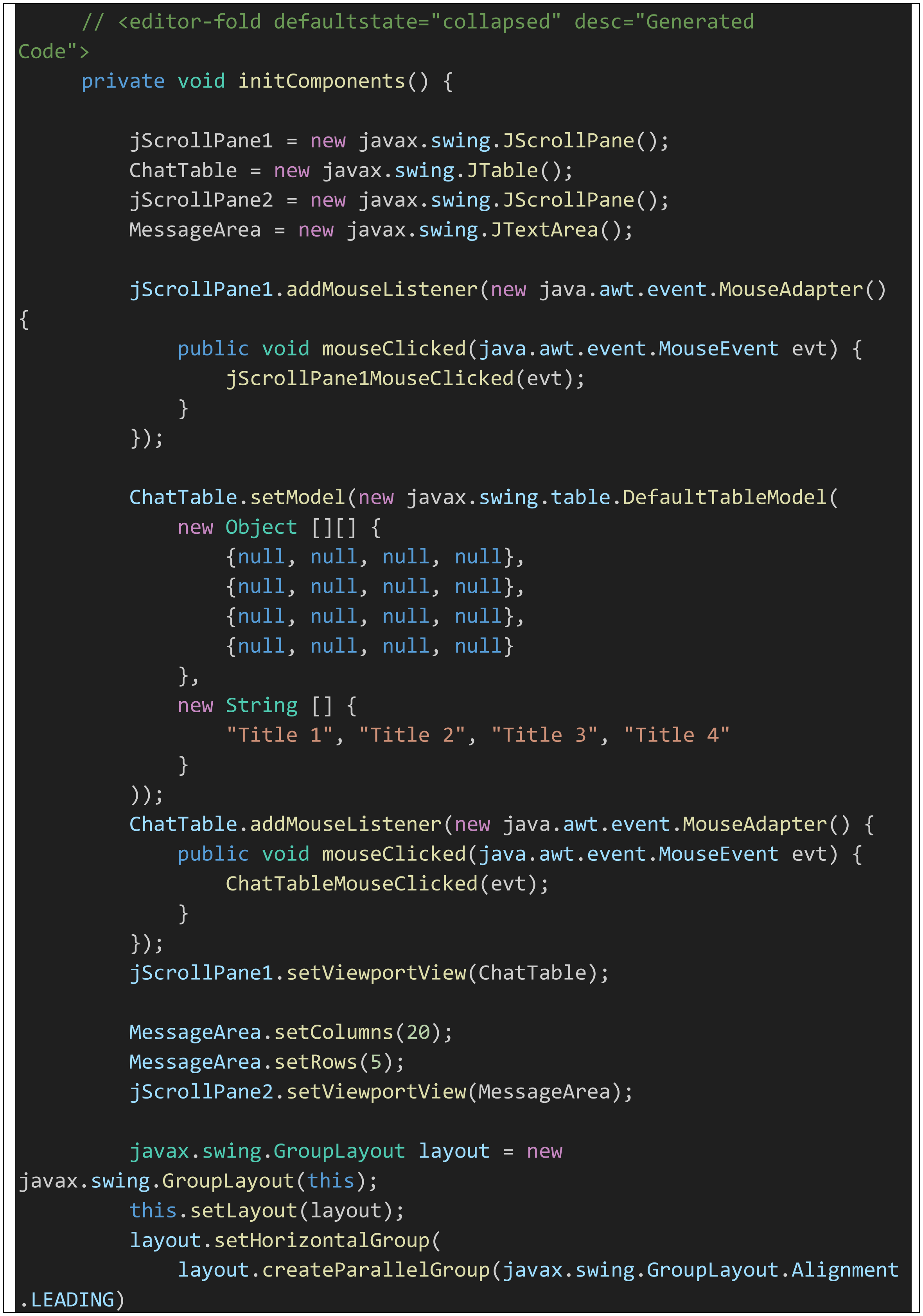
```
        txtAreaBroadcast.setRows(5);
        jScrollPane1.setViewportView(txtAreaBroadcast);

        btnKirim.setText("Kirim");
        btnKirim.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnKirimMouseClicked(evt);
            }
        });
        btnKirim.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnKirimActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
        this.setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 187, Short.MAX_VALUE)
            .addComponent(btnKirim, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 174,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacemen
t.RELATED)
                .addComponent(btnKirim)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
    }// </editor-fold>

    private void btnKirimActionPerformed(java.awt.event.ActionEvent evt)
{
        // TODO add your handling code here:
    }
    private void btnKirimMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        MikaBot mikabot = new MikaBot();
        String textBroadcast = txtAreaBroadcast.getText();
        if (mikabot.sendBroadcast(textBroadcast)) {
            JOptionPane.showMessageDialog(null, "Berhasil mengirim
broadcast");
            txtAreaBroadcast.setText("");
        } else {
            JOptionPane.showMessageDialog(null, "Gagal mengirim
broadcast");
        }
    }


    // Variables declaration - do not modify
    private javax.swing.JButton btnKirim;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea txtAreaBroadcast;
    // End of variables declaration
}
```

- pnlChat.java

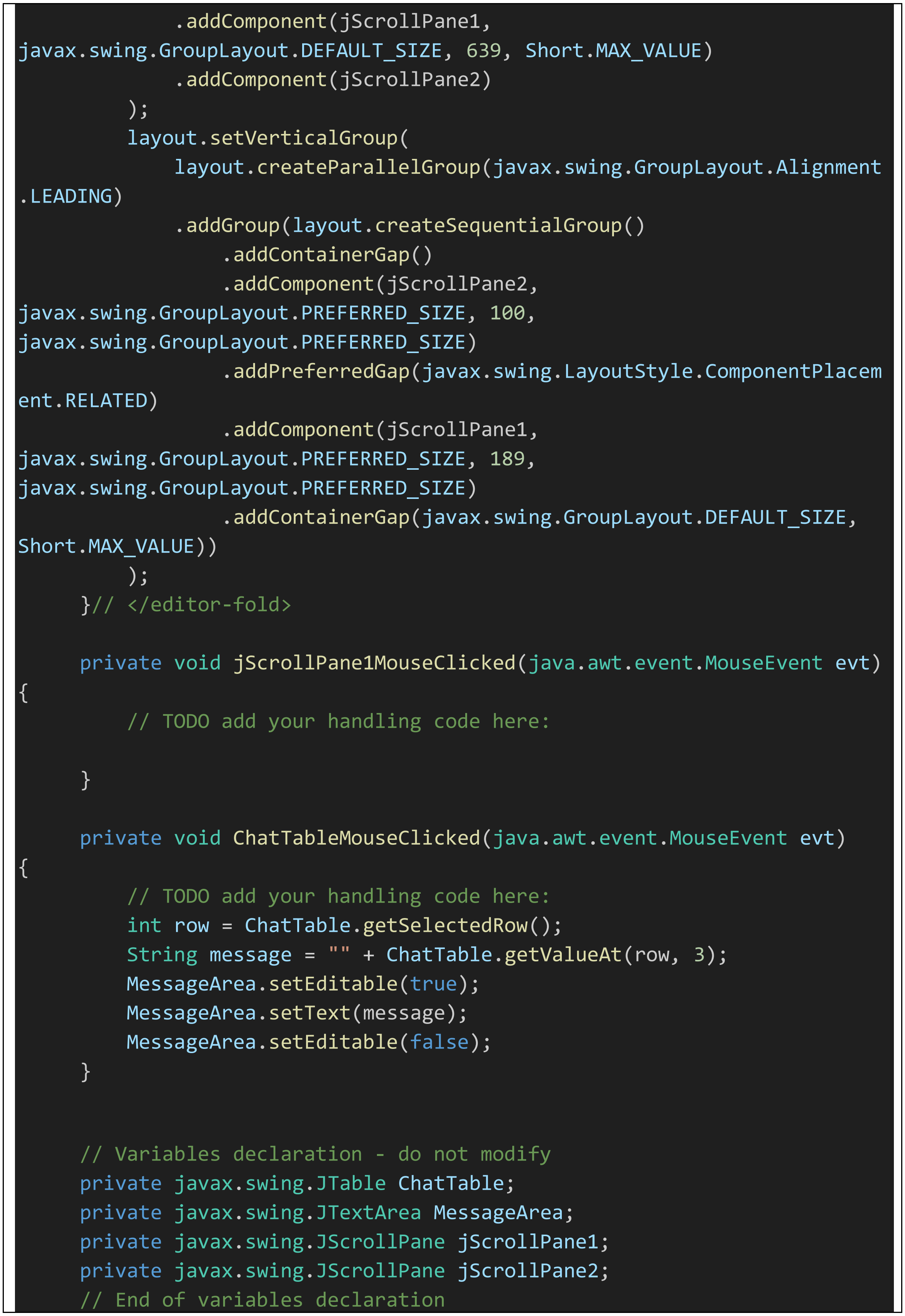
```
 /*
  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
  * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java
to edit this template
  */
 package Panel;

 import Controllers.ChatControl;
 import Model.Chat;
 import java.util.List;
 import javax.swing.table.DefaultTableModel;

 /**
  *
  * @author user
  */
 public class pnlChat extends javax.swing.JPanel {
```



```
    /**
     * Creates new form pnlChat
     */
    public pnlChat() {
        initComponents();
        refresh();

        // agar background putih
        ChatTable.setOpaque(true);
        ChatTable.setFillsViewportHeight(true);

        ChatTable.setShowGrid(true);
    }

    public void refresh() {
        Object[][] dataTable;
        String[] HEADER = {"Id", "Category", "Sender", "Message"};
        int columnSize = HEADER.length;

        ChatControl chatControl = new ChatControl();
        List<Chat> listChat = chatControl.index();
        int baris = listChat.size();
        dataTable = new Object[baris][columnSize];

        for(int i = 0; i < baris; i++) {
            Chat chat = listChat.get(i);
            dataTable[i][0] = chat.getId();
            dataTable[i][1] = chat.getCategory();
            dataTable[i][2] = chat.getSender();
            dataTable[i][3] = chat.getMessage();
        }

        ChatTable.setModel(new DefaultTableModel(dataTable, HEADER));
    }


    /**
     * 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();
        ChatTable = new javax.swing.JTable();
        jScrollPane2 = new javax.swing.JScrollPane();
        MessageArea = new javax.swing.JTextArea();

        jScrollPane1.addMouseListener(new java.awt.event.MouseAdapter()
{
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jScrollPane1MouseClicked(evt);
            }
        });

        ChatTable.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 [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        ChatTable.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                ChatTableMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(ChatTable);

        MessageArea.setColumns(20);
        MessageArea.setRows(5);
        jScrollPane2.setViewportView(MessageArea);

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
        this.setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
```

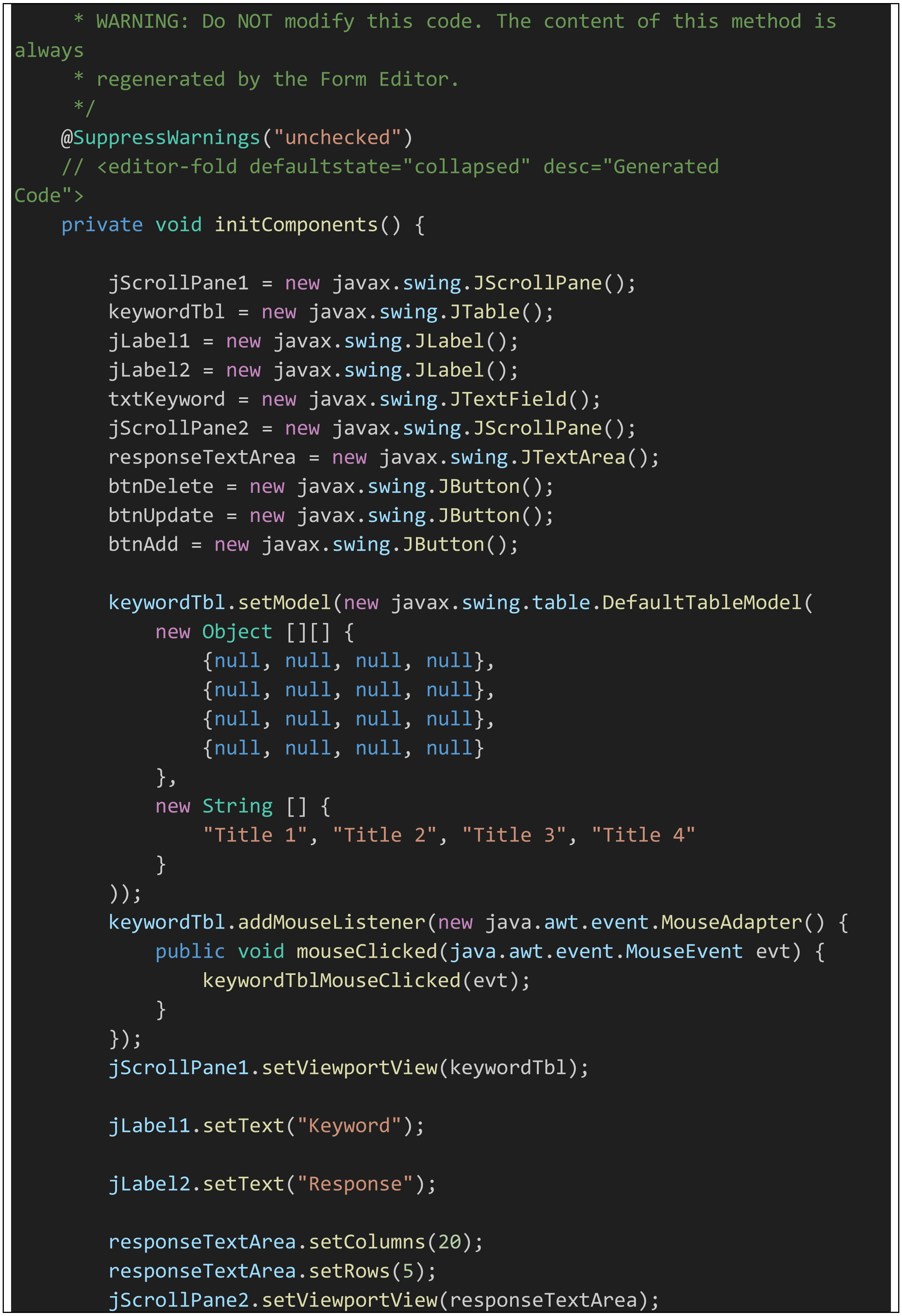


```
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 639, Short.MAX_VALUE)
                .addComponent(jScrollPane2)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacem
ent.RELATED)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 189,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
    }// </editor-fold>

    private void jScrollPane1MouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:

    }

    private void ChatTableMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        int row = ChatTable.getSelectedRow();
        String message = "" + ChatTable.getValueAt(row, 3);
        MessageArea.setEditable(true);
        MessageArea.setText(message);
        MessageArea.setEditable(false);
    }


    // Variables declaration - do not modify
    private javax.swing.JTable ChatTable;
    private javax.swing.JTextArea MessageArea;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    // End of variables declaration
```

```
}
```

- pnlKeyword.java

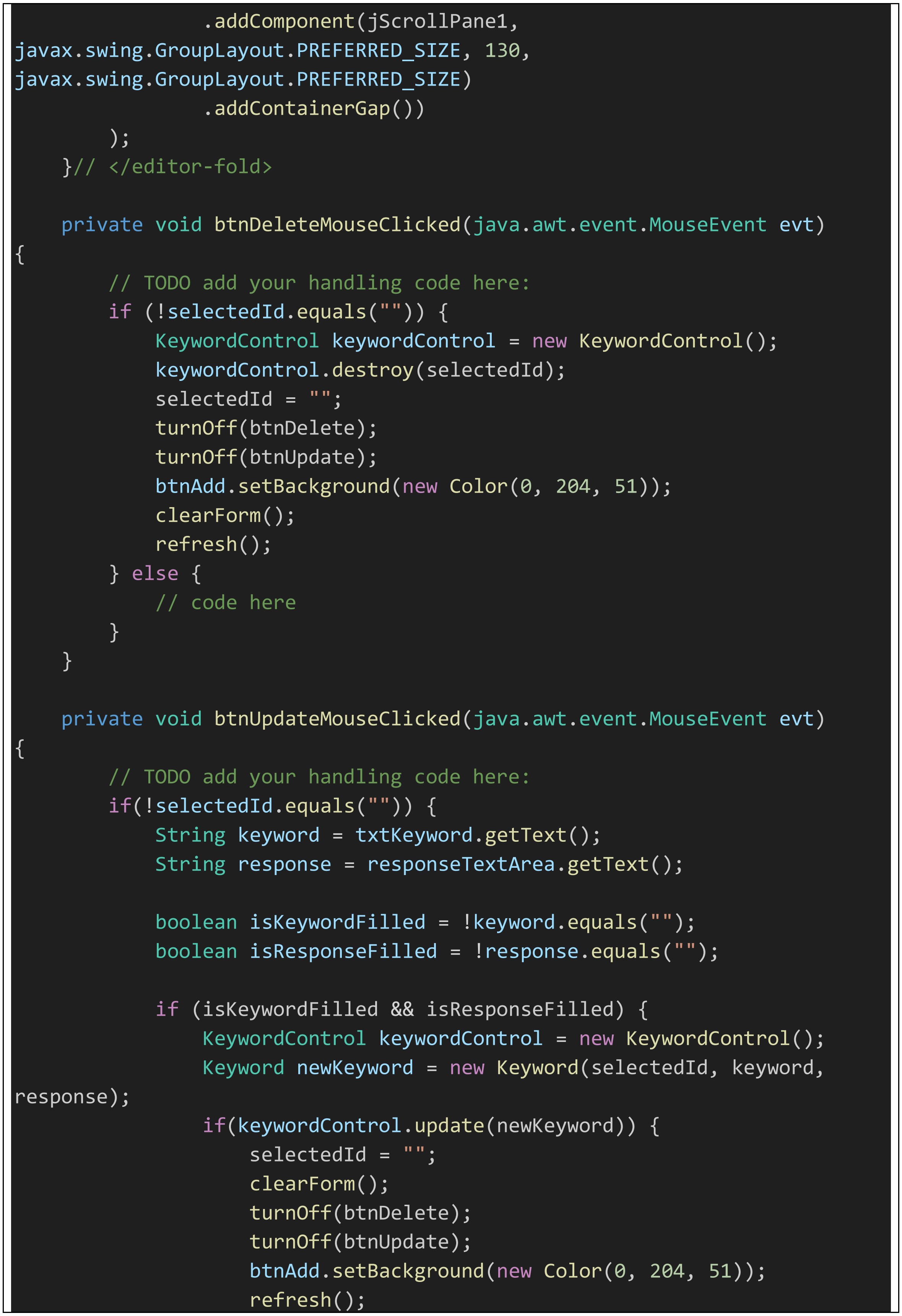
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this template
 */
package Panel;

import Controllers.KeywordControl;
import Model.Keyword;
import java.awt.Color;
import java.util.List;
import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author user
 */
public class pnlKeyword extends javax.swing.JPanel {
    private String selectedId = "";
    /**
     * Creates new form pnlKeyword
     */
    public pnlKeyword() {
        initComponents();
        turnOff(btnDelete);
        turnOff(btnUpdate);
        refresh();

        keywordTbl.setOpaque(true);
        keywordTbl.setFillsViewportHeight(true);

        keywordTbl.setShowGrid(true);
    }

    /**
     * This method is called from within the constructor to initialize the form.
```



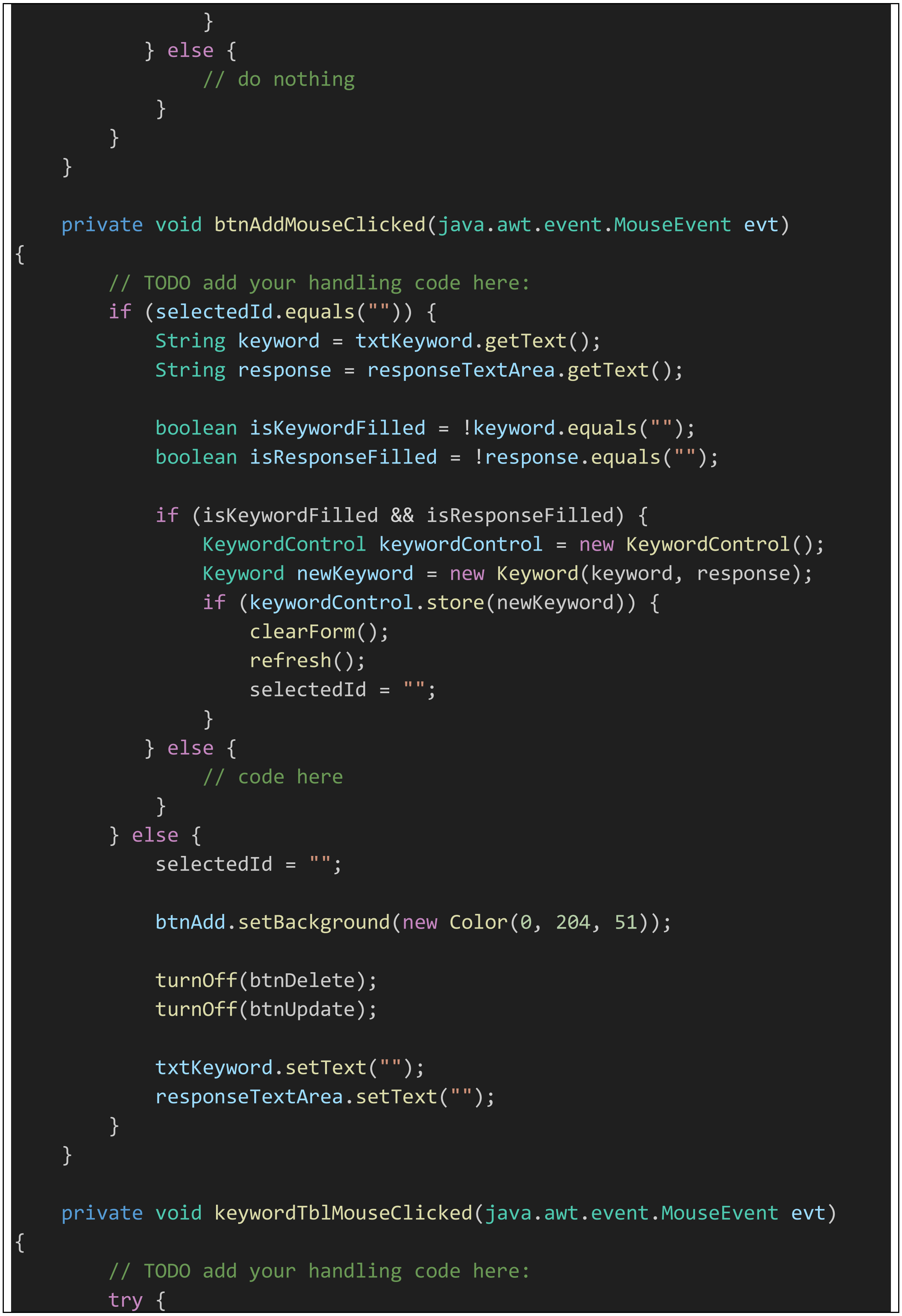
```
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        keywordTbl = new javax.swing.JTable();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        txtKeyword = new javax.swing.JTextField();
        jScrollPane2 = new javax.swing.JScrollPane();
        responseTextArea = new javax.swing.JTextArea();
        btnDelete = new javax.swing.JButton();
        btnUpdate = new javax.swing.JButton();
        btnAdd = new javax.swing.JButton();

        keywordTbl.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 [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        keywordTbl.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                keywordTblMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(keywordTbl);

        jLabel1.setText("Keyword");

        jLabel2.setText("Response");

        responseTextArea.setColumns(20);
        responseTextArea.setRows(5);
        jScrollPane2.setViewportView(responseTextArea);
```

```java
        btnDelete.setText("Delete");
        btnDelete.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnDeleteMouseClicked(evt);
            }
        });

        btnUpdate.setText("Update");
        btnUpdate.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnUpdateMouseClicked(evt);
            }
        });

        btnAdd.setText("ADD");
        btnAdd.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnAddMouseClicked(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
        this.setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 471, Short.MAX_VALUE)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.LEADING, false)
                            .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 65,
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addPreferredGap(javax.swing.LayoutStyle.Component
Placement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

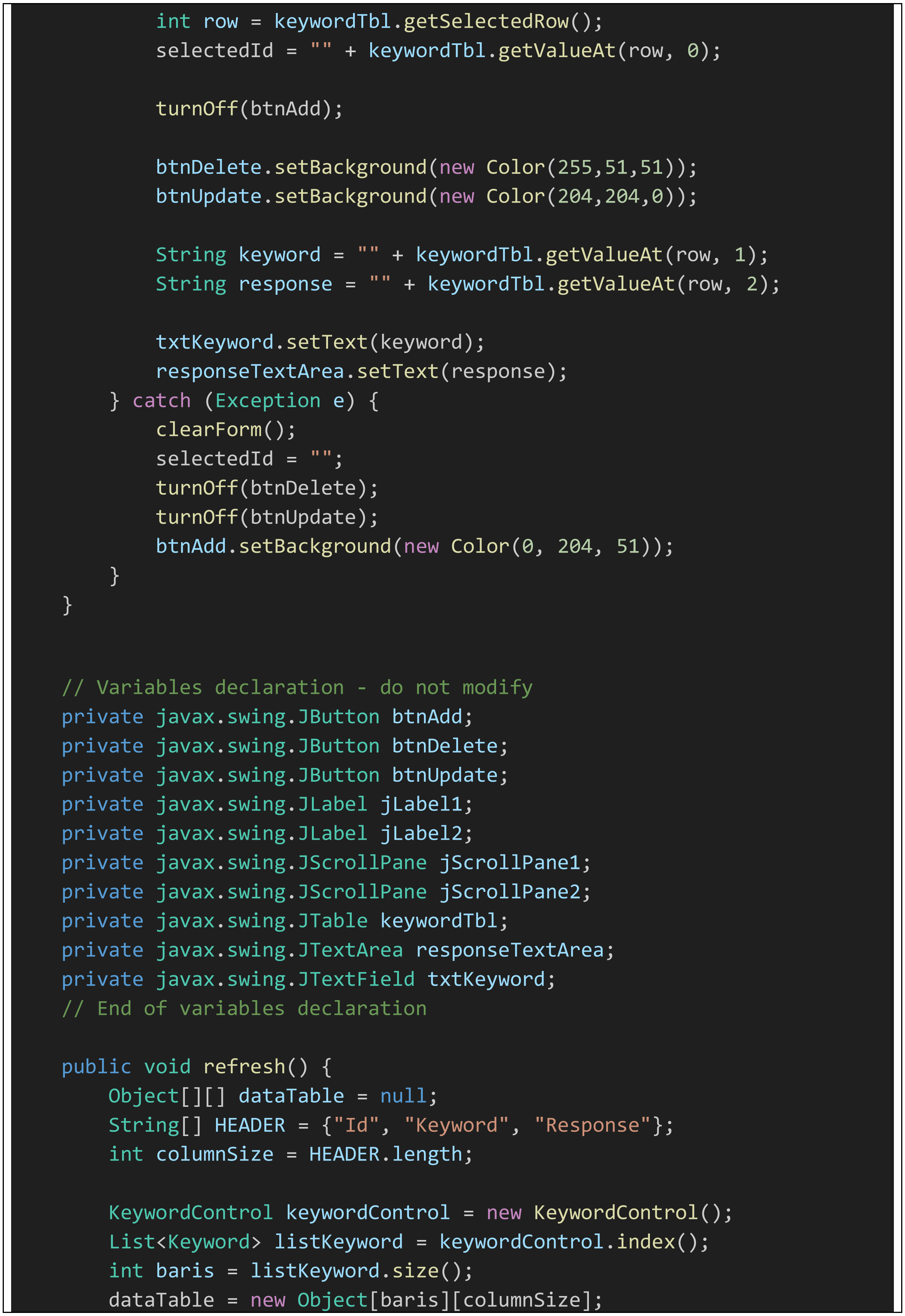
```
                        .addGroup(layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.LEADING, false)
                            .addComponent(txtKeyword)
                            .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 300,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(btnDelete)
                        .addGap(18, 18, 18)
                        .addComponent(btnUpdate)
                        .addPreferredGap(javax.swing.LayoutStyle.Component
Placement.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(btnAdd))))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addGap(27, 27, 27)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(txtKeyword,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacemen
t.RELATED)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
                    .addComponent(jLabel2)
                    .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacemen
t.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
                    .addComponent(btnDelete)
                    .addComponent(btnUpdate)
                    .addComponent(btnAdd))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacemen
t.UNRELATED)
```



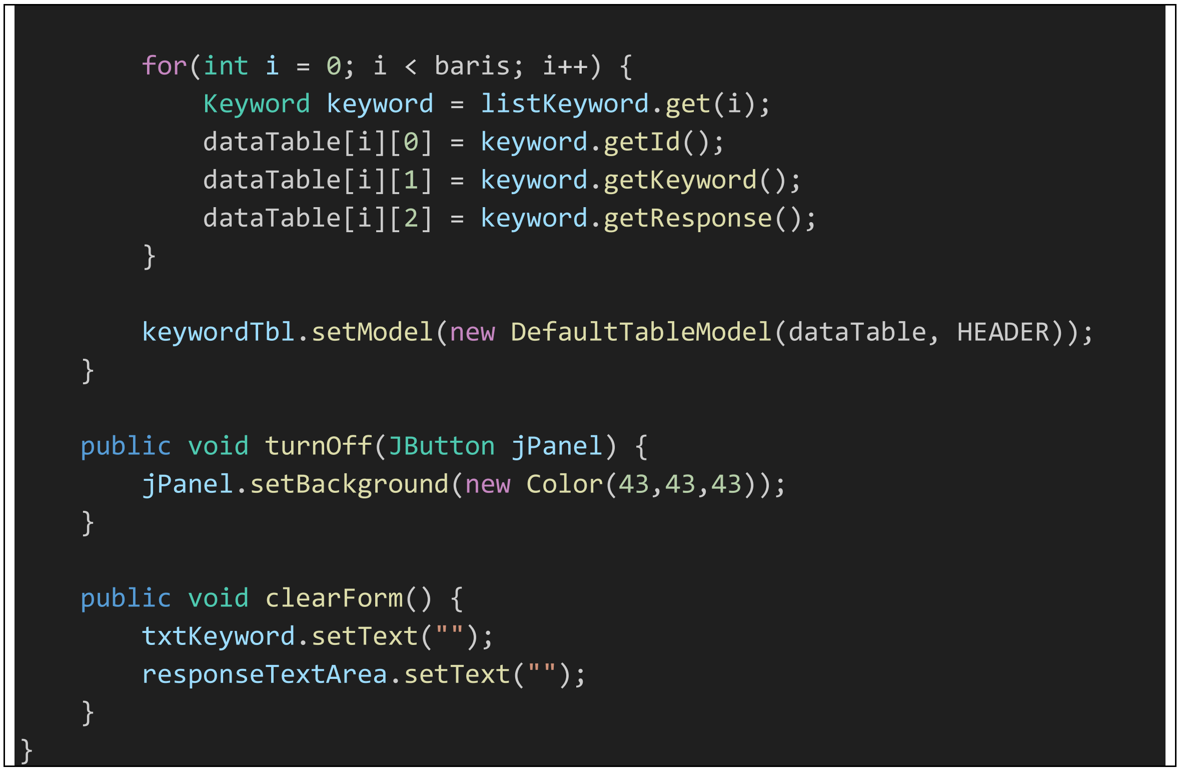
```
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 130,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );
    }// </editor-fold>

    private void btnDeleteMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        if (!selectedId.equals("")) {
            KeywordControl keywordControl = new KeywordControl();
            keywordControl.destroy(selectedId);
            selectedId = "";
            turnOff(btnDelete);
            turnOff(btnUpdate);
            btnAdd.setBackground(new Color(0, 204, 51));
            clearForm();
            refresh();
        } else {
            // code here
        }
    }

    private void btnUpdateMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        if(!selectedId.equals("")) {
            String keyword = txtKeyword.getText();
            String response = responseTextArea.getText();

            boolean isKeywordFilled = !keyword.equals("");
            boolean isResponseFilled = !response.equals("");

            if (isKeywordFilled && isResponseFilled) {
                KeywordControl keywordControl = new KeywordControl();
                Keyword newKeyword = new Keyword(selectedId, keyword,
response);
                if(keywordControl.update(newKeyword)) {
                    selectedId = "";
                    clearForm();
                    turnOff(btnDelete);
                    turnOff(btnUpdate);
                    btnAdd.setBackground(new Color(0, 204, 51));
                    refresh();
```



```
                }
           } else {
                // do nothing
            }
        }
    }

    private void btnAddMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        if (selectedId.equals("")) {
            String keyword = txtKeyword.getText();
            String response = responseTextArea.getText();

            boolean isKeywordFilled = !keyword.equals("");
            boolean isResponseFilled = !response.equals("");

            if (isKeywordFilled && isResponseFilled) {
                KeywordControl keywordControl = new KeywordControl();
                Keyword newKeyword = new Keyword(keyword, response);
                if (keywordControl.store(newKeyword)) {
                    clearForm();
                    refresh();
                    selectedId = "";
                }
           } else {
                // code here
            }
        } else {
            selectedId = "";

            btnAdd.setBackground(new Color(0, 204, 51));

            turnOff(btnDelete);
            turnOff(btnUpdate);

            txtKeyword.setText("");
            responseTextArea.setText("");
        }
    }

    private void keywordTblMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        try {
```

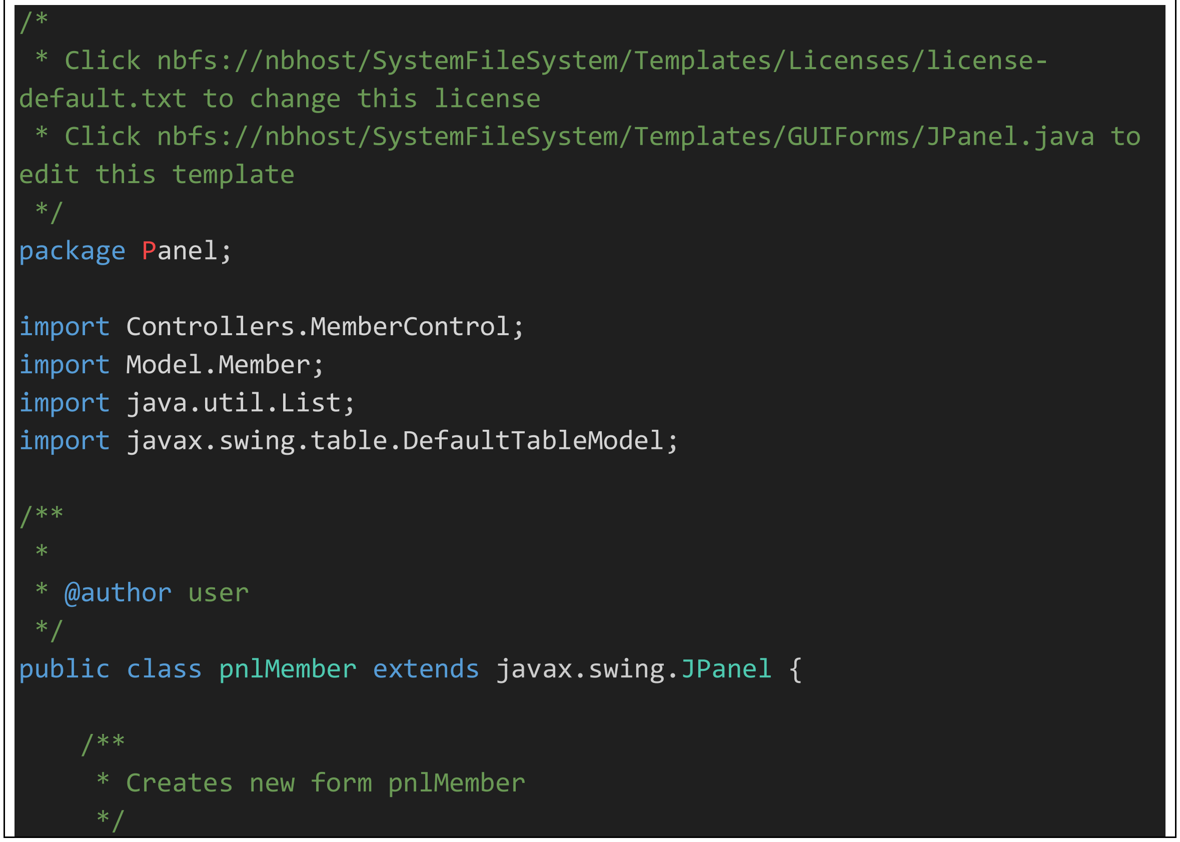


```
            int row = keywordTbl.getSelectedRow();
            selectedId = "" + keywordTbl.getValueAt(row, 0);

            turnOff(btnAdd);

            btnDelete.setBackground(new Color(255,51,51));
            btnUpdate.setBackground(new Color(204,204,0));

            String keyword = "" + keywordTbl.getValueAt(row, 1);
            String response = "" + keywordTbl.getValueAt(row, 2);

            txtKeyword.setText(keyword);
            responseTextArea.setText(response);
        } catch (Exception e) {
            clearForm();
            selectedId = "";
            turnOff(btnDelete);
            turnOff(btnUpdate);
            btnAdd.setBackground(new Color(0, 204, 51));
        }
    }


    // Variables declaration - do not modify
    private javax.swing.JButton btnAdd;
    private javax.swing.JButton btnDelete;
    private javax.swing.JButton btnUpdate;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTable keywordTbl;
    private javax.swing.JTextArea responseTextArea;
    private javax.swing.JTextField txtKeyword;
    // End of variables declaration

    public void refresh() {
        Object[][] dataTable = null;
        String[] HEADER = {"Id", "Keyword", "Response"};
        int columnSize = HEADER.length;

        KeywordControl keywordControl = new KeywordControl();
        List<Keyword> listKeyword = keywordControl.index();
        int baris = listKeyword.size();
        dataTable = new Object[baris][columnSize];
```

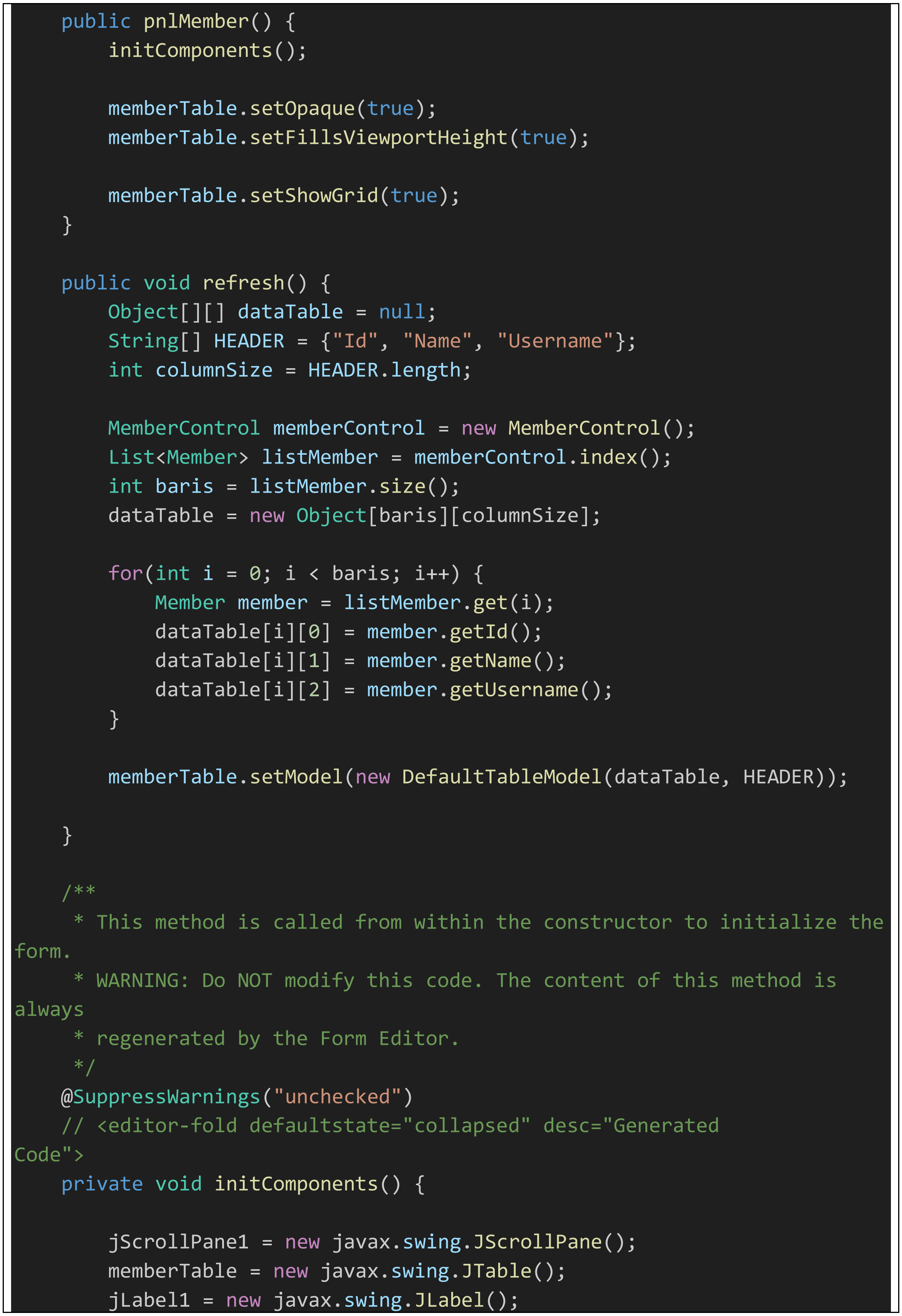


```java
        for(int i = 0; i < baris; i++) {
            Keyword keyword = listKeyword.get(i);
            dataTable[i][0] = keyword.getId();
            dataTable[i][1] = keyword.getKeyword();
            dataTable[i][2] = keyword.getResponse();
        }
        
        keywordTbl.setModel(new DefaultTableModel(dataTable, HEADER));
    }
    
    public void turnOff(JButton jPanel) {
        jPanel.setBackground(new Color(43,43,43));
    }
    
    public void clearForm() {
        txtKeyword.setText("");
        responseTextArea.setText("");
    }
}
```

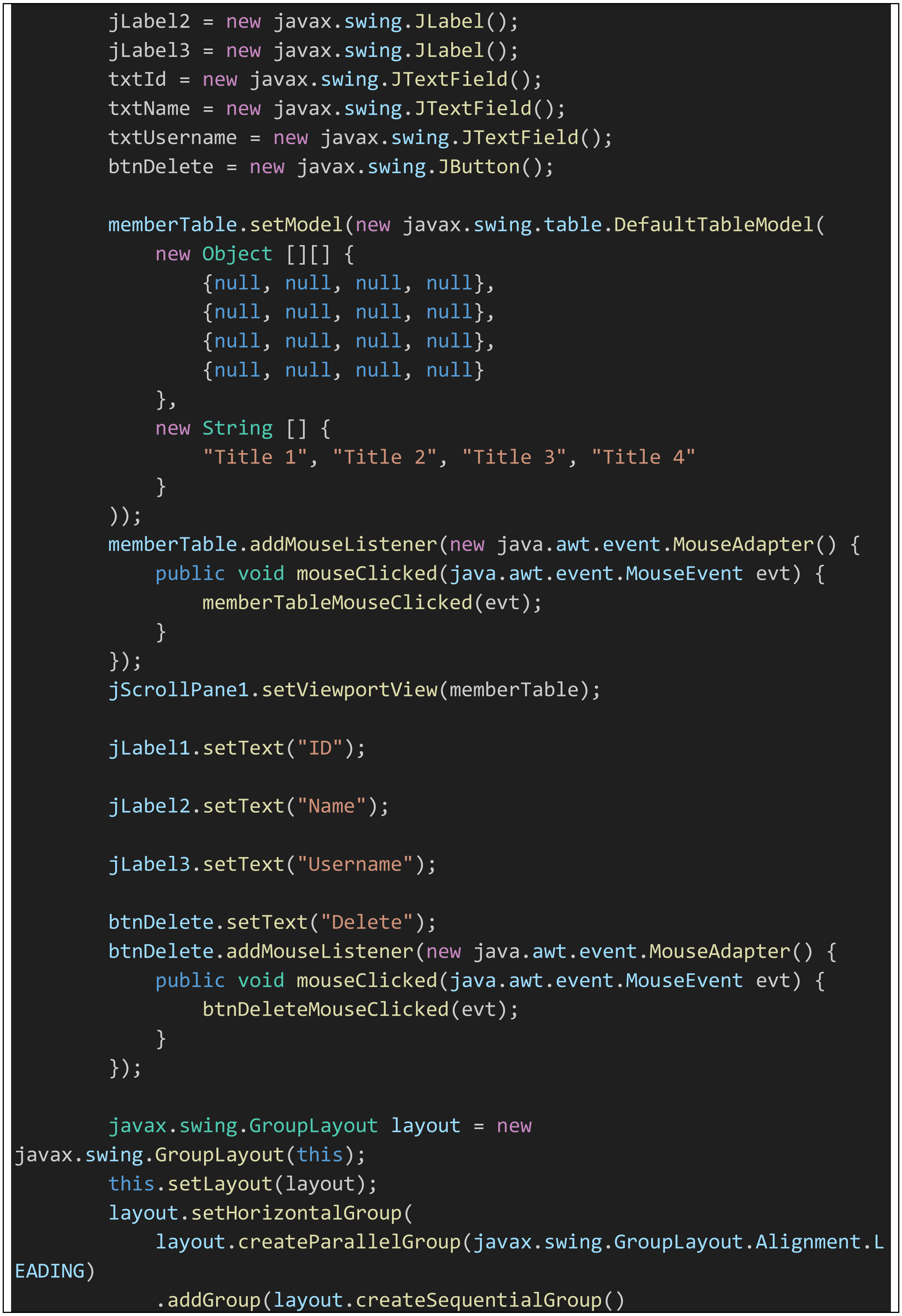
- pnlMember.java



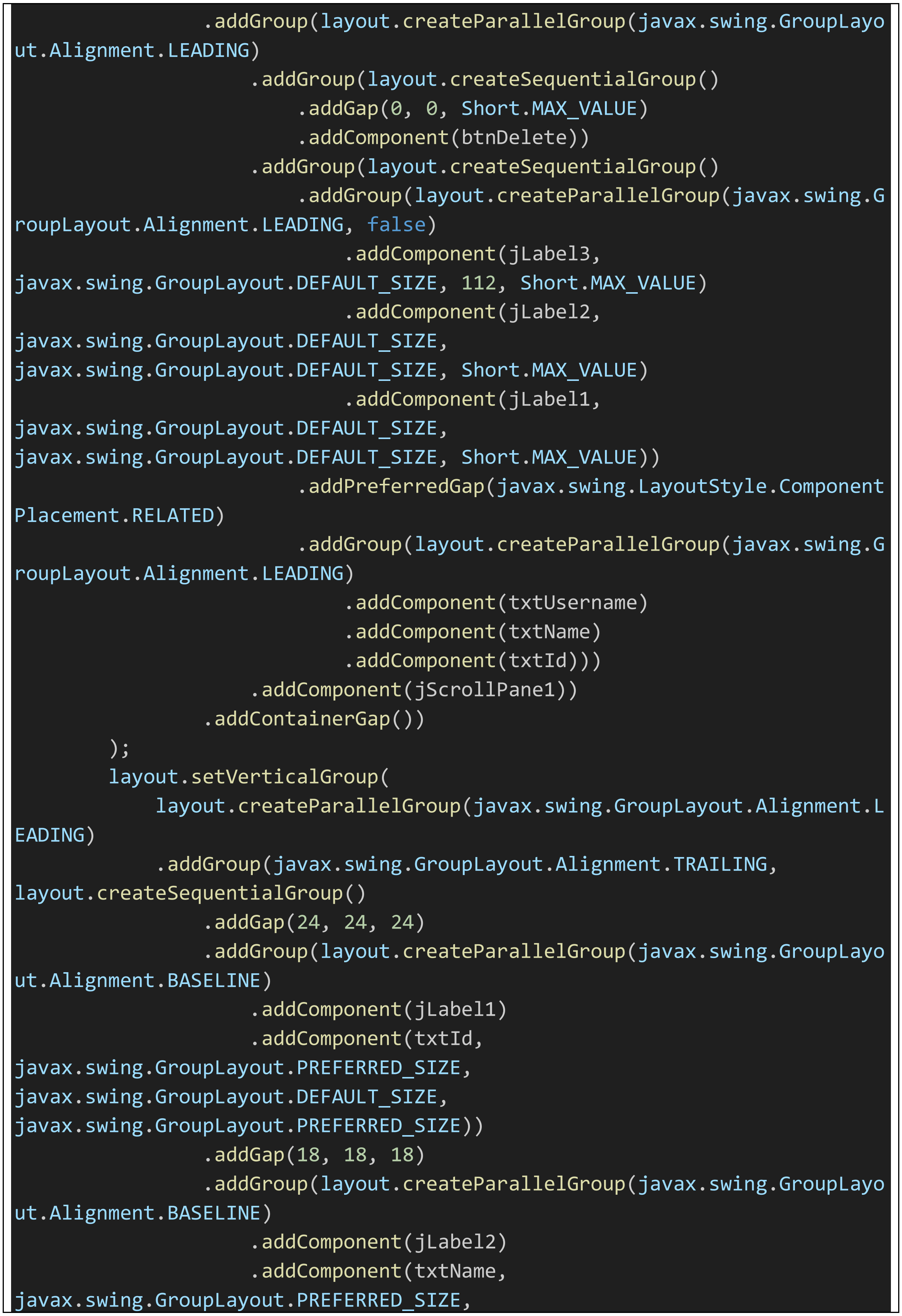
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to
edit this template
 */
package Panel;

import Controllers.MemberControl;
import Model.Member;
import java.util.List;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author user
 */
public class pnlMember extends javax.swing.JPanel {

    /**
     * Creates new form pnlMember
     */
```



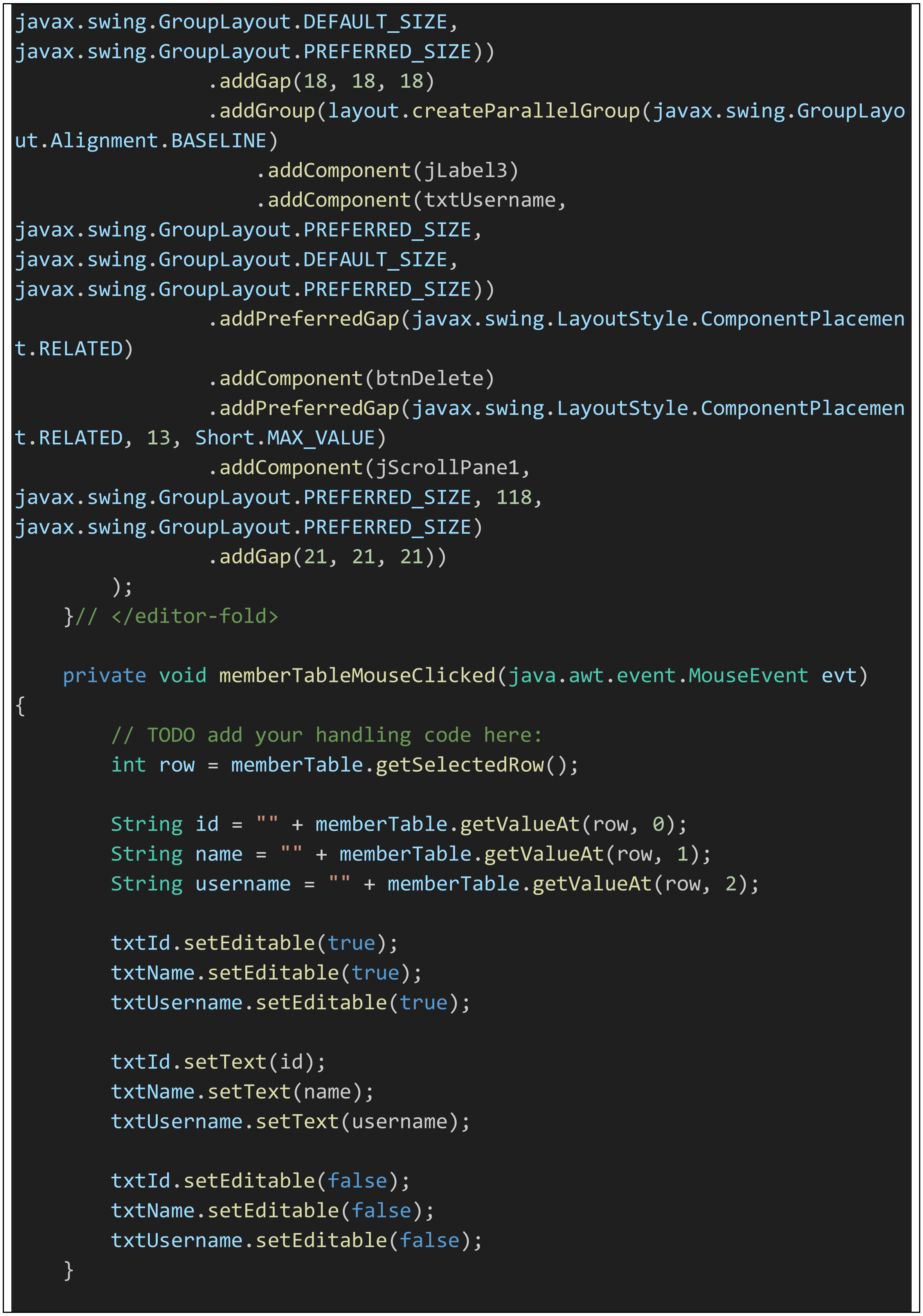
```
    public pnlMember() {
        initComponents();

        memberTable.setOpaque(true);
        memberTable.setFillsViewportHeight(true);

        memberTable.setShowGrid(true);
    }

    public void refresh() {
        Object[][] dataTable = null;
        String[] HEADER = {"Id", "Name", "Username"};
        int columnSize = HEADER.length;

        MemberControl memberControl = new MemberControl();
        List<Member> listMember = memberControl.index();
        int baris = listMember.size();
        dataTable = new Object[baris][columnSize];

        for(int i = 0; i < baris; i++) {
            Member member = listMember.get(i);
            dataTable[i][0] = member.getId();
            dataTable[i][1] = member.getName();
            dataTable[i][2] = member.getUsername();
        }

        memberTable.setModel(new DefaultTableModel(dataTable, HEADER));

    }

    /**
     * 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();
        memberTable = new javax.swing.JTable();
        jLabel1 = new javax.swing.JLabel();
```



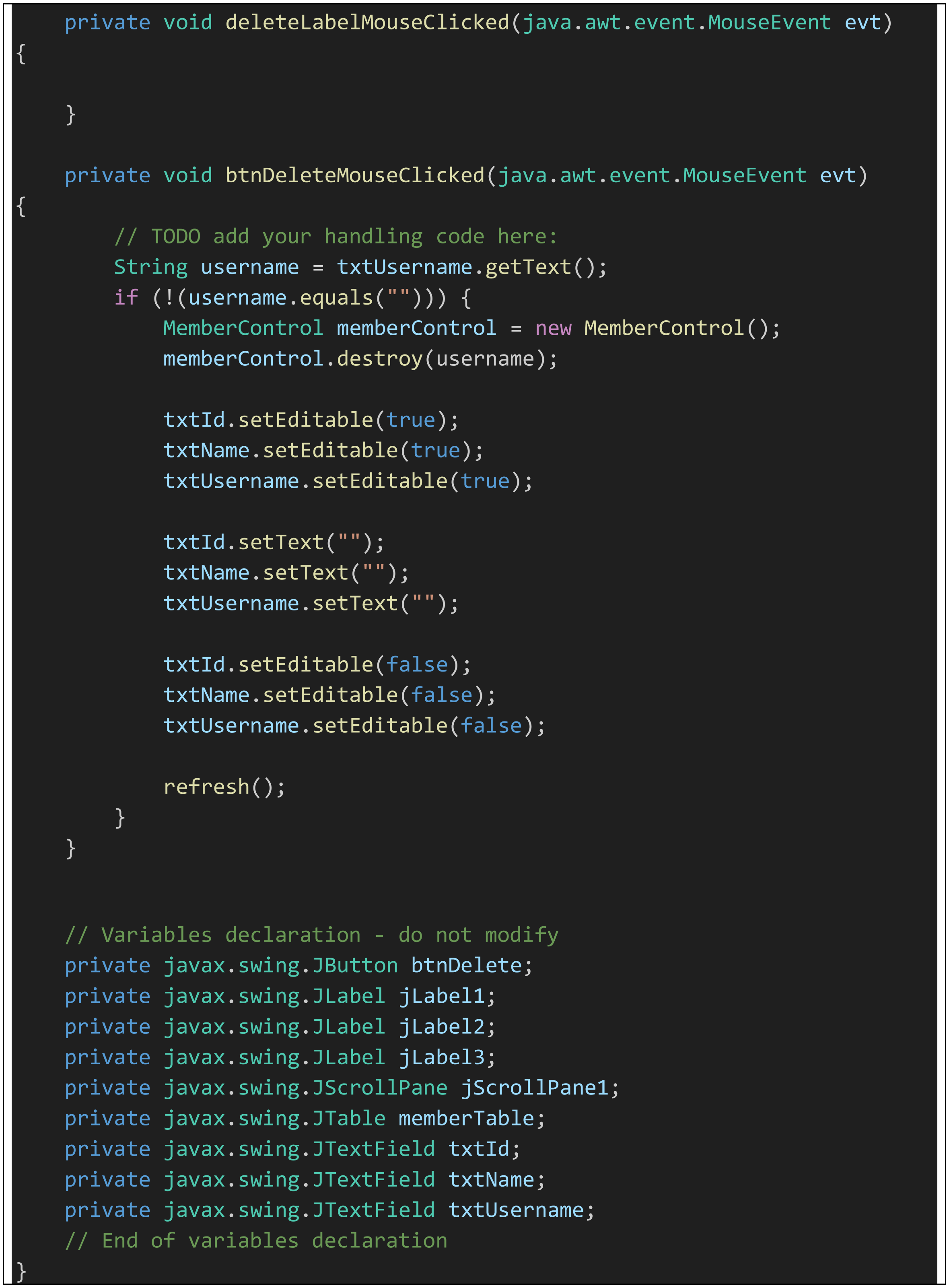
```
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        txtId = new javax.swing.JTextField();
        txtName = new javax.swing.JTextField();
        txtUsername = new javax.swing.JTextField();
        btnDelete = new javax.swing.JButton();

        memberTable.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 [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        memberTable.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                memberTableMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(memberTable);

        jLabel1.setText("ID");

        jLabel2.setText("Name");

        jLabel3.setText("Username");

        btnDelete.setText("Delete");
        btnDelete.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnDeleteMouseClicked(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
        this.setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addGroup(layout.createSequentialGroup()
```



```
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(0, 0, Short.MAX_VALUE)
                        .addComponent(btnDelete))
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.LEADING, false)
                            .addComponent(jLabel3,
javax.swing.GroupLayout.DEFAULT_SIZE, 112, Short.MAX_VALUE)
                            .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jLabel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addPreferredGap(javax.swing.LayoutStyle.Component
Placement.RELATED)
                        .addGroup(layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.LEADING)
                            .addComponent(txtUsername)
                            .addComponent(txtName)
                            .addComponent(txtId)))
                    .addComponent(jScrollPane1))
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addGap(24, 24, 24)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(txtId,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(txtName,
javax.swing.GroupLayout.PREFERRED_SIZE,
```

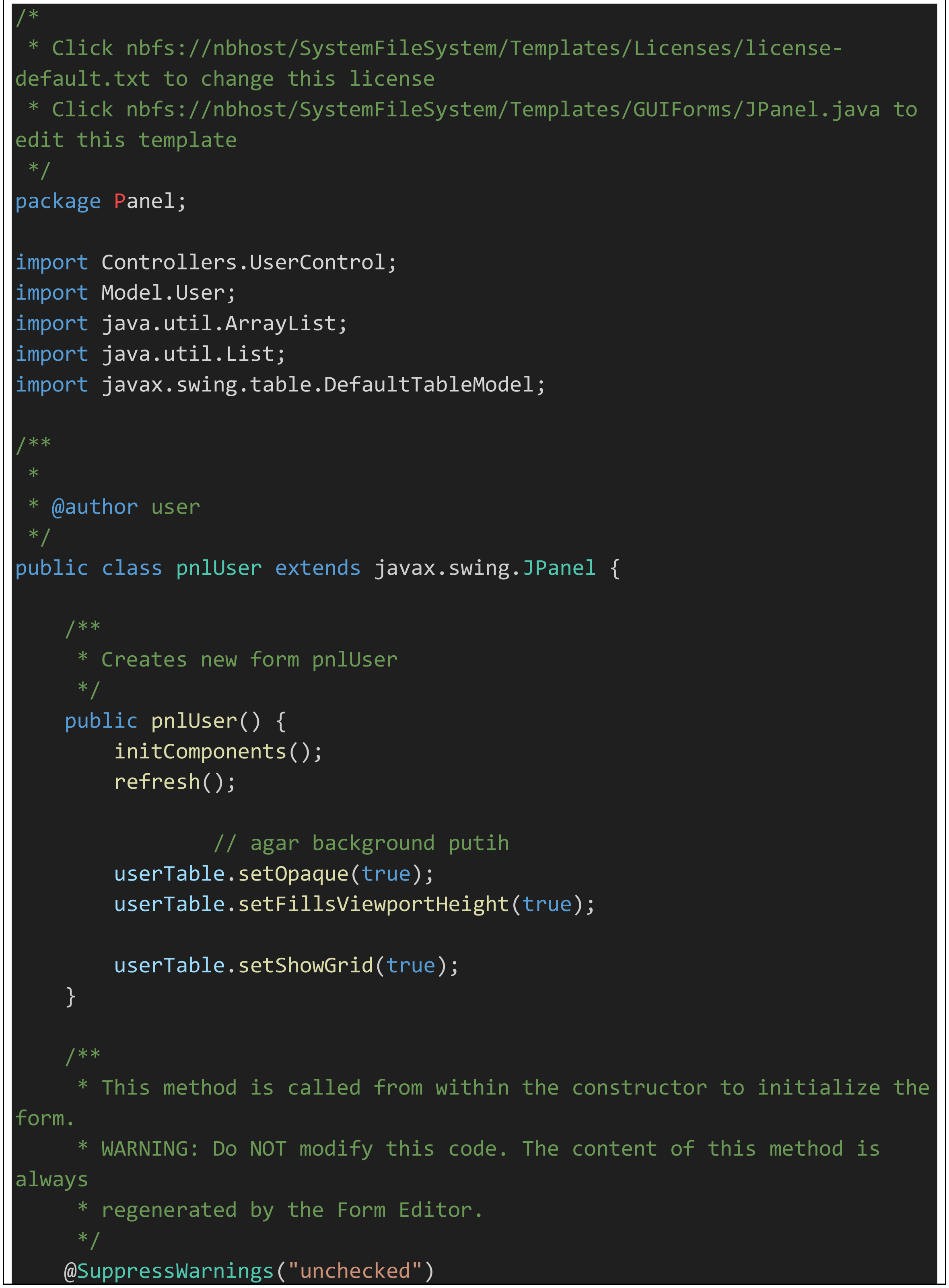


```
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(txtUsername,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacemen
t.RELATED)
                .addComponent(btnDelete)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacemen
t.RELATED, 13, Short.MAX_VALUE)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 118,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(21, 21, 21))
        );
    }// </editor-fold>

    private void memberTableMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        int row = memberTable.getSelectedRow();

        String id = "" + memberTable.getValueAt(row, 0);
        String name = "" + memberTable.getValueAt(row, 1);
        String username = "" + memberTable.getValueAt(row, 2);

        txtId.setEditable(true);
        txtName.setEditable(true);
        txtUsername.setEditable(true);

        txtId.setText(id);
        txtName.setText(name);
        txtUsername.setText(username);

        txtId.setEditable(false);
        txtName.setEditable(false);
        txtUsername.setEditable(false);
    }
```

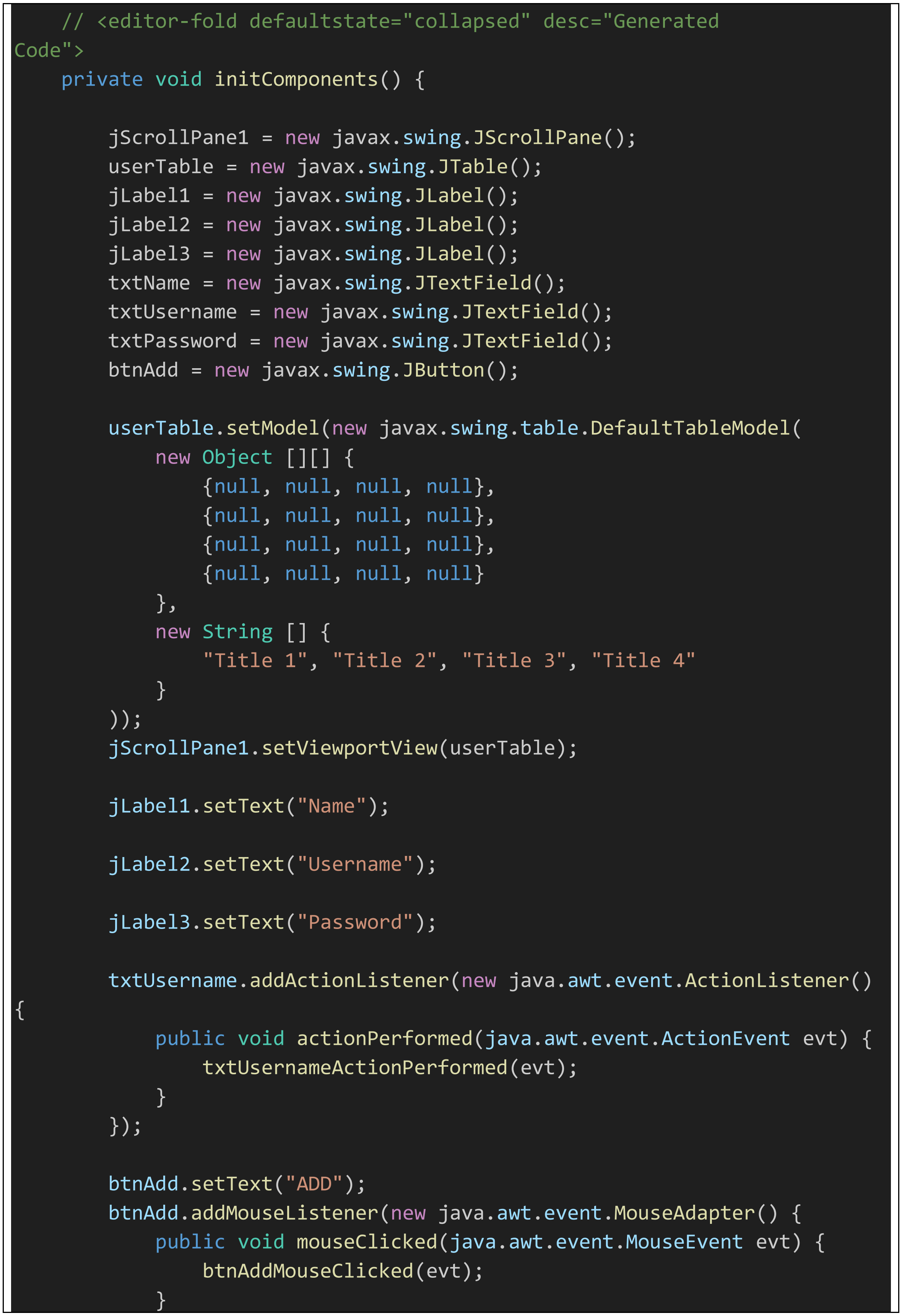


```
    private void deleteLabelMouseClicked(java.awt.event.MouseEvent evt)
{

    }

    private void btnDeleteMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        String username = txtUsername.getText();
        if (!(username.equals(""))) {
            MemberControl memberControl = new MemberControl();
            memberControl.destroy(username);

            txtId.setEditable(true);
            txtName.setEditable(true);
            txtUsername.setEditable(true);

            txtId.setText("");
            txtName.setText("");
            txtUsername.setText("");

            txtId.setEditable(false);
            txtName.setEditable(false);
            txtUsername.setEditable(false);

            refresh();
        }
    }


    // Variables declaration - do not modify
    private javax.swing.JButton btnDelete;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable memberTable;
    private javax.swing.JTextField txtId;
    private javax.swing.JTextField txtName;
    private javax.swing.JTextField txtUsername;
    // End of variables declaration
}
```

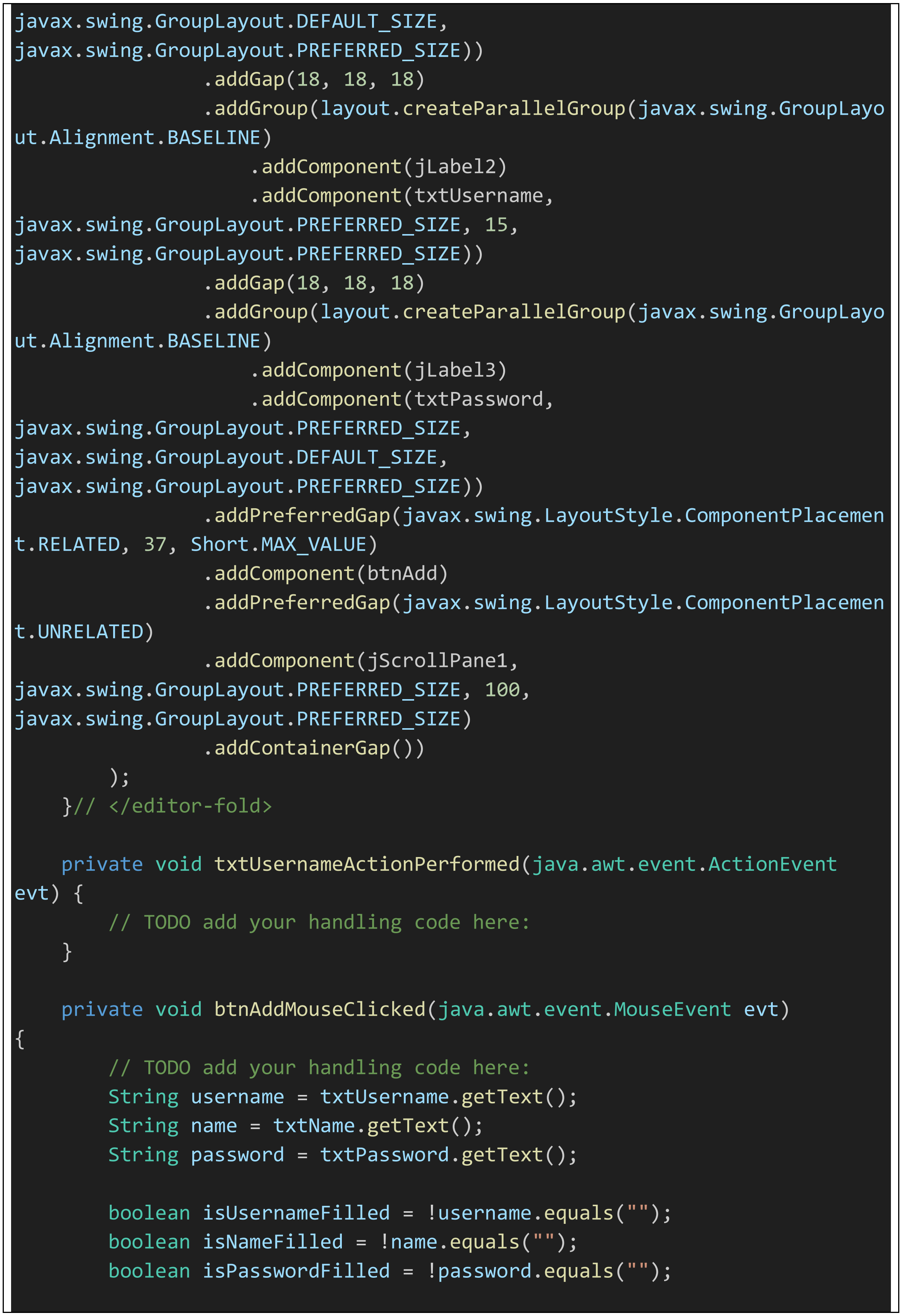
- pnlUser,java



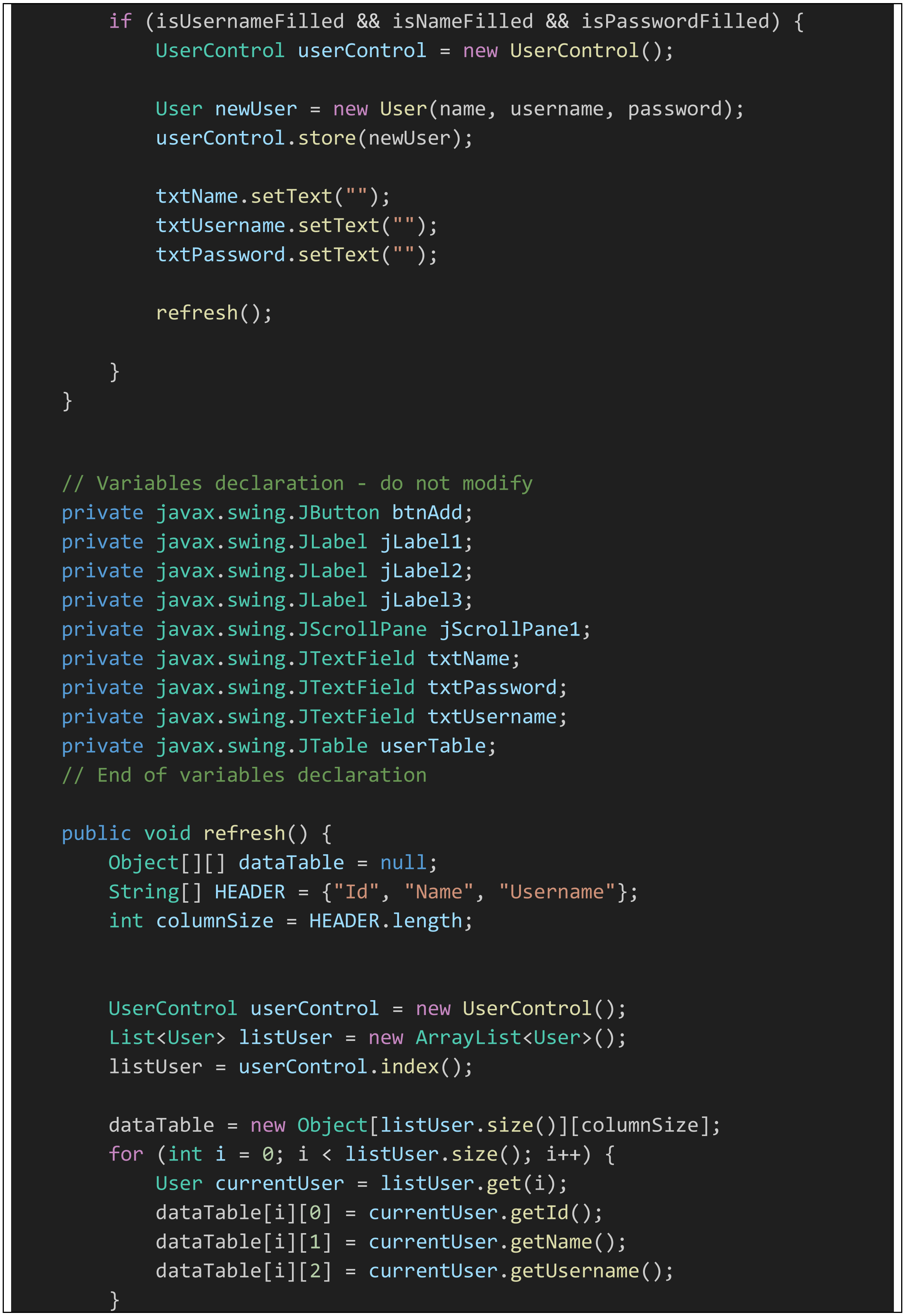
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to edit this template
 */
package Panel;

import Controllers.UserControl;
import Model.User;
import java.util.ArrayList;
import java.util.List;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author user
 */
public class pnlUser extends javax.swing.JPanel {

    /**
     * Creates new form pnlUser
     */
    public pnlUser() {
        initComponents();
        refresh();

                // agar background putih
        userTable.setOpaque(true);
        userTable.setFillsViewportHeight(true);

        userTable.setShowGrid(true);
    }

    /**
     * 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();
        userTable = new javax.swing.JTable();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        txtName = new javax.swing.JTextField();
        txtUsername = new javax.swing.JTextField();
        txtPassword = new javax.swing.JTextField();
        btnAdd = new javax.swing.JButton();

        userTable.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 [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        jScrollPane1.setViewportView(userTable);

        jLabel1.setText("Name");

        jLabel2.setText("Username");

        jLabel3.setText("Password");

        txtUsername.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtUsernameActionPerformed(evt);
            }
        });

        btnAdd.setText("ADD");
        btnAdd.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnAddMouseClicked(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
        this.setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 477, Short.MAX_VALUE)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.TRAILING)
                    .addComponent(jLabel3)
                    .addGroup(layout.createParallelGroup(javax.swing.Group
Layout.Alignment.LEADING)
                        .addComponent(jLabel1)
                        .addComponent(jLabel2)))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacemen
t.RELATED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING, false)
                    .addComponent(txtName)
                    .addComponent(txtPassword,
javax.swing.GroupLayout.DEFAULT_SIZE, 350, Short.MAX_VALUE)
                    .addComponent(txtUsername))
                .addGap(0, 0, Short.MAX_VALUE))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(btnAdd)
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addGap(26, 26, 26)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(txtName,
javax.swing.GroupLayout.PREFERRED_SIZE,
```



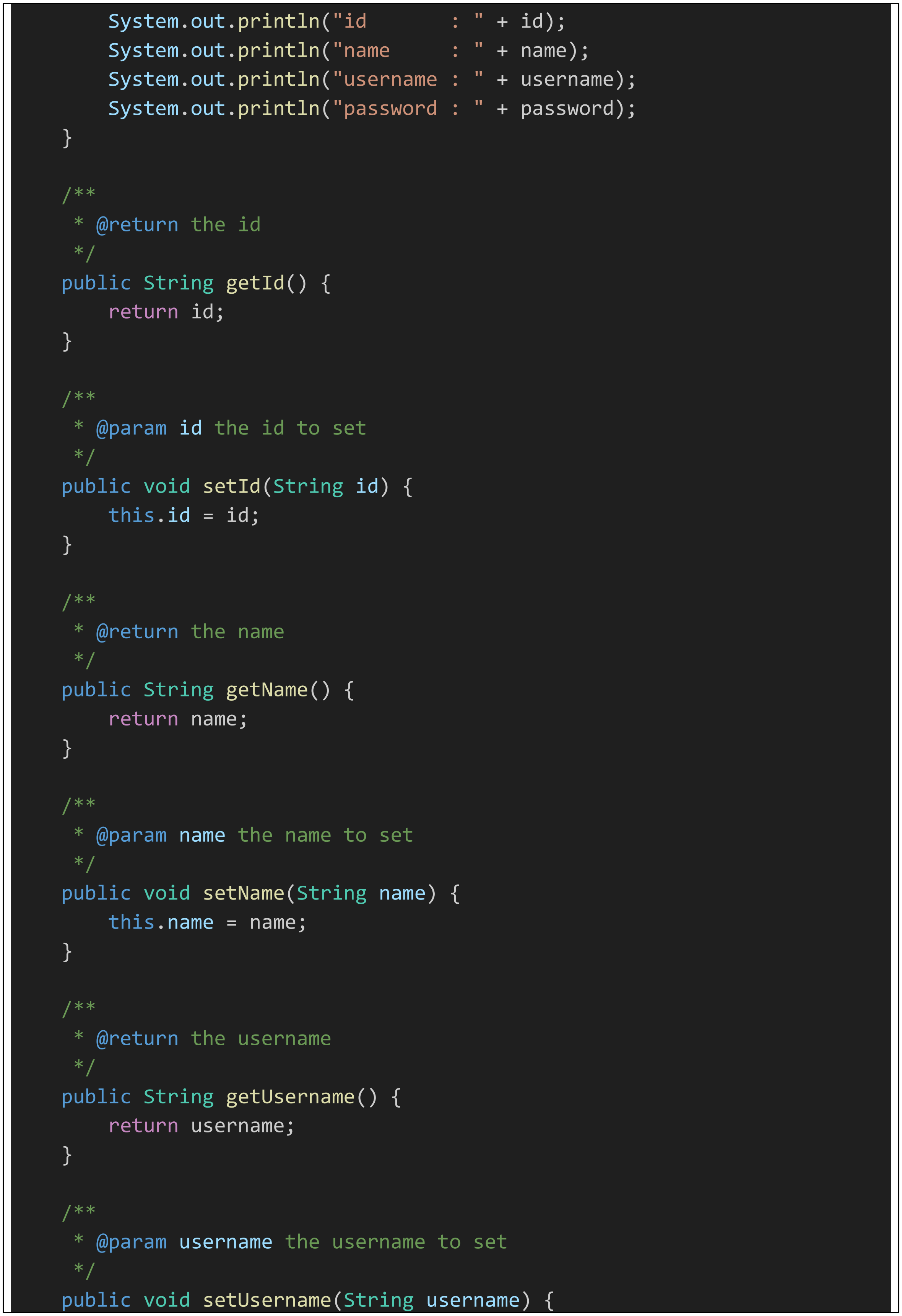
```
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(txtUsername,
javax.swing.GroupLayout.PREFERRED_SIZE, 15,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(txtPassword,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacemen
t.RELATED, 37, Short.MAX_VALUE)
                .addComponent(btnAdd)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacemen
t.UNRELATED)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );
    }// </editor-fold>

    private void txtUsernameActionPerformed(java.awt.event.ActionEvent
evt) {
        // TODO add your handling code here:
    }

    private void btnAddMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        String username = txtUsername.getText();
        String name = txtName.getText();
        String password = txtPassword.getText();

        boolean isUsernameFilled = !username.equals("");
        boolean isNameFilled = !name.equals("");
        boolean isPasswordFilled = !password.equals("");
```



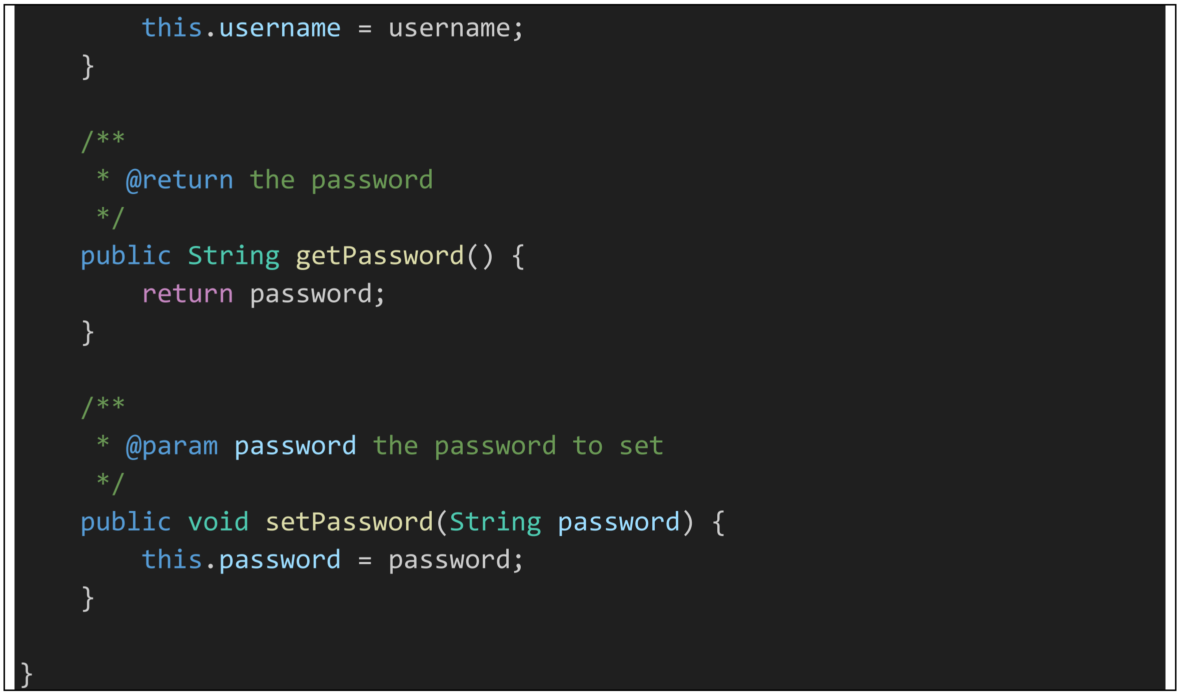
```
        if (isUsernameFilled && isNameFilled && isPasswordFilled) {
            UserControl userControl = new UserControl();

            User newUser = new User(name, username, password);
            userControl.store(newUser);

            txtName.setText("");
            txtUsername.setText("");
            txtPassword.setText("");

            refresh();

        }
    }


    // Variables declaration - do not modify
    private javax.swing.JButton btnAdd;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextField txtName;
    private javax.swing.JTextField txtPassword;
    private javax.swing.JTextField txtUsername;
    private javax.swing.JTable userTable;
    // End of variables declaration

    public void refresh() {
        Object[][] dataTable = null;
        String[] HEADER = {"Id", "Name", "Username"};
        int columnSize = HEADER.length;


        UserControl userControl = new UserControl();
        List<User> listUser = new ArrayList<User>();
        listUser = userControl.index();

        dataTable = new Object[listUser.size()][columnSize];
        for (int i = 0; i < listUser.size(); i++) {
            User currentUser = listUser.get(i);
            dataTable[i][0] = currentUser.getId();
            dataTable[i][1] = currentUser.getName();
            dataTable[i][2] = currentUser.getUsername();
        }
        userTable.setModel(new DefaultTableModel(dataTable, HEADER));
    }
}
```

- User,java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package Model;

import Generator.generator;

/**
 *
 * @author user
 */
public class User {
     private String id;
    private String name;
    private String username;
    private String password;

    public User(String id, String name, String username, String password)
{
        this.id = id;
        this.name = name;
        this.username = username;
        this.password = password;
    }

    public User(String name, String username, String password) {
        generator Generator = new generator();
        this.id = Generator.getId();
        this.name = name;
        this.username = username;
        this.password = password;
    }

    public void print() {
```

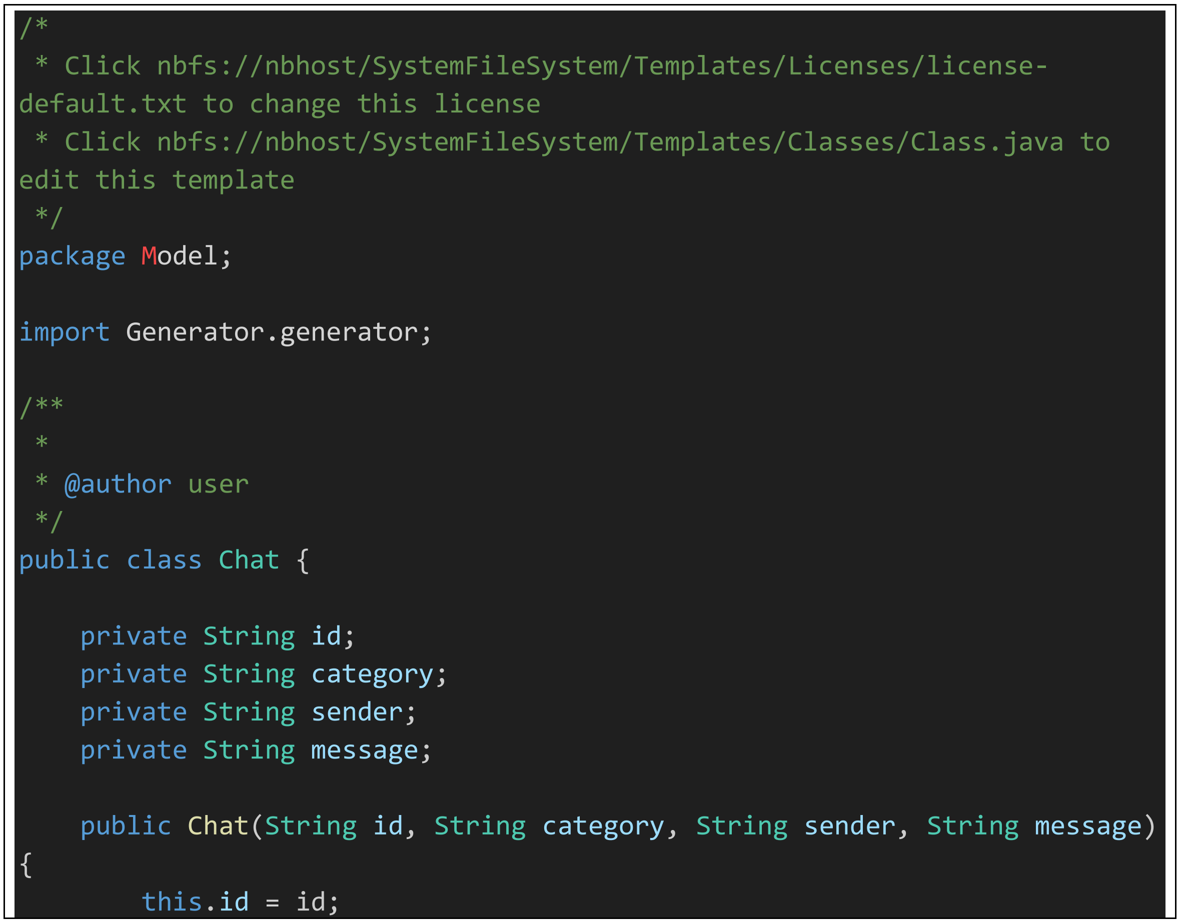


```
        System.out.println("id       : " + id);
        System.out.println("name     : " + name);
        System.out.println("username : " + username);
        System.out.println("password : " + password);
    }

    /**
     * @return the id
     */
    public String getId() {
        return id;
    }

    /**
     * @param id the id to set
     */
    public void setId(String id) {
        this.id = id;
    }

    /**
     * @return the name
     */
    public String getName() {
        return name;
    }

    /**
     * @param name the name to set
     */
    public void setName(String name) {
        this.name = name;
    }

    /**
     * @return the username
     */
    public String getUsername() {
        return username;
    }

    /**
     * @param username the username to set
     */
    public void setUsername(String username) {
```

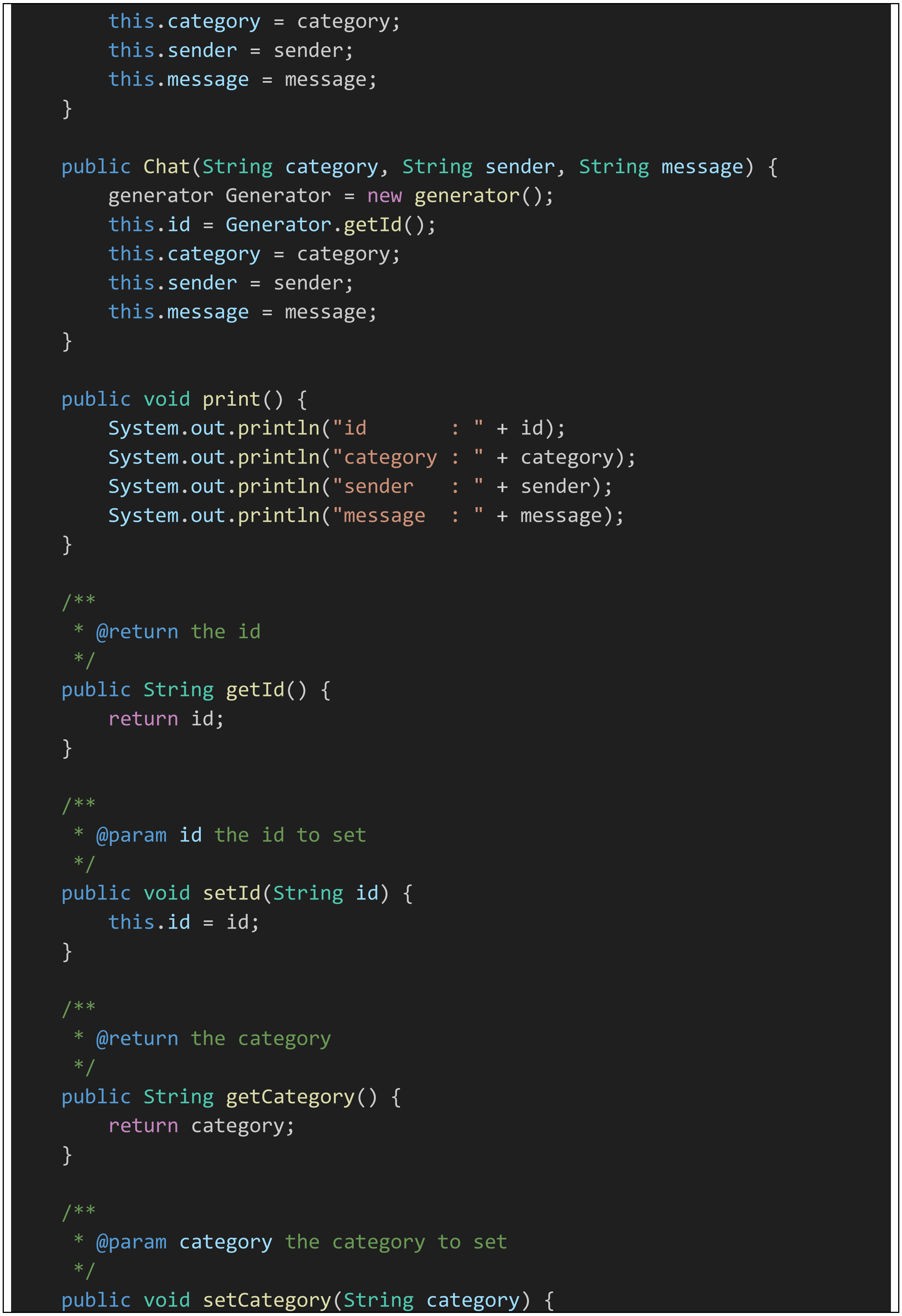


```java
        this.username = username;
    }

    /**
     * @return the password
     */
    public String getPassword() {
        return password;
    }

    /**
     * @param password the password to set
     */
    public void setPassword(String password) {
        this.password = password;
    }

}
```

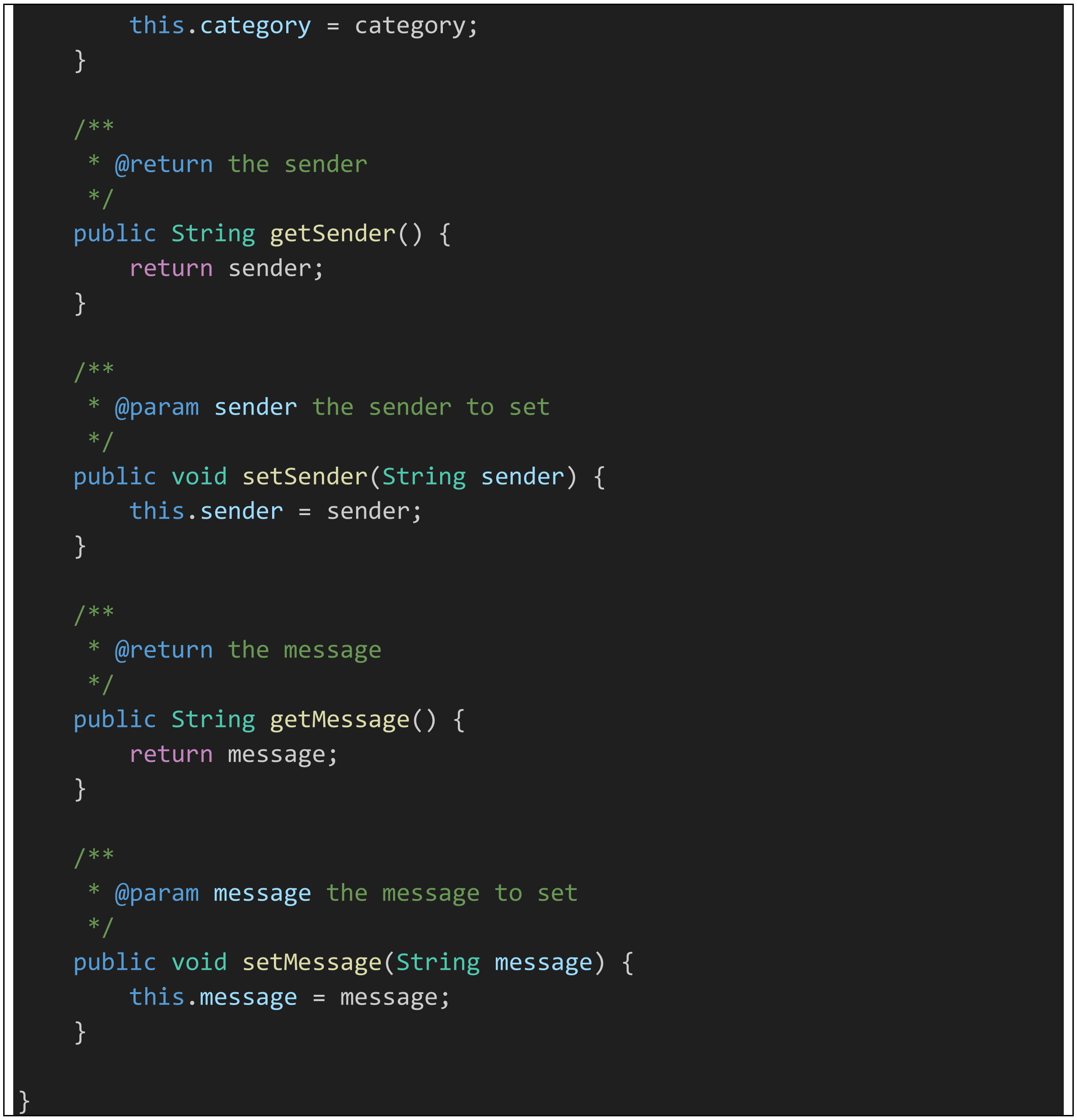
- Chat.java



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package Model;

import Generator.generator;

/**
 *
 * @author user
 */
public class Chat {

    private String id;
    private String category;
    private String sender;
    private String message;

    public Chat(String id, String category, String sender, String message)
{
        this.id = id;
```

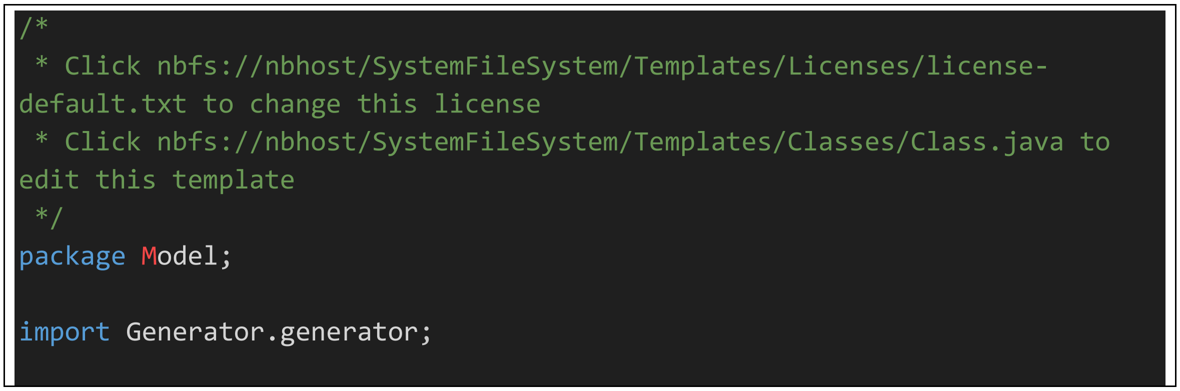


```
        this.category = category;
        this.sender = sender;
        this.message = message;
    }

    public Chat(String category, String sender, String message) {
        generator Generator = new generator();
        this.id = Generator.getId();
        this.category = category;
        this.sender = sender;
        this.message = message;
    }

    public void print() {
        System.out.println("id       : " + id);
        System.out.println("category : " + category);
        System.out.println("sender   : " + sender);
        System.out.println("message  : " + message);
    }

    /**
     * @return the id
     */
    public String getId() {
        return id;
    }

    /**
     * @param id the id to set
     */
    public void setId(String id) {
        this.id = id;
    }

    /**
     * @return the category
     */
    public String getCategory() {
        return category;
    }

    /**
     * @param category the category to set
     */
    public void setCategory(String category) {
```

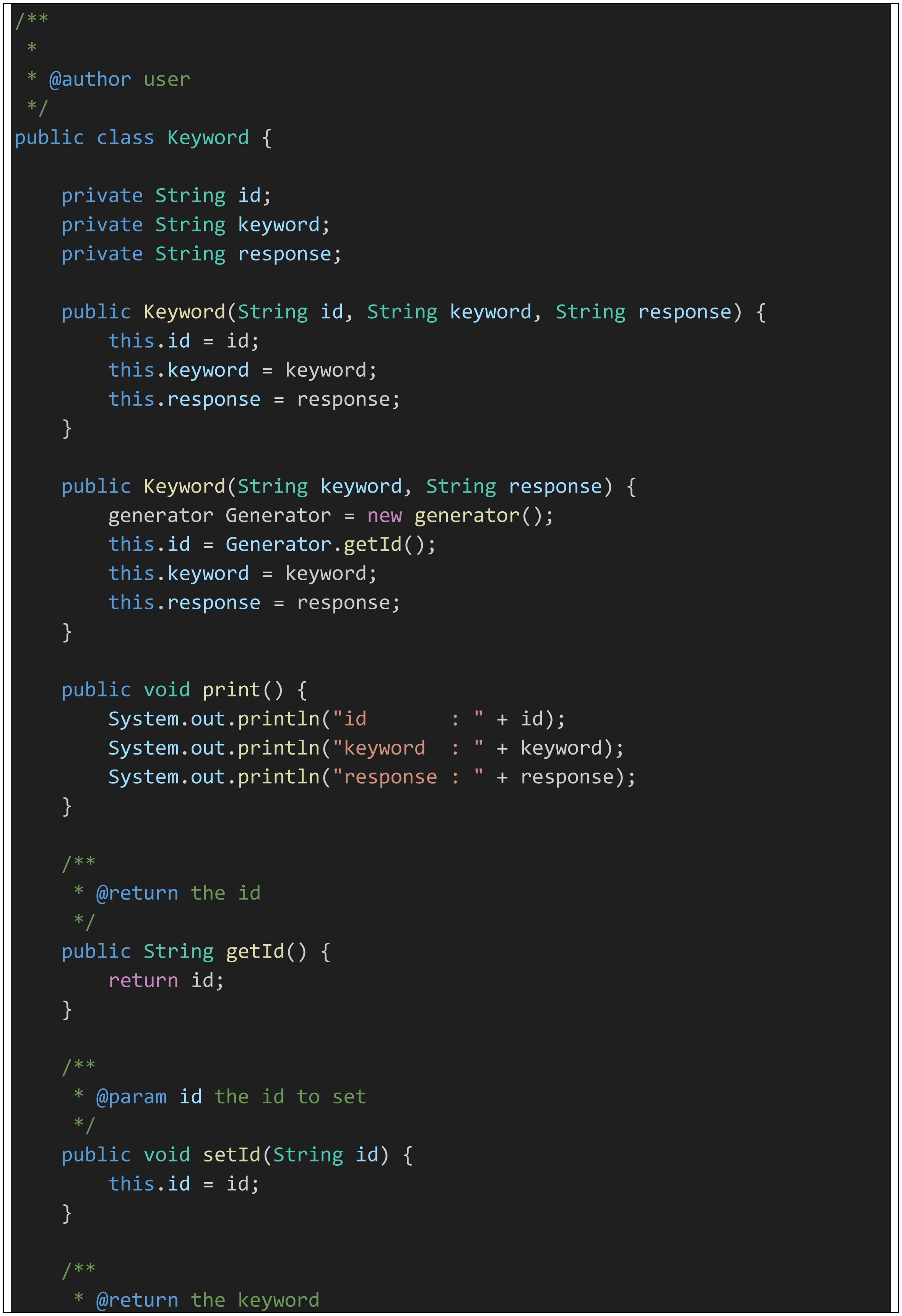


```
        this.category = category;
    }

    /**
     * @return the sender
     */
    public String getSender() {
        return sender;
    }

    /**
     * @param sender the sender to set
     */
    public void setSender(String sender) {
        this.sender = sender;
    }

    /**
     * @return the message
     */
    public String getMessage() {
        return message;
    }

    /**
     * @param message the message to set
     */
    public void setMessage(String message) {
        this.message = message;
    }

}
```

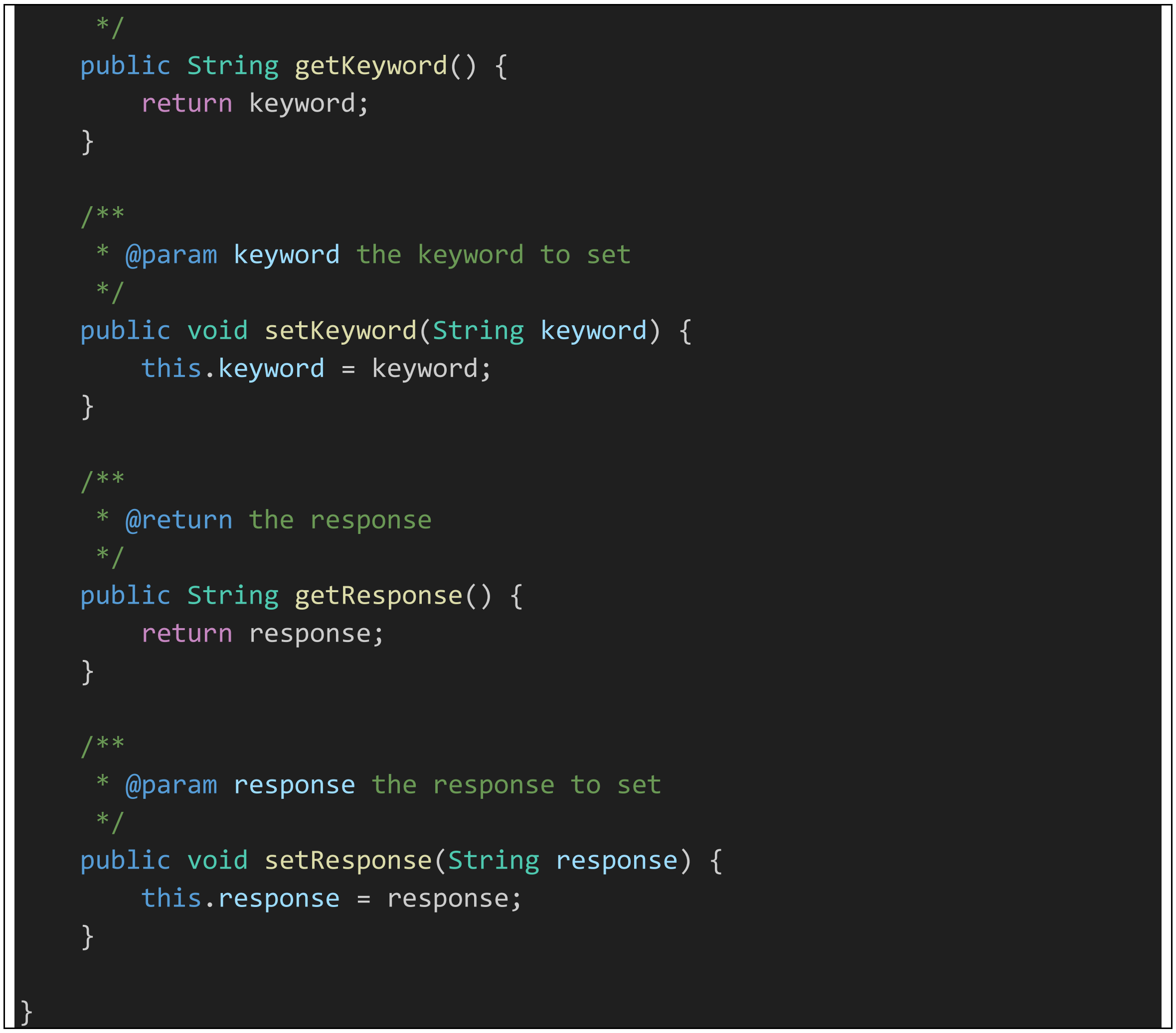
- Keyword.java



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package Model;

import Generator.generator;
```

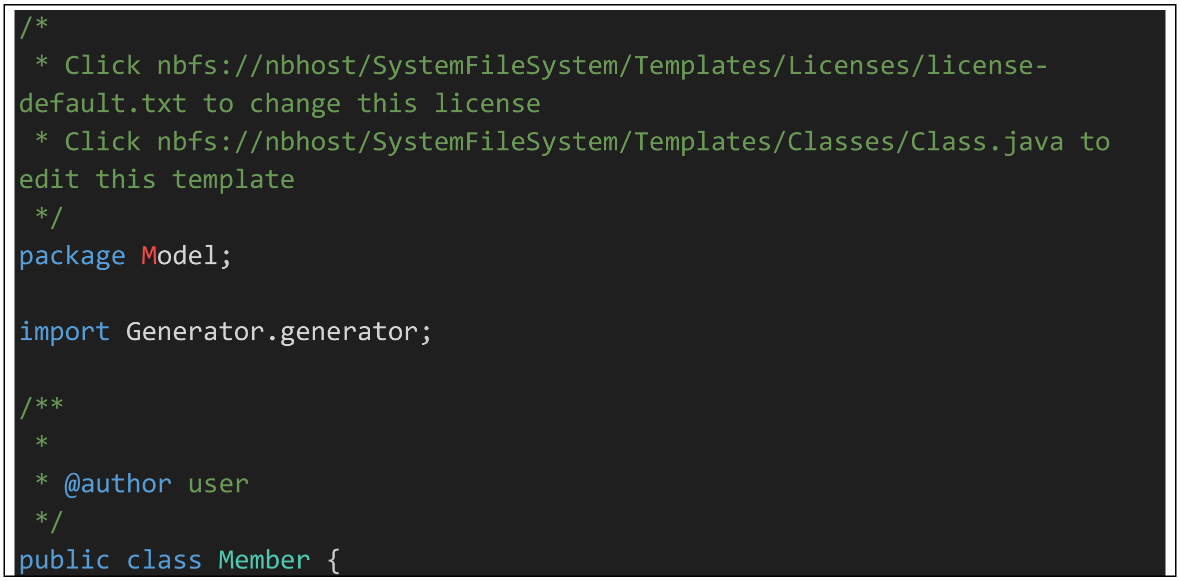


```
/**
 *
 * @author user
 */
public class Keyword {

    private String id;
    private String keyword;
    private String response;

    public Keyword(String id, String keyword, String response) {
        this.id = id;
        this.keyword = keyword;
        this.response = response;
    }

    public Keyword(String keyword, String response) {
        generator Generator = new generator();
        this.id = Generator.getId();
        this.keyword = keyword;
        this.response = response;
    }

    public void print() {
        System.out.println("id       : " + id);
        System.out.println("keyword  : " + keyword);
        System.out.println("response : " + response);
    }

    /**
     * @return the id
     */
    public String getId() {
        return id;
    }

    /**
     * @param id the id to set
     */
    public void setId(String id) {
        this.id = id;
    }

    /**
     * @return the keyword
```

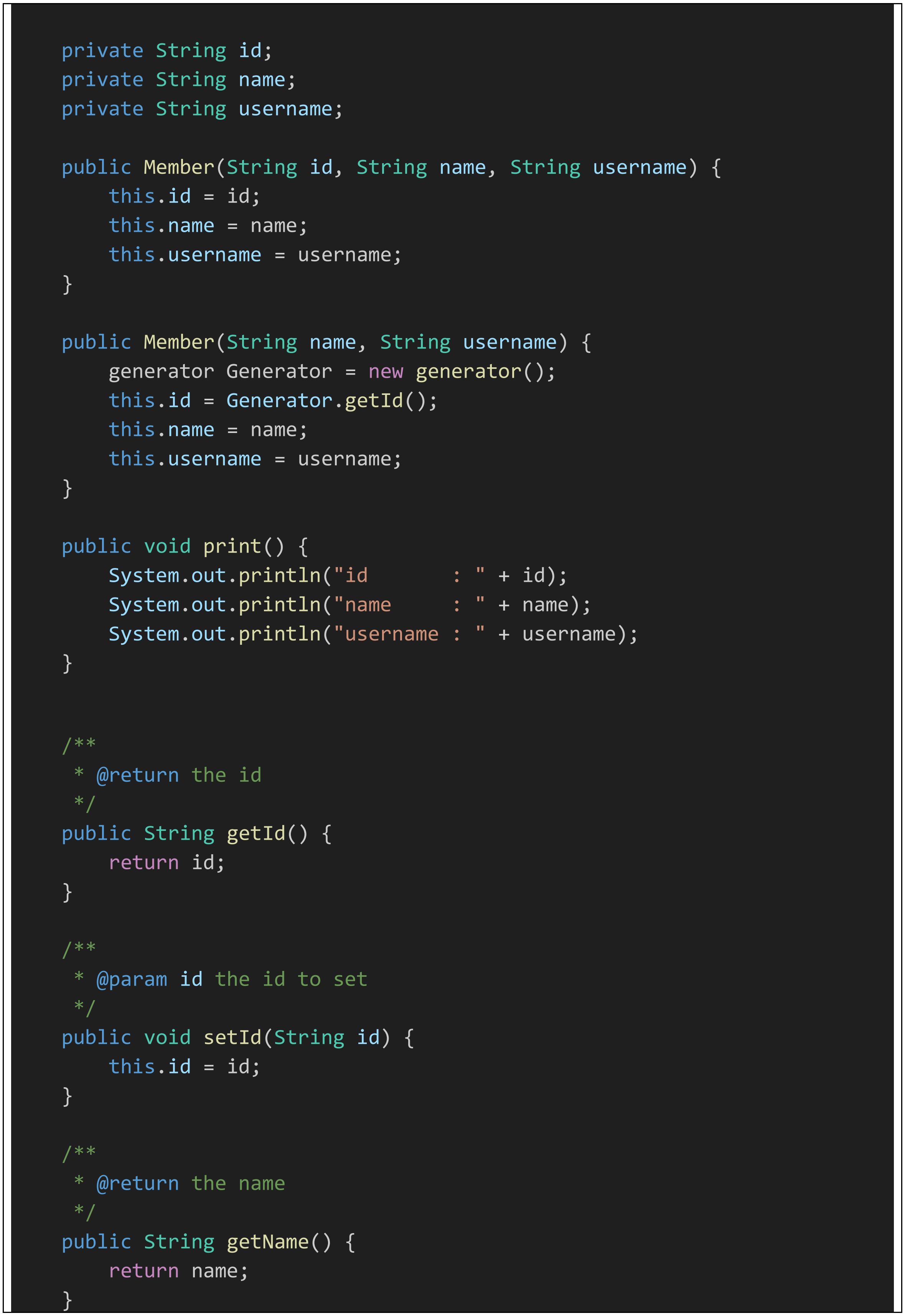


```
     */
    public String getKeyword() {
        return keyword;
    }

    /**
     * @param keyword the keyword to set
     */
    public void setKeyword(String keyword) {
        this.keyword = keyword;
    }

    /**
     * @return the response
     */
    public String getResponse() {
        return response;
    }

    /**
     * @param response the response to set
     */
    public void setResponse(String response) {
        this.response = response;
    }

}
```

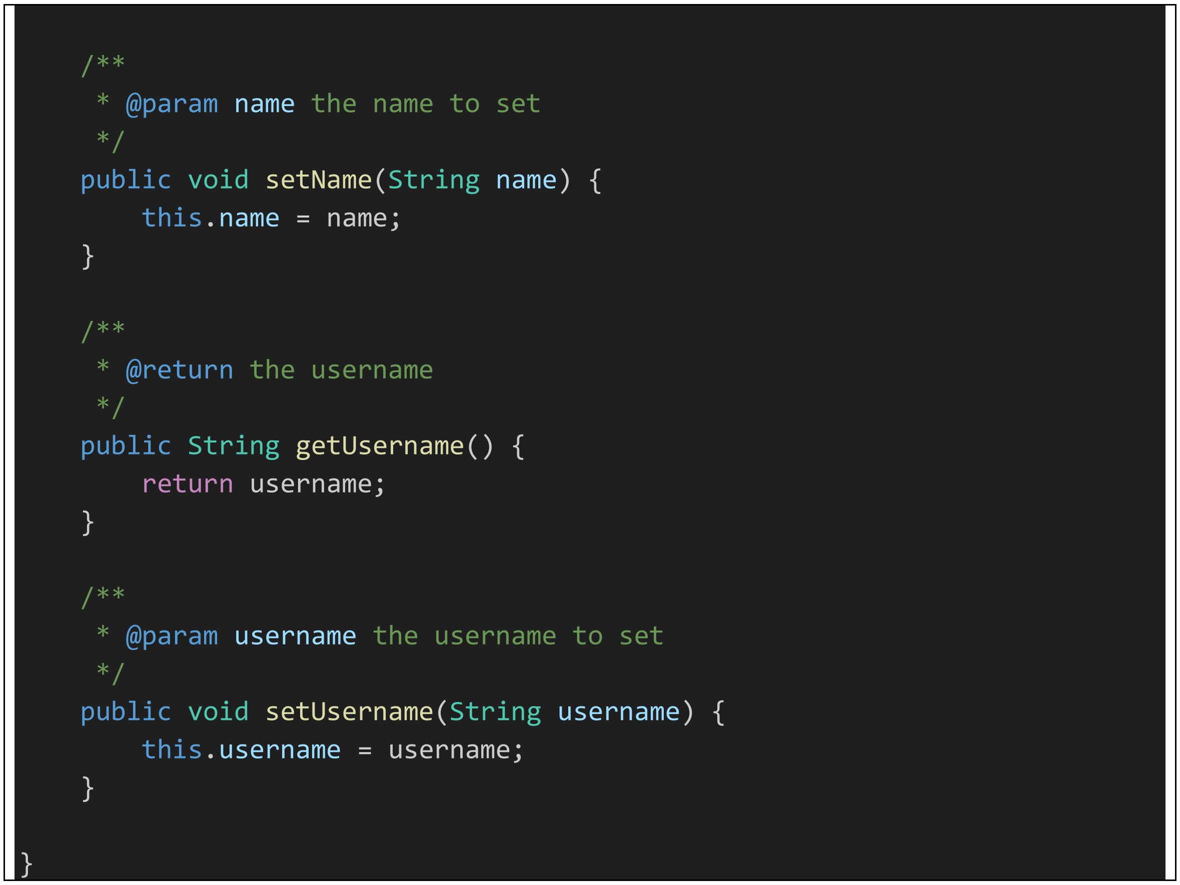
- Member.java



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package Model;

import Generator.generator;

/**
 *
 * @author user
 */
public class Member {
```

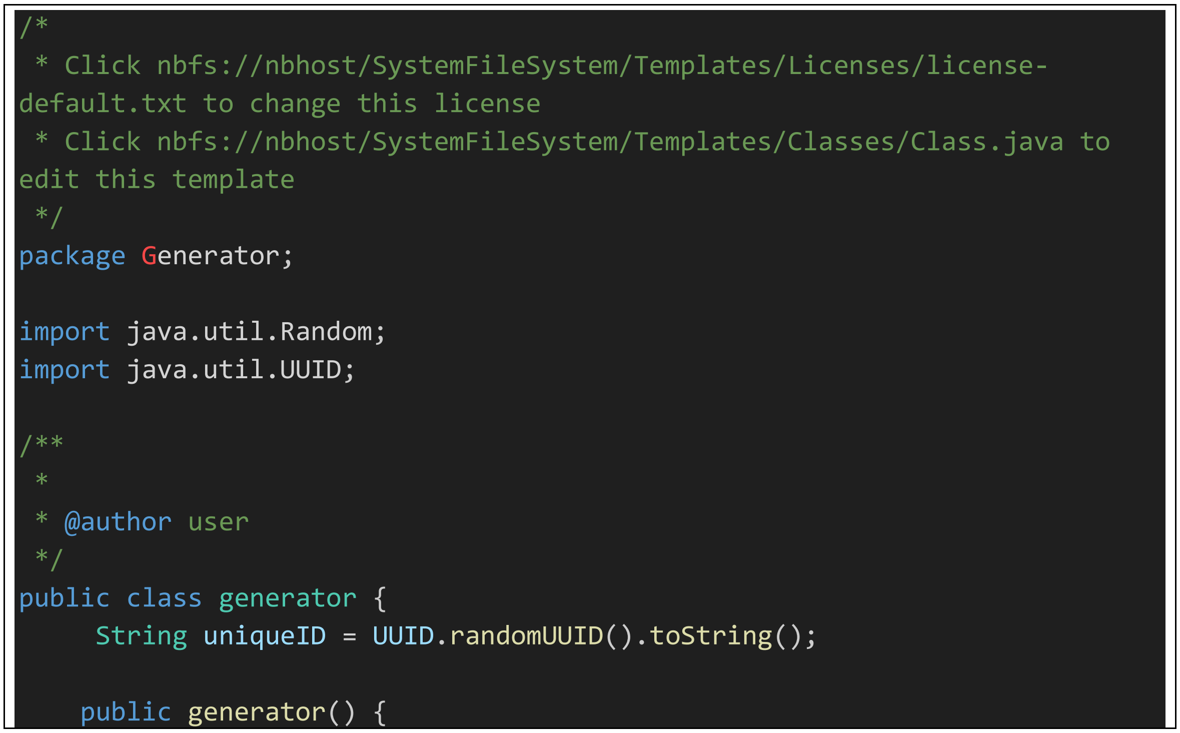


```java
    private String id;
    private String name;
    private String username;

    public Member(String id, String name, String username) {
        this.id = id;
        this.name = name;
        this.username = username;
    }

    public Member(String name, String username) {
        generator Generator = new generator();
        this.id = Generator.getId();
        this.name = name;
        this.username = username;
    }

    public void print() {
        System.out.println("id       : " + id);
        System.out.println("name     : " + name);
        System.out.println("username : " + username);
    }


    /**
     * @return the id
     */
    public String getId() {
        return id;
    }

    /**
     * @param id the id to set
     */
    public void setId(String id) {
        this.id = id;
    }

    /**
     * @return the name
     */
    public String getName() {
        return name;
    }
```



```
    /**
     * @param name the name to set
     */
    public void setName(String name) {
        this.name = name;
    }

    /**
     * @return the username
     */
    public String getUsername() {
        return username;
    }

    /**
     * @param username the username to set
     */
    public void setUsername(String username) {
        this.username = username;
    }

}
```

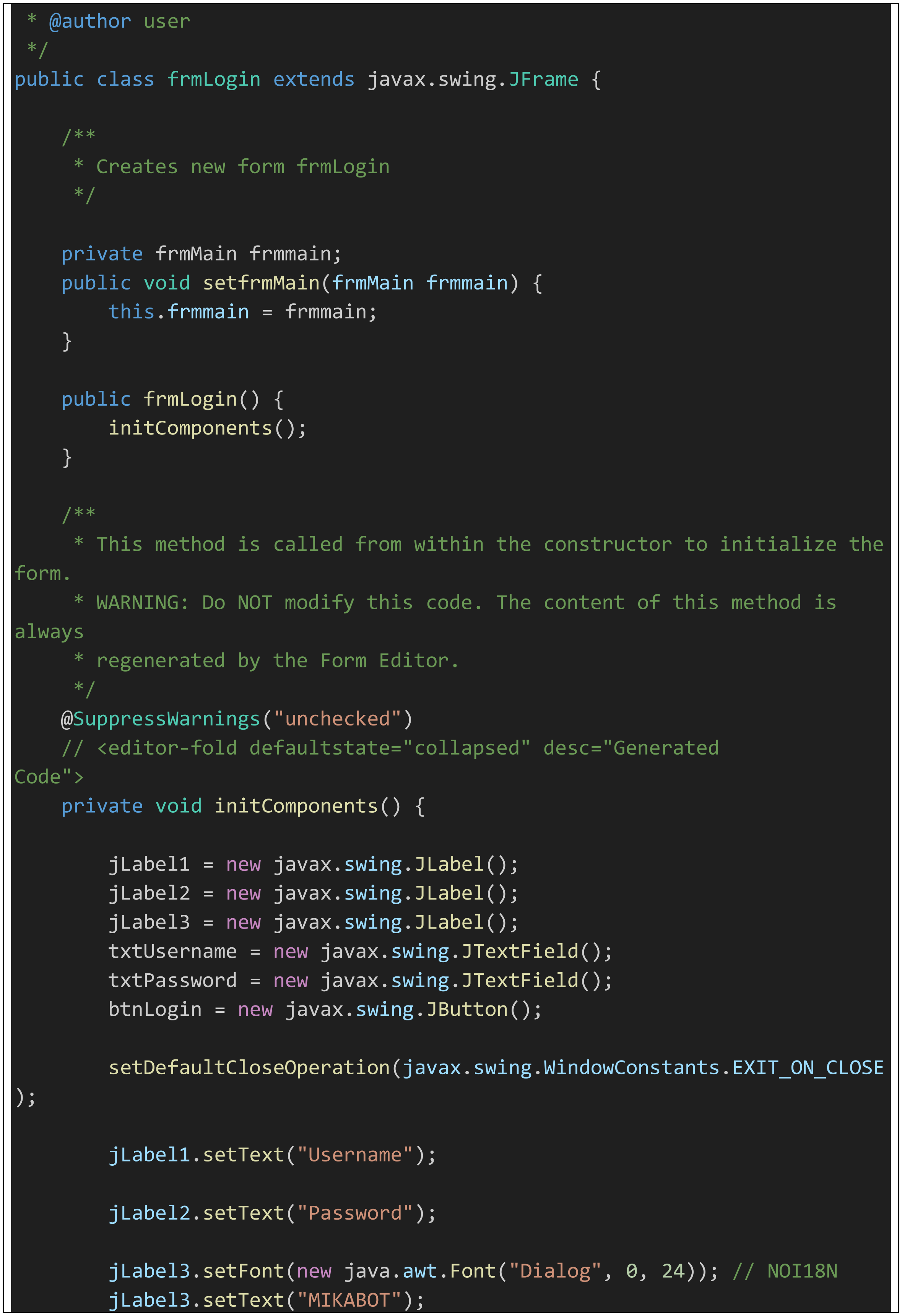
- Generator.java



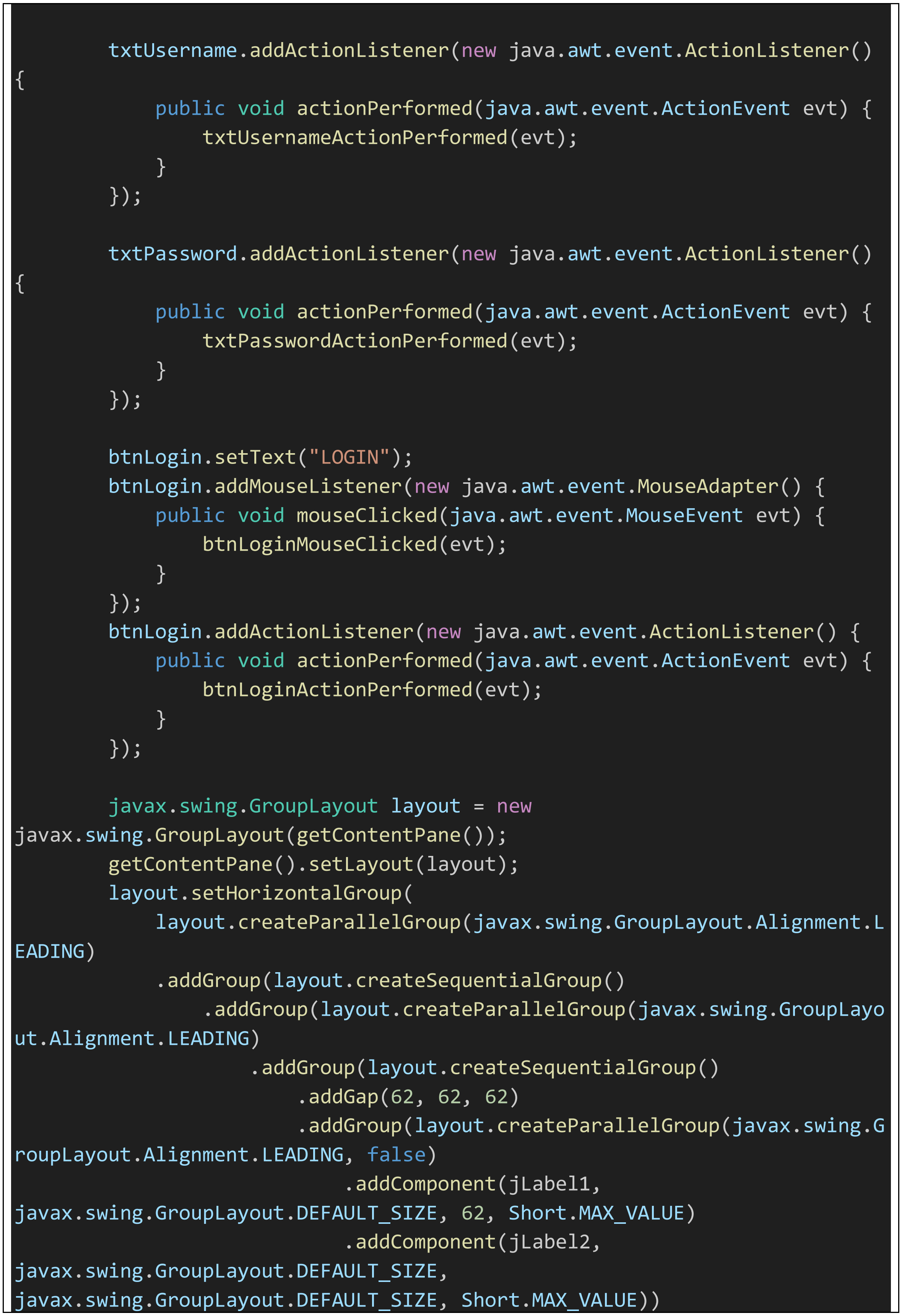
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package Generator;

import java.util.Random;
import java.util.UUID;

/**
 *
 * @author user
 */
public class generator {
     String uniqueID = UUID.randomUUID().toString();

    public generator() {
    }

    public String getId() {
        Random r = new Random();
        char a = (char)(r.nextInt(26) + 'a');
        char b = (char)(r.nextInt(26) + 'a');
        char c = (char) (r.nextInt(26) + 'a');
        char d = (char) (r.nextInt(26) + 'a');

        String randomString = "" + a + b + c;
        String randomChar = "" + d;

        String time = "" + System.currentTimeMillis();

        String id = randomString + time.substring(time.length() - 3, time.length()) + randomChar;

        return id;
    }

}
```

- frmLogin.java

```
/*
 * 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 Form;

import Controllers.UserControl;
import Model.User;
import com.sun.tools.javac.Main;
import com.wahyubagus.project.chatbot.MikaBot;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import org.telegram.telegrambots.meta.TelegramBotsApi;
import org.telegram.telegrambots.meta.exceptions.TelegramApiException;
import org.telegram.telegrambots.updatesreceivers.DefaultBotSession;

/**
 *
```

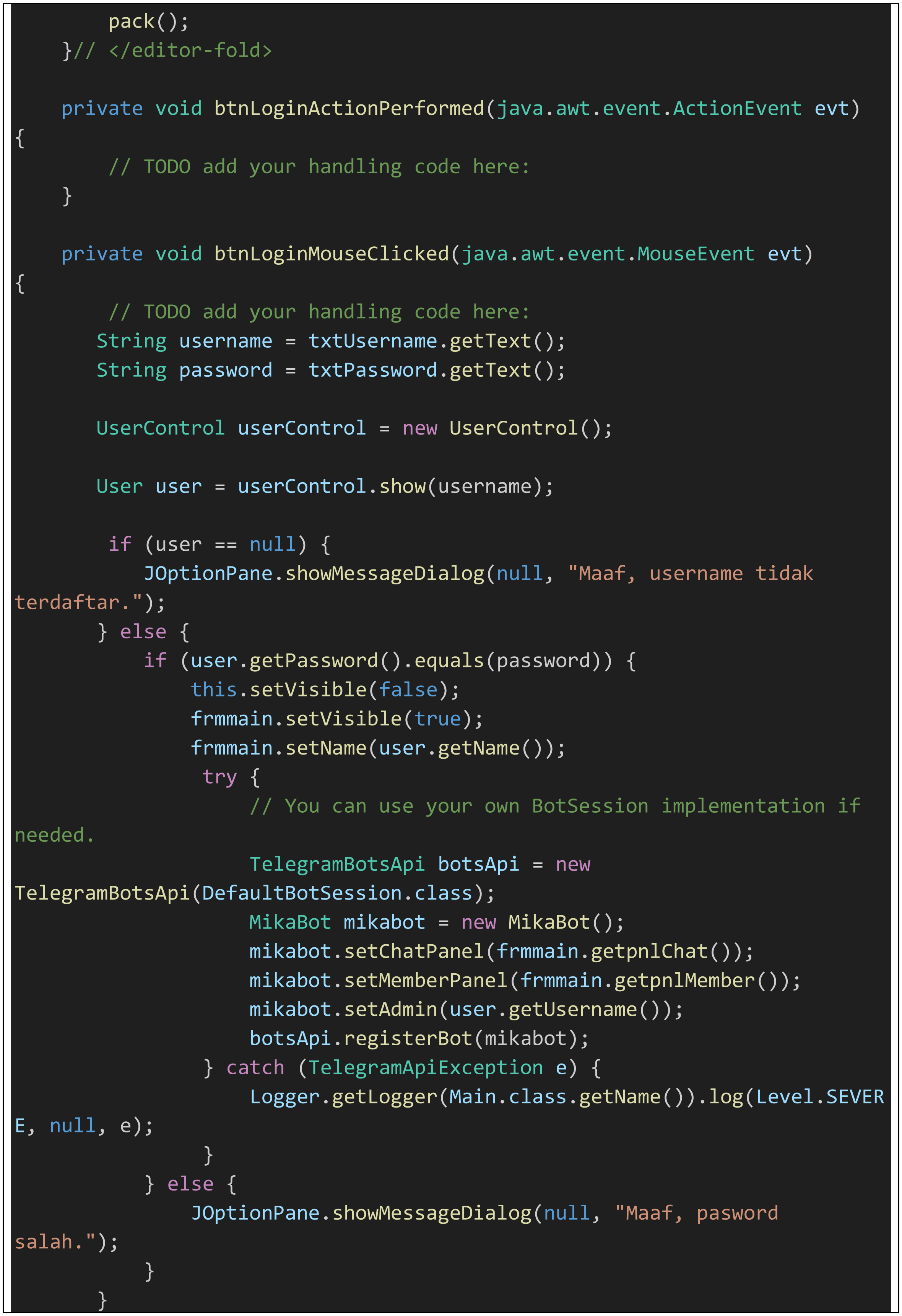


```
 * @author user
 */
public class frmLogin extends javax.swing.JFrame {

    /**
     * Creates new form frmLogin
     */

    private frmMain frmmain;
    public void setfrmMain(frmMain frmmain) {
        this.frmmain = frmmain;
    }

    public frmLogin() {
        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() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        txtUsername = new javax.swing.JTextField();
        txtPassword = new javax.swing.JTextField();
        btnLogin = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE
);

        jLabel1.setText("Username");

        jLabel2.setText("Password");

        jLabel3.setFont(new java.awt.Font("Dialog", 0, 24)); // NOI18N
        jLabel3.setText("MIKABOT");
```

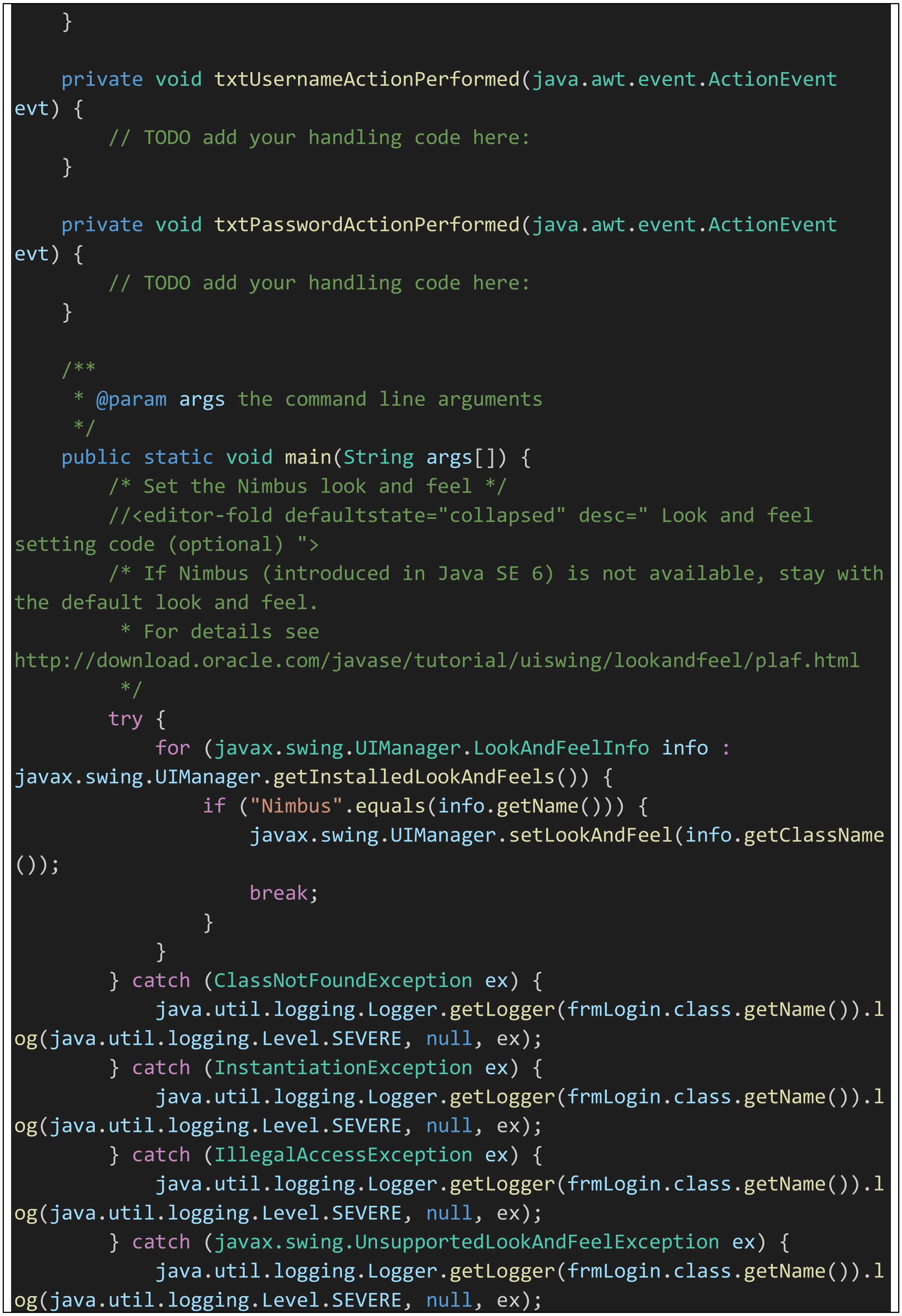


```
        txtUsername.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtUsernameActionPerformed(evt);
            }
        });

        txtPassword.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtPasswordActionPerformed(evt);
            }
        });

        btnLogin.setText("LOGIN");
        btnLogin.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnLoginMouseClicked(evt);
            }
        });
        btnLogin.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnLoginActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(62, 62, 62)
                        .addGroup(layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.LEADING, false)
                            .addComponent(jLabel1,
javax.swing.GroupLayout.DEFAULT_SIZE, 62, Short.MAX_VALUE)
                            .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
```

```
                        .addGap(33, 33, 33)
                        .addGroup(layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.LEADING)
                            .addComponent(txtPassword)
                            .addComponent(txtUsername)))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(200, 200, 200)
                        .addComponent(jLabel3)))
                .addGap(197, 197, 197))
            .addGroup(layout.createSequentialGroup()
                .addGap(209, 209, 209)
                .addComponent(btnLogin,
javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(27, 27, 27)
                .addComponent(jLabel3)
                .addGap(50, 50, 50)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(txtUsername,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacemen
t.UNRELATED)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(txtPassword,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(51, 51, 51)
                .addComponent(btnLogin)
                .addContainerGap(95, Short.MAX_VALUE))
        );
```



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

    private void btnLoginActionPerformed(java.awt.event.ActionEvent evt)
{
        // TODO add your handling code here:
    }

    private void btnLoginMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
       String username = txtUsername.getText();
       String password = txtPassword.getText();

       UserControl userControl = new UserControl();

       User user = userControl.show(username);

        if (user == null) {
           JOptionPane.showMessageDialog(null, "Maaf, username tidak
terdaftar.");
       } else {
           if (user.getPassword().equals(password)) {
               this.setVisible(false);
               frmmain.setVisible(true);
               frmmain.setName(user.getName());
                try {
                    // You can use your own BotSession implementation if
needed.
                    TelegramBotsApi botsApi = new
TelegramBotsApi(DefaultBotSession.class);
                    MikaBot mikabot = new MikaBot();
                    mikabot.setChatPanel(frmmain.getpnlChat());
                    mikabot.setMemberPanel(frmmain.getpnlMember());
                    mikabot.setAdmin(user.getUsername());
                    botsApi.registerBot(mikabot);
                } catch (TelegramApiException e) {
                    Logger.getLogger(Main.class.getName()).log(Level.SEVER
E, null, e);
                }
           } else {
               JOptionPane.showMessageDialog(null, "Maaf, pasword
salah.");
           }
       }
```

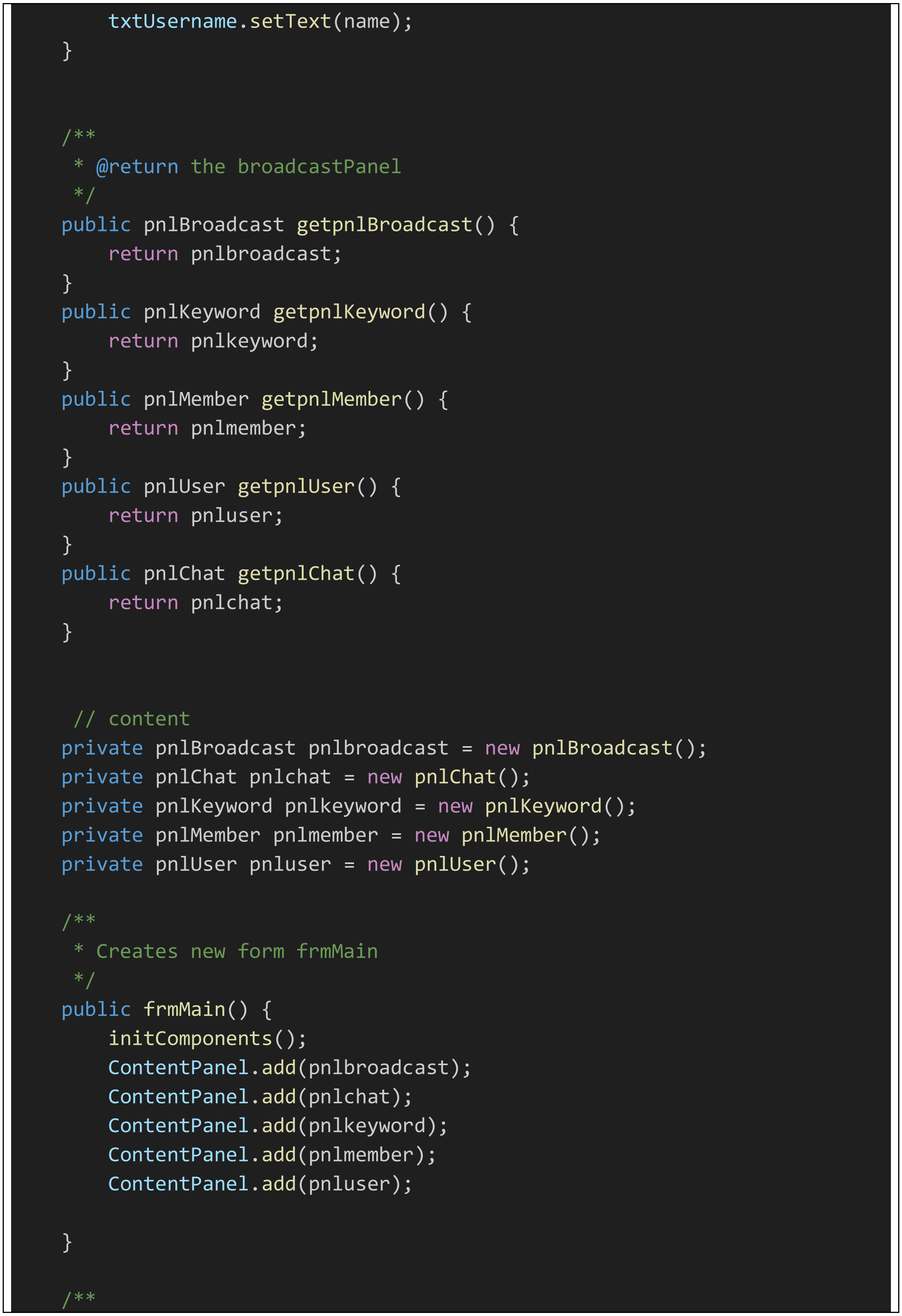


```
    }

    private void txtUsernameActionPerformed(java.awt.event.ActionEvent
evt) {
        // TODO add your handling code here:
    }

    private void txtPasswordActionPerformed(java.awt.event.ActionEvent
evt) {
        // TODO add your handling code here:
    }

    /**
     * @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(frmLogin.class.getName()).l
og(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(frmLogin.class.getName()).l
og(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(frmLogin.class.getName()).l
og(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(frmLogin.class.getName()).l
og(java.util.logging.Level.SEVERE, null, ex);
```

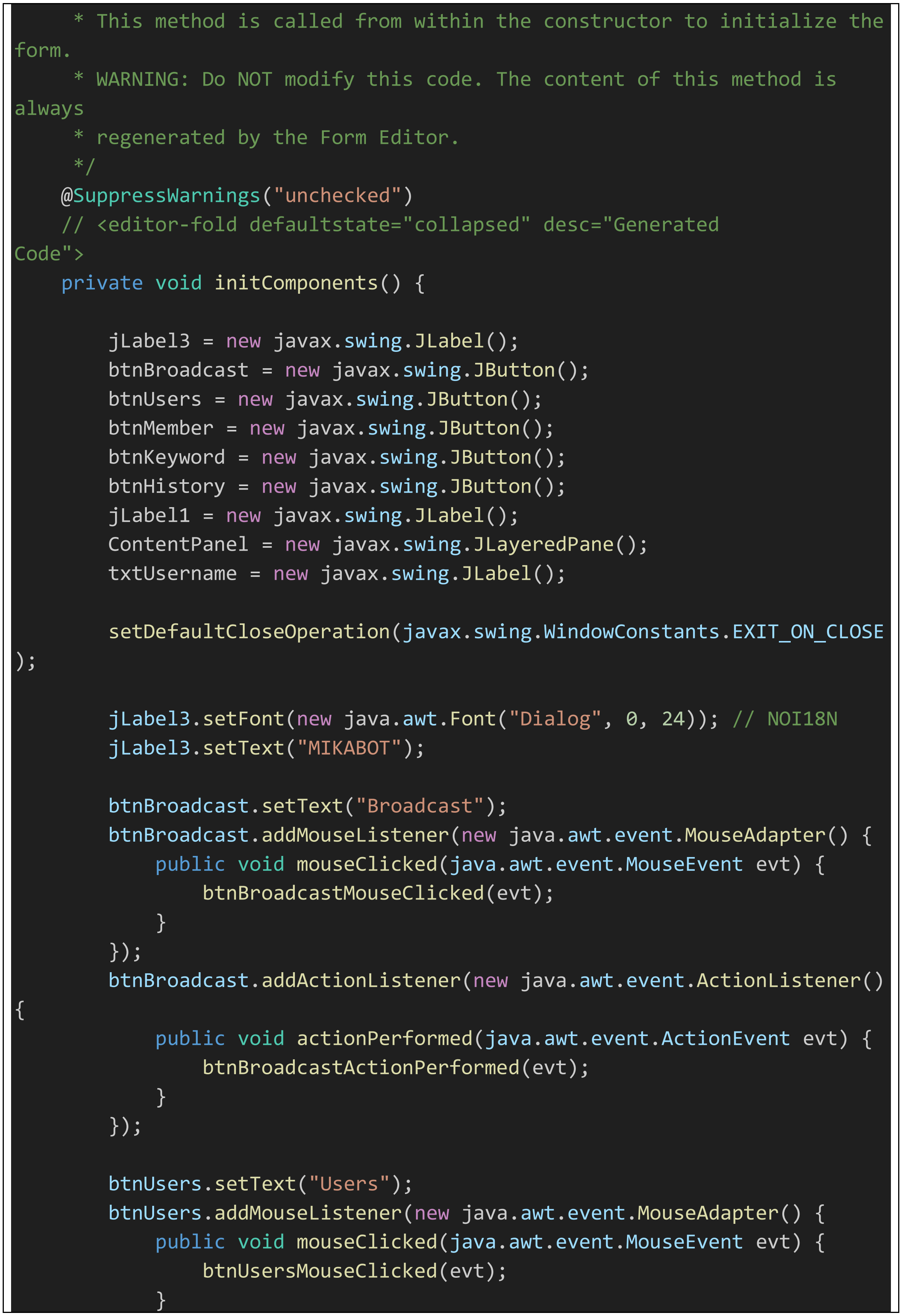
```
        }
        //</editor-fold>


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

    // Variables declaration - do not modify
    private javax.swing.JButton btnLogin;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JTextField txtPassword;
    private javax.swing.JTextField txtUsername;
    // End of variables declaration
}
```

- frmMain.java

```
/*
 * 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 Form;

import Panel.pnlBroadcast;
import Panel.pnlChat;
import Panel.pnlKeyword;
import Panel.pnlMember;
import Panel.pnlUser;


/**
 *
 * @author user
 */
public class frmMain extends javax.swing.JFrame {
    public void setName(String name) {
```

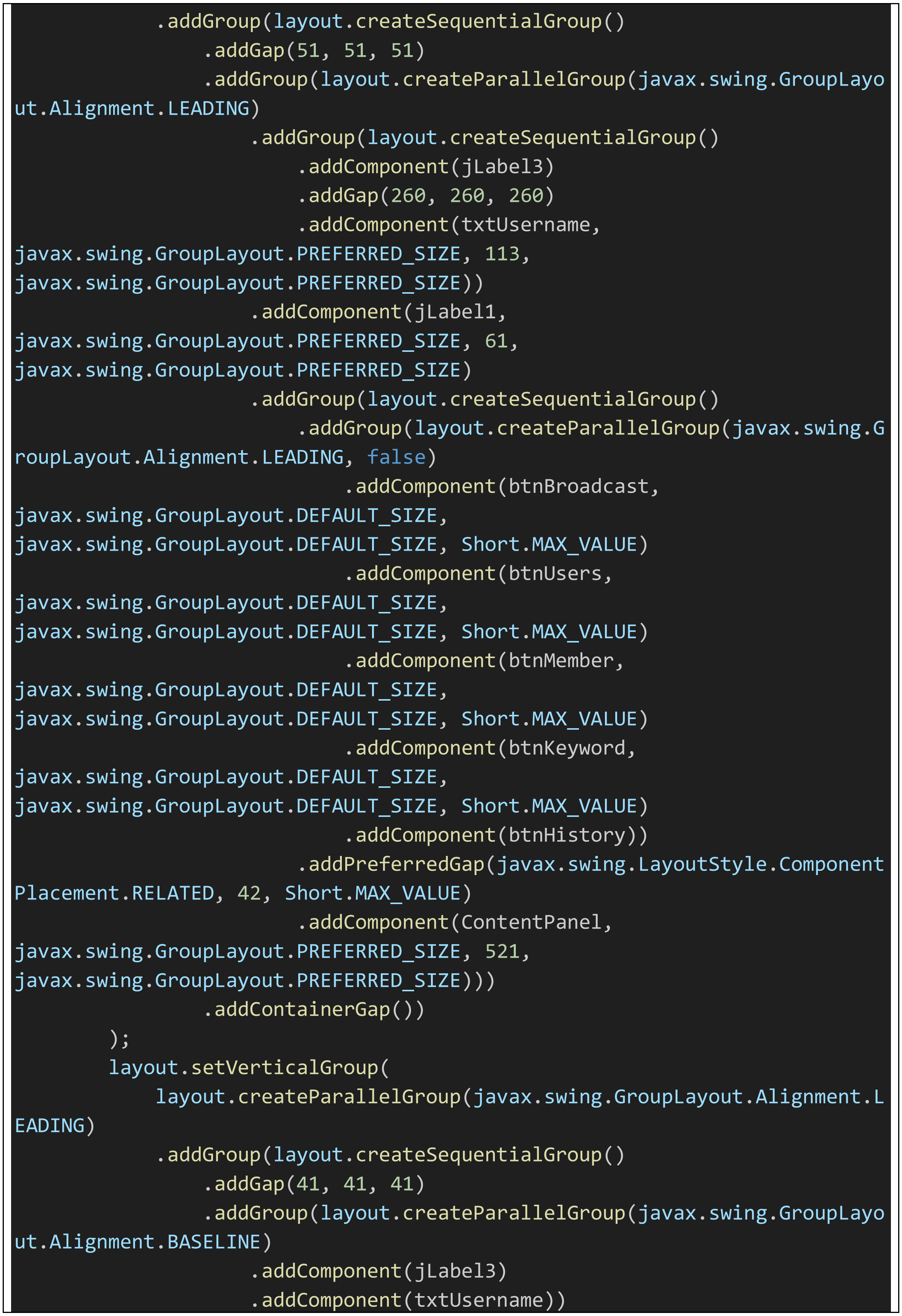


```
        txtUsername.setText(name);
    }


    /**
     * @return the broadcastPanel
     */
    public pnlBroadcast getpnlBroadcast() {
        return pnlbroadcast;
    }
    public pnlKeyword getpnlKeyword() {
        return pnlkeyword;
    }
    public pnlMember getpnlMember() {
        return pnlmember;
    }
    public pnlUser getpnlUser() {
        return pnluser;
    }
    public pnlChat getpnlChat() {
        return pnlchat;
    }


     // content
    private pnlBroadcast pnlbroadcast = new pnlBroadcast();
    private pnlChat pnlchat = new pnlChat();
    private pnlKeyword pnlkeyword = new pnlKeyword();
    private pnlMember pnlmember = new pnlMember();
    private pnlUser pnluser = new pnlUser();

    /**
     * Creates new form frmMain
     */
    public frmMain() {
        initComponents();
        ContentPanel.add(pnlbroadcast);
        ContentPanel.add(pnlchat);
        ContentPanel.add(pnlkeyword);
        ContentPanel.add(pnlmember);
        ContentPanel.add(pnluser);

    }

    /**
```

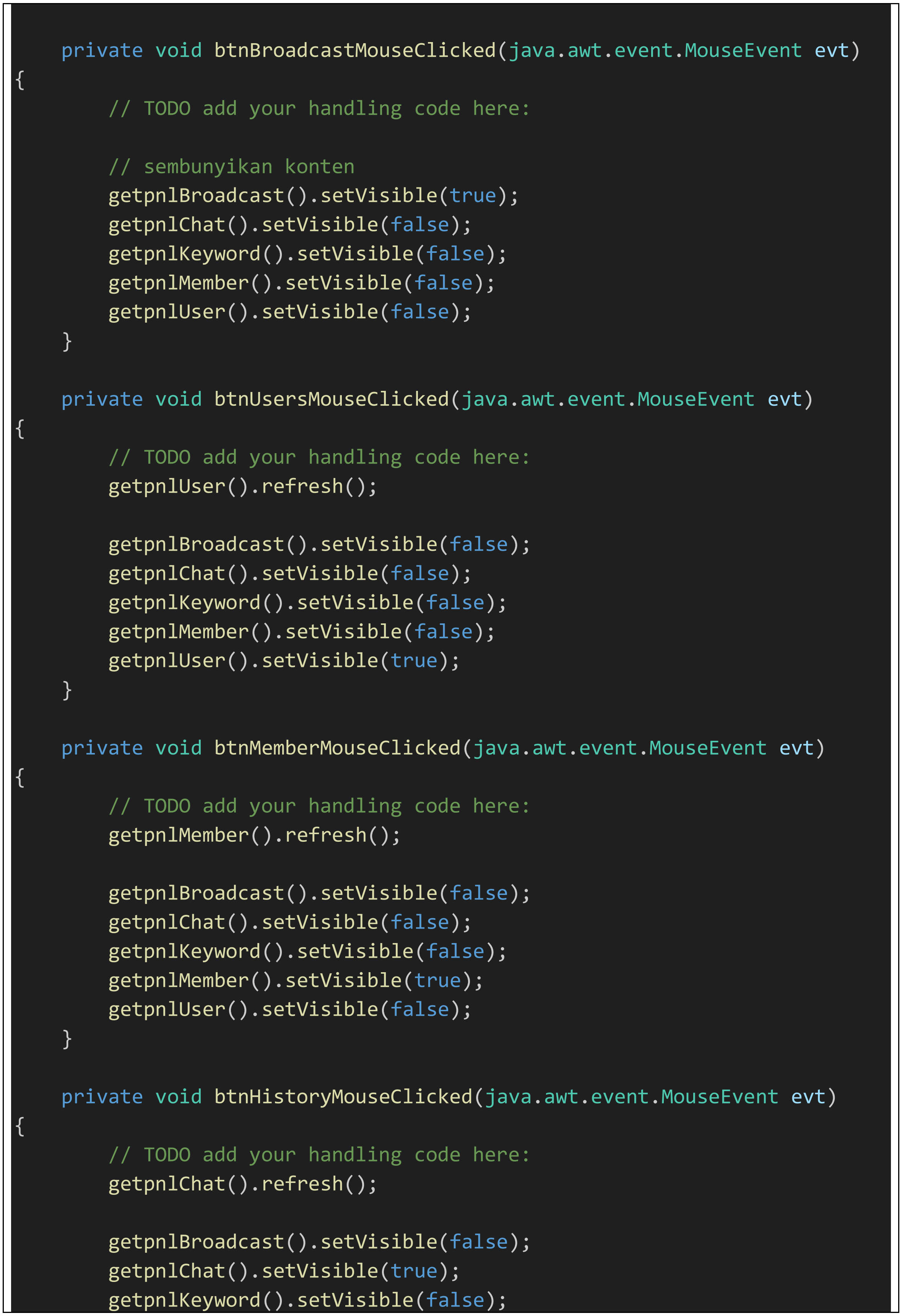


```
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated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

        jLabel3 = new javax.swing.JLabel();
        btnBroadcast = new javax.swing.JButton();
        btnUsers = new javax.swing.JButton();
        btnMember = new javax.swing.JButton();
        btnKeyword = new javax.swing.JButton();
        btnHistory = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        ContentPanel = new javax.swing.JLayeredPane();
        txtUsername = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE
);

        jLabel3.setFont(new java.awt.Font("Dialog", 0, 24)); // NOI18N
        jLabel3.setText("MIKABOT");

        btnBroadcast.setText("Broadcast");
        btnBroadcast.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnBroadcastMouseClicked(evt);
            }
        });
        btnBroadcast.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnBroadcastActionPerformed(evt);
            }
        });

        btnUsers.setText("Users");
        btnUsers.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnUsersMouseClicked(evt);
            }
```
]]>

```
        });

        btnMember.setText("Member");
        btnMember.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnMemberMouseClicked(evt);
            }
        });

        btnKeyword.setText("Keyword");
        btnKeyword.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnKeywordMouseClicked(evt);
            }
        });
        btnKeyword.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnKeywordActionPerformed(evt);
            }
        });

        btnHistory.setText("History Chat");
        btnHistory.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                btnHistoryMouseClicked(evt);
            }
        });

        jLabel1.setText("Log out");
        jLabel1.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jLabel1MouseClicked(evt);
            }
        });

        ContentPanel.setLayout(new java.awt.CardLayout());

        txtUsername.setText("username");

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
```



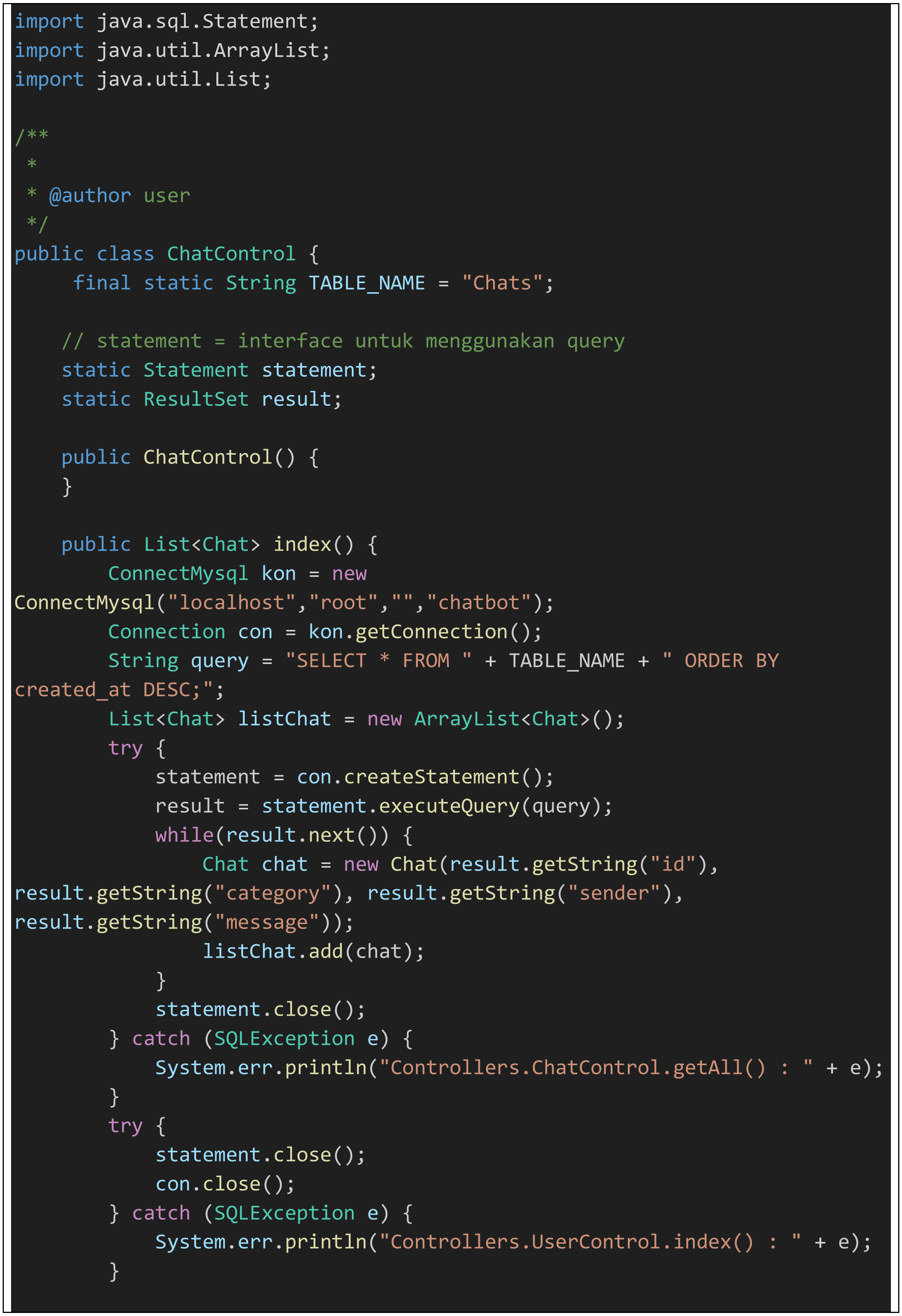
```
            .addGroup(layout.createSequentialGroup()
                .addGap(51, 51, 51)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel3)
                        .addGap(260, 260, 260)
                        .addComponent(txtUsername,
javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.G
roupLayout.Alignment.LEADING, false)
                            .addComponent(btnBroadcast,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(btnUsers,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(btnMember,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(btnKeyword,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(btnHistory))
                        .addPreferredGap(javax.swing.LayoutStyle.Component
Placement.RELATED, 42, Short.MAX_VALUE)
                        .addComponent(ContentPanel,
javax.swing.GroupLayout.PREFERRED_SIZE, 521,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(41, 41, 41)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayo
ut.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(txtUsername))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(btnBroadcast)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(btnUsers)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(btnMember)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(btnKeyword)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(btnHistory))
                    .addComponent(ContentPanel, javax.swing.GroupLayout.PREFERRED_SIZE, 323, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 12, Short.MAX_VALUE)
                .addComponent(jLabel1)
                .addContainerGap())
        );

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

    private void btnBroadcastActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void btnKeywordActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }
```



```
    private void btnBroadcastMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:

        // sembunyikan konten
        getpnlBroadcast().setVisible(true);
        getpnlChat().setVisible(false);
        getpnlKeyword().setVisible(false);
        getpnlMember().setVisible(false);
        getpnlUser().setVisible(false);
    }

    private void btnUsersMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        getpnlUser().refresh();

        getpnlBroadcast().setVisible(false);
        getpnlChat().setVisible(false);
        getpnlKeyword().setVisible(false);
        getpnlMember().setVisible(false);
        getpnlUser().setVisible(true);
    }

    private void btnMemberMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        getpnlMember().refresh();

        getpnlBroadcast().setVisible(false);
        getpnlChat().setVisible(false);
        getpnlKeyword().setVisible(false);
        getpnlMember().setVisible(true);
        getpnlUser().setVisible(false);
    }

    private void btnHistoryMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        getpnlChat().refresh();

        getpnlBroadcast().setVisible(false);
        getpnlChat().setVisible(true);
        getpnlKeyword().setVisible(false);
        getpnlMember().setVisible(false);
        getpnlUser().setVisible(false);

    }

    private void btnKeywordMouseClicked(java.awt.event.MouseEvent evt)
{
        // TODO add your handling code here:
        getpnlKeyword().refresh();

        getpnlBroadcast().setVisible(false);
        getpnlChat().setVisible(false);
        getpnlKeyword().setVisible(true);
        getpnlMember().setVisible(false);
        getpnlUser().setVisible(false);
    }

    /**
     * @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(frmMain.class.getName()).lo
g(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(frmMain.class.getName()).lo
g(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(frmMain.class.getName()).lo
g(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(frmMain.class.getName()).lo
g(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>


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

    // Variables declaration - do not modify
    private javax.swing.JLayeredPane ContentPanel;
    private javax.swing.JButton btnBroadcast;
    private javax.swing.JButton btnHistory;
    private javax.swing.JButton btnKeyword;
    private javax.swing.JButton btnMember;
    private javax.swing.JButton btnUsers;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel txtUsername;
    // End of variables declaration
}
```

- ChatControl.java

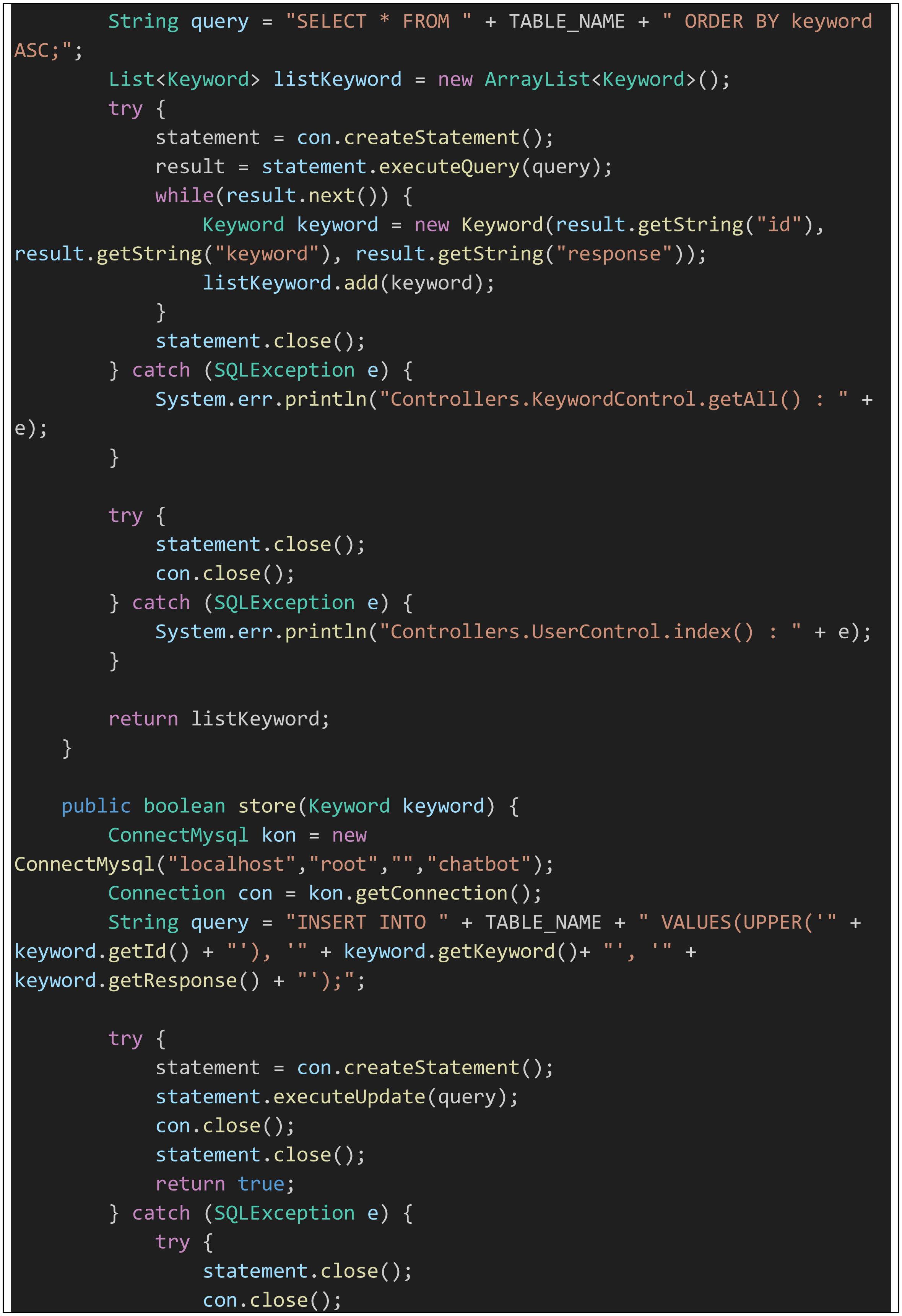
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package Controllers;

import Model.Chat;
import com.wahyubagus.project.chatbot.ConnectMysql;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
```



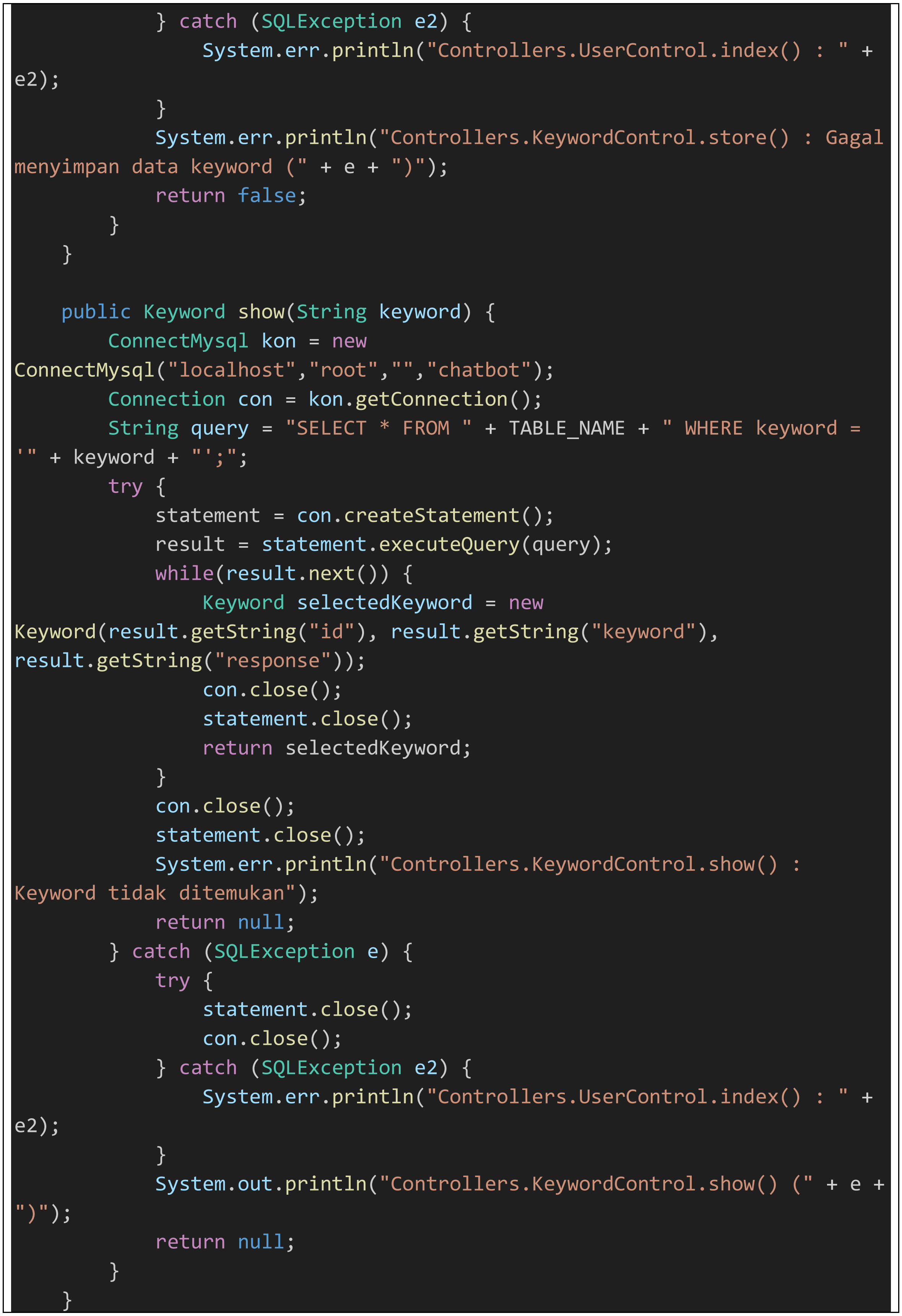
```
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author user
 */
public class ChatControl {
     final static String TABLE_NAME = "Chats";

    // statement = interface untuk menggunakan query
    static Statement statement;
    static ResultSet result;

    public ChatControl() {
    }

    public List<Chat> index() {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "SELECT * FROM " + TABLE_NAME + " ORDER BY
created_at DESC;";
        List<Chat> listChat = new ArrayList<Chat>();
        try {
            statement = con.createStatement();
            result = statement.executeQuery(query);
            while(result.next()) {
                Chat chat = new Chat(result.getString("id"),
result.getString("category"), result.getString("sender"),
result.getString("message"));
                listChat.add(chat);
            }
            statement.close();
        } catch (SQLException e) {
            System.err.println("Controllers.ChatControl.getAll() : " + e);
        }
        try {
            statement.close();
            con.close();
        } catch (SQLException e) {
            System.err.println("Controllers.UserControl.index() : " + e);
        }
        return listChat;
    }

    public boolean store(Chat chat) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "INSERT INTO " + TABLE_NAME + " VALUES(UPPER('" +
chat.getId() + "'), '" + chat.getCategory()+ "', '" + chat.getSender()+
"', '" + chat.getMessage() + "', NOW());";

        try {
            statement = con.createStatement();
            statement.executeUpdate(query);
//            System.out.println("controllers.ChatController.store() :
Berhasil menyimpan data chat");
            con.close();
            statement.close();
            return true;
        } catch (SQLException e) {
            try {
                statement.close();
                con.close();
            } catch (SQLException e2) {
                System.err.println("Controllers.UserControl.index() : " +
e2);
            }
            System.err.println("Controllers.ChatControl.store() : Gagal
menyimpan data chat (" + e + ")");
            return false;
        }
    }

    public Chat show(String id) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "SELECT * FROM " + TABLE_NAME + " WHERE id = '" +
id + "';";
        try {
            statement = con.createStatement();
            result = statement.executeQuery(query);
            while(result.next()) {
                Chat selectedChat = new Chat(result.getString("id"),
result.getString("category"), result.getString("sender"),
result.getString("message"));
                con.close();
                statement.close();
                return selectedChat;
            }
            statement.close();
            return null;
        } catch (SQLException e) {
            try {
                statement.close();
                con.close();
            } catch (SQLException e2) {
                System.err.println("Controllers.UserControl.index() : " +
e2);
            }
            System.err.println("Controllers.ChatControl.show() : " + e);
            return null;
        }
    }

    public boolean destroy(String id) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "DELETE FROM " + TABLE_NAME + " WHERE id = '" + id
+ "';";
        try {
            statement = con.createStatement();
            statement.executeUpdate(query);
//            System.out.println("controllers.ChatController.destroy() :
Data Chat berhasil dihapus");
            con.close();
            statement.close();
            return true;
        } catch (SQLException e) {
            try {
                statement.close();
                con.close();
            } catch (SQLException e2) {
                System.err.println("Controllers.UserControl.index() : " +
e2);
            }
            System.err.println("Controllers.ChatControl.destroy() : " +
e);
            return false;
        }
    }
}
```

- KeywordControl.java

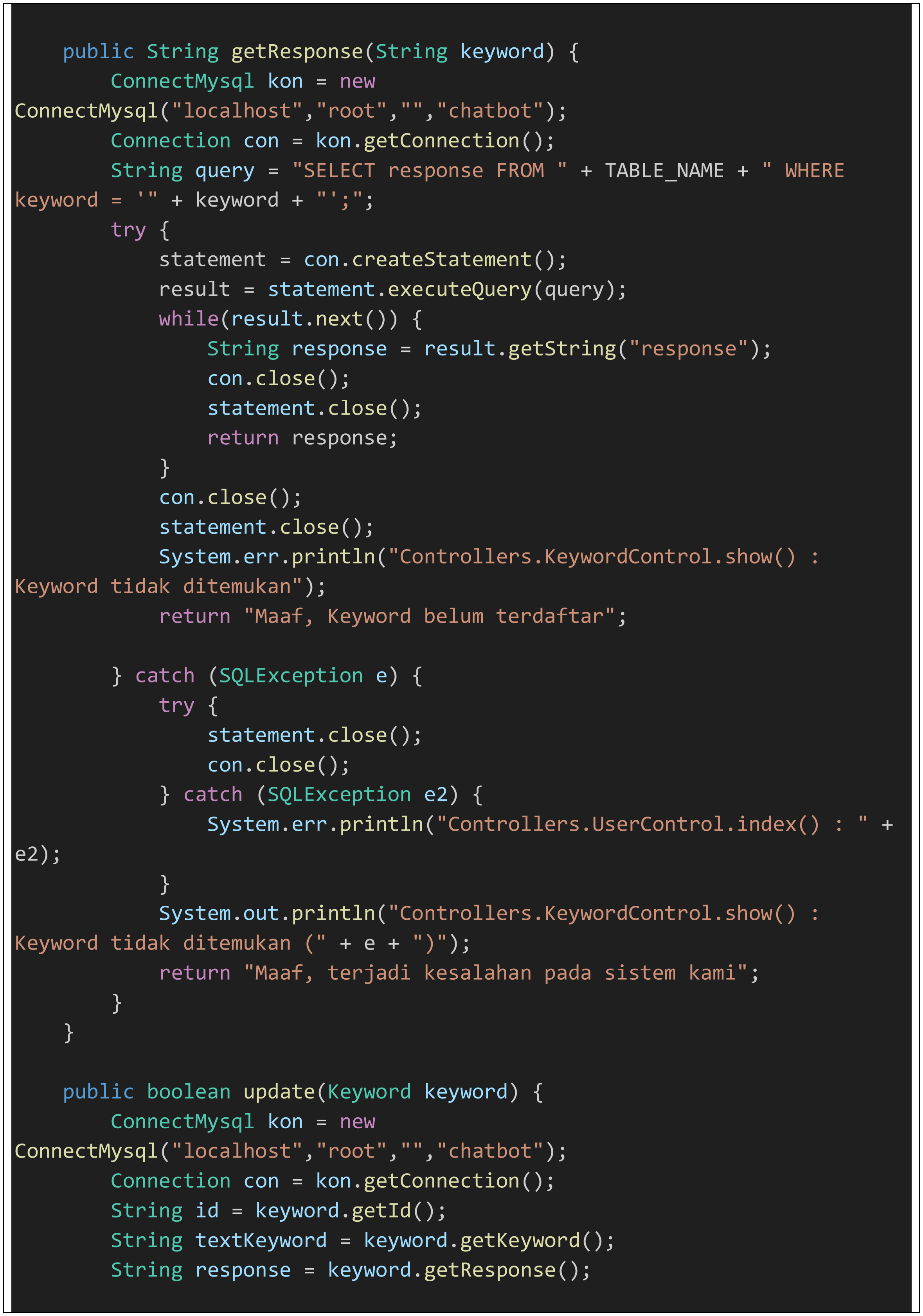
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package Controllers;

import Model.Keyword;
import com.wahyubagus.project.chatbot.ConnectMysql;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author user
 */
public class KeywordControl {
    final static String TABLE_NAME = "Keywords";

    // statement = interface untuk menggunakan query
    static Statement statement;
    static ResultSet result;

    public KeywordControl() {
    }

    public List<Keyword> index() {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
```



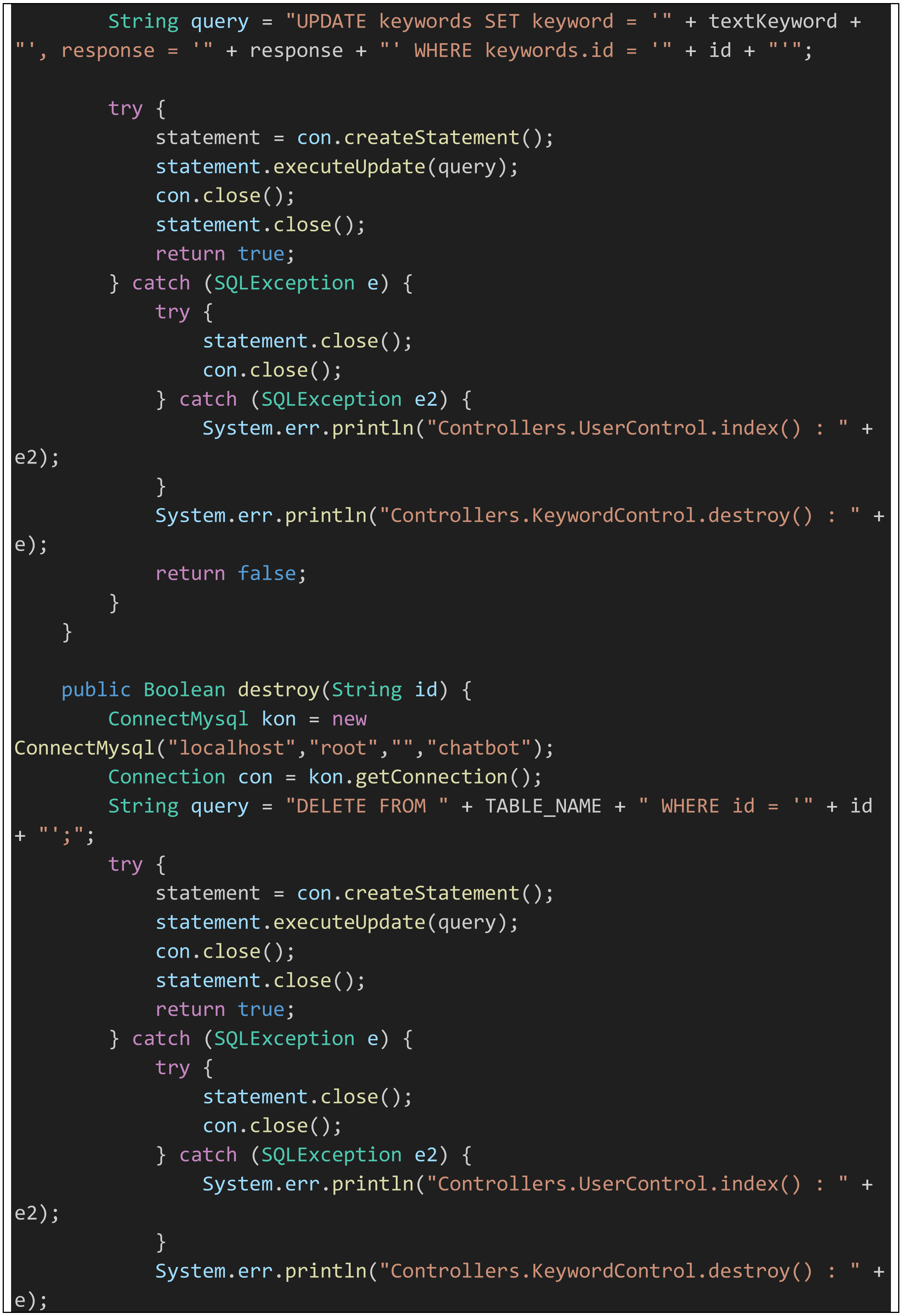
```
        String query = "SELECT * FROM " + TABLE_NAME + " ORDER BY keyword
ASC;";
        List<Keyword> listKeyword = new ArrayList<Keyword>();
        try {
            statement = con.createStatement();
            result = statement.executeQuery(query);
            while(result.next()) {
                Keyword keyword = new Keyword(result.getString("id"),
result.getString("keyword"), result.getString("response"));
                listKeyword.add(keyword);
            }
            statement.close();
        } catch (SQLException e) {
            System.err.println("Controllers.KeywordControl.getAll() : " +
e);
        }

        try {
            statement.close();
            con.close();
        } catch (SQLException e) {
            System.err.println("Controllers.UserControl.index() : " + e);
        }

        return listKeyword;
    }

    public boolean store(Keyword keyword) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "INSERT INTO " + TABLE_NAME + " VALUES(UPPER('" +
keyword.getId() + "'), '" + keyword.getKeyword()+ "', '" +
keyword.getResponse() + "');";

        try {
            statement = con.createStatement();
            statement.executeUpdate(query);
            con.close();
            statement.close();
            return true;
        } catch (SQLException e) {
            try {
                statement.close();
                con.close();
```



```
            } catch (SQLException e2) {
                System.err.println("Controllers.UserControl.index() : " +
e2);
            }
            System.err.println("Controllers.KeywordControl.store() : Gagal
menyimpan data keyword (" + e + ")");
            return false;
        }
    }

    public Keyword show(String keyword) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "SELECT * FROM " + TABLE_NAME + " WHERE keyword =
'" + keyword + "';";
        try {
            statement = con.createStatement();
            result = statement.executeQuery(query);
            while(result.next()) {
                Keyword selectedKeyword = new
Keyword(result.getString("id"), result.getString("keyword"),
result.getString("response"));
                con.close();
                statement.close();
                return selectedKeyword;
            }
            con.close();
            statement.close();
            System.err.println("Controllers.KeywordControl.show() :
Keyword tidak ditemukan");
            return null;
        } catch (SQLException e) {
            try {
                statement.close();
                con.close();
            } catch (SQLException e2) {
                System.err.println("Controllers.UserControl.index() : " +
e2);
            }
            System.out.println("Controllers.KeywordControl.show() (" + e +
")");
            return null;
        }
    }
```



```
    public String getResponse(String keyword) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "SELECT response FROM " + TABLE_NAME + " WHERE
keyword = '" + keyword + "';";
        try {
            statement = con.createStatement();
            result = statement.executeQuery(query);
            while(result.next()) {
                String response = result.getString("response");
                con.close();
                statement.close();
                return response;
            }
            con.close();
            statement.close();
            System.err.println("Controllers.KeywordControl.show() :
Keyword tidak ditemukan");
            return "Maaf, Keyword belum terdaftar";

        } catch (SQLException e) {
            try {
                statement.close();
                con.close();
            } catch (SQLException e2) {
                System.err.println("Controllers.UserControl.index() : " +
e2);
            }
            System.out.println("Controllers.KeywordControl.show() :
Keyword tidak ditemukan (" + e + ")");
            return "Maaf, terjadi kesalahan pada sistem kami";
        }
    }

    public boolean update(Keyword keyword) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String id = keyword.getId();
        String textKeyword = keyword.getKeyword();
        String response = keyword.getResponse();
```

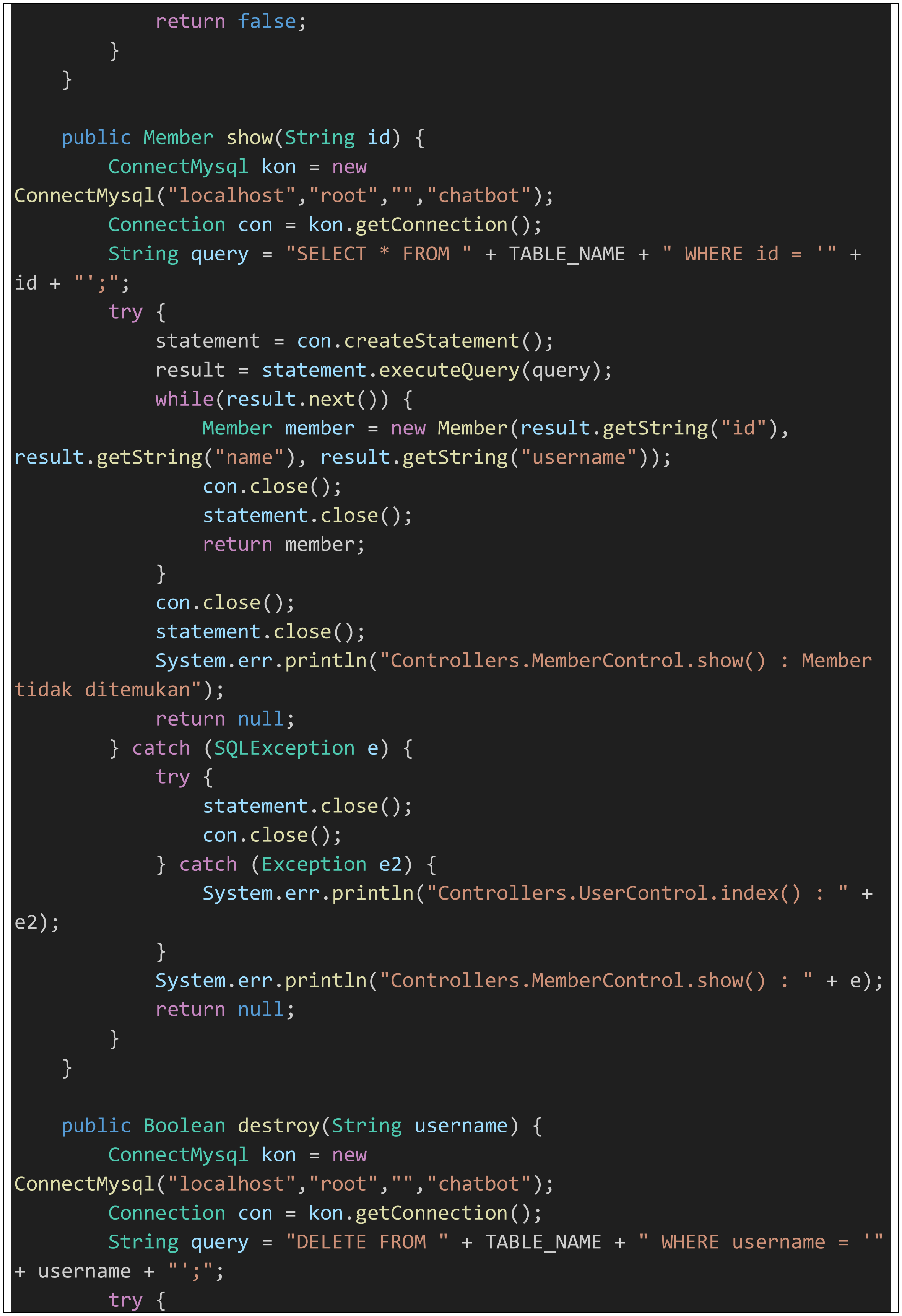


```
        String query = "UPDATE keywords SET keyword = '" + textKeyword +
"', response = '" + response + "' WHERE keywords.id = '" + id + "'";

        try {
            statement = con.createStatement();
            statement.executeUpdate(query);
            con.close();
            statement.close();
            return true;
        } catch (SQLException e) {
            try {
                statement.close();
                con.close();
            } catch (SQLException e2) {
                System.err.println("Controllers.UserControl.index() : " +
e2);
            }
            System.err.println("Controllers.KeywordControl.destroy() : " +
e);
            return false;
        }
    }

    public Boolean destroy(String id) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "DELETE FROM " + TABLE_NAME + " WHERE id = '" + id
+ "';";
        try {
            statement = con.createStatement();
            statement.executeUpdate(query);
            con.close();
            statement.close();
            return true;
        } catch (SQLException e) {
            try {
                statement.close();
                con.close();
            } catch (SQLException e2) {
                System.err.println("Controllers.UserControl.index() : " +
e2);
            }
            System.err.println("Controllers.KeywordControl.destroy() : " +
e);
```

```
            return false;
        }
    }
}
```

- MemberControl.java

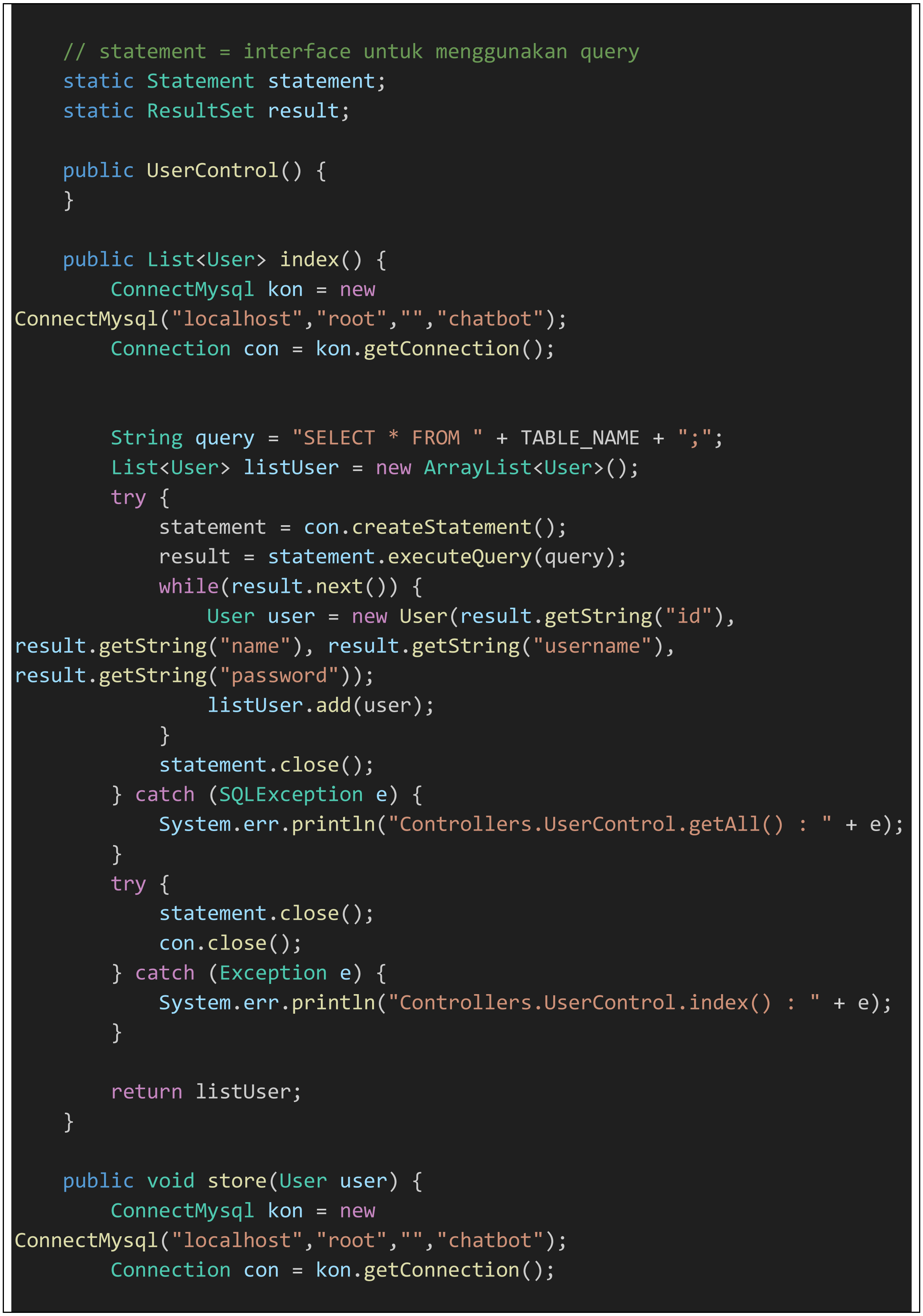
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package Controllers;

import Model.Member;
import com.wahyubagus.project.chatbot.ConnectMysql;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author user
 */
public class MemberControl {
    final static String TABLE_NAME = "Members";

    // statement = interface untuk menggunakan query
    static Statement statement;
    static ResultSet result;

    public MemberControl() {
    }

    public List<Member> index() {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "SELECT * FROM " + TABLE_NAME + " ORDER BY name;";
        List<Member> listMember = new ArrayList<Member>();
        try {
            statement = con.createStatement();
            result = statement.executeQuery(query);
            while(result.next()) {
                Member member = new Member(result.getString("id"), 
result.getString("name"), result.getString("username"));
                listMember.add(member);
            }
            statement.close();
        } catch (SQLException e) {
            System.err.println("Controllers.MemberControl.getAll() : " + 
e);
        }
        try {
            statement.close();
            con.close();
        } catch (Exception e) {
            System.err.println("Controllers.UserControl.index() : " + e);
        }
        return listMember;
    }

    public boolean store(Member member) {
        ConnectMysql kon = new 
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "INSERT INTO " + TABLE_NAME + " VALUES(UPPER('" + 
member.getId() + "'), '" + member.getName() + "', '" + 
member.getUsername()+ "')";

        try {
            statement = con.createStatement();
            statement.executeUpdate(query);
            con.close();
            statement.close();
            return true;
        } catch (SQLException e) {
            try {
                statement.close();
                con.close();
            } catch (Exception e2) {
                System.err.println("Controllers.UserControl.index() : " + 
e2);
            }
            System.err.println("Controllers.MemberControl.store() : Gagal 
menyimpan data member (" + e + ")");
```



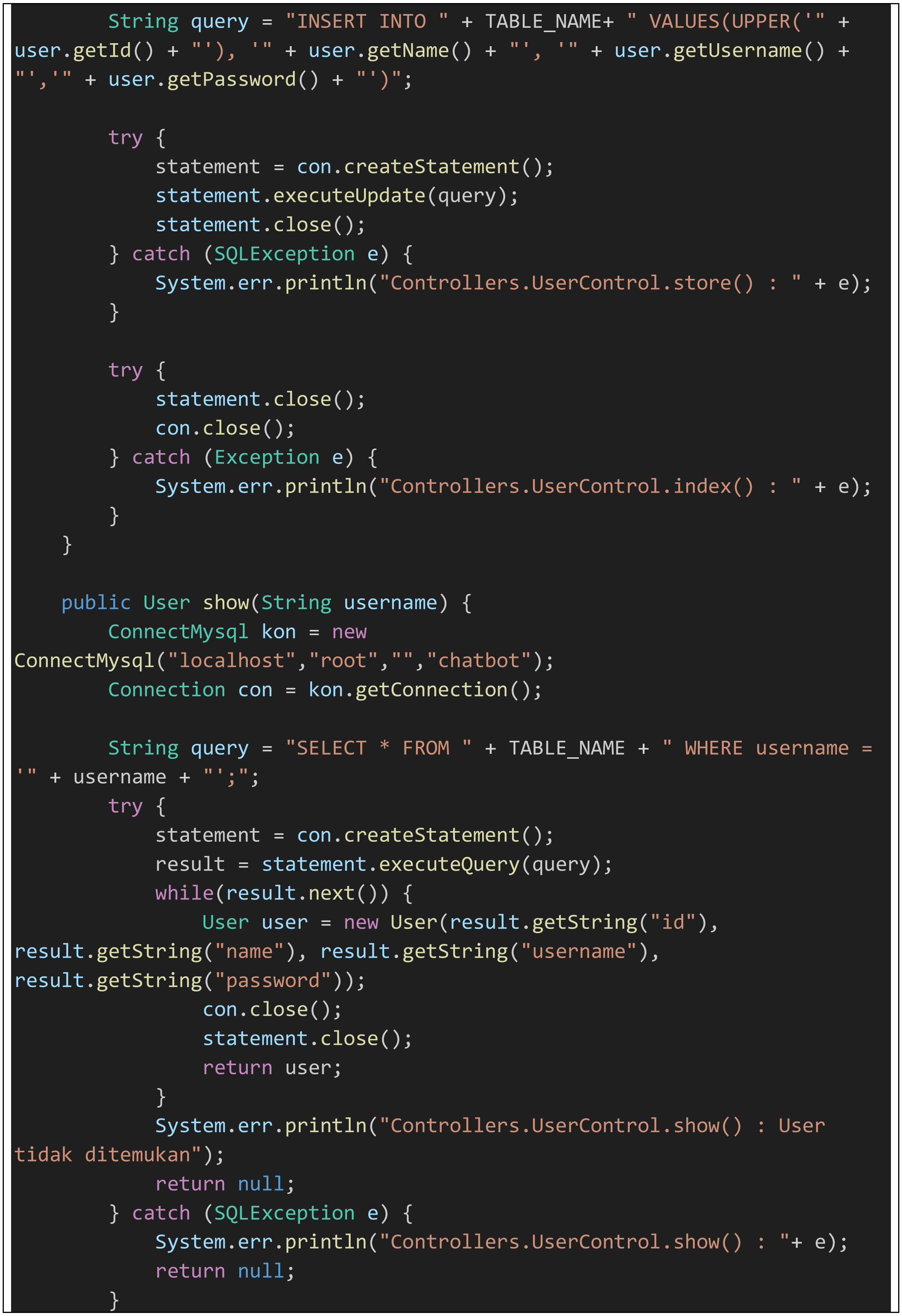
```
            return false;
        }
    }

    public Member show(String id) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "SELECT * FROM " + TABLE_NAME + " WHERE id = '" +
id + "';";
        try {
            statement = con.createStatement();
            result = statement.executeQuery(query);
            while(result.next()) {
                Member member = new Member(result.getString("id"),
result.getString("name"), result.getString("username"));
                con.close();
                statement.close();
                return member;
            }
            con.close();
            statement.close();
            System.err.println("Controllers.MemberControl.show() : Member
tidak ditemukan");
            return null;
        } catch (SQLException e) {
            try {
                statement.close();
                con.close();
            } catch (Exception e2) {
                System.err.println("Controllers.UserControl.index() : " +
e2);
            }
            System.err.println("Controllers.MemberControl.show() : " + e);
            return null;
        }
    }

    public Boolean destroy(String username) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
        String query = "DELETE FROM " + TABLE_NAME + " WHERE username = '"
+ username + "';";
        try {
            statement = con.createStatement();
            statement.executeUpdate(query);
            con.close();
            statement.close();
            return true;
        } catch (Exception e) {
            try {
                statement.close();
                con.close();
            } catch (Exception e2) {
                System.err.println("Controllers.UserControl.index() : " +
e2);
            }
            System.err.println("Controllers.MemberControl.destroy() : " +
e);
            return false;
        }
    }
}
```

- UserControl.java

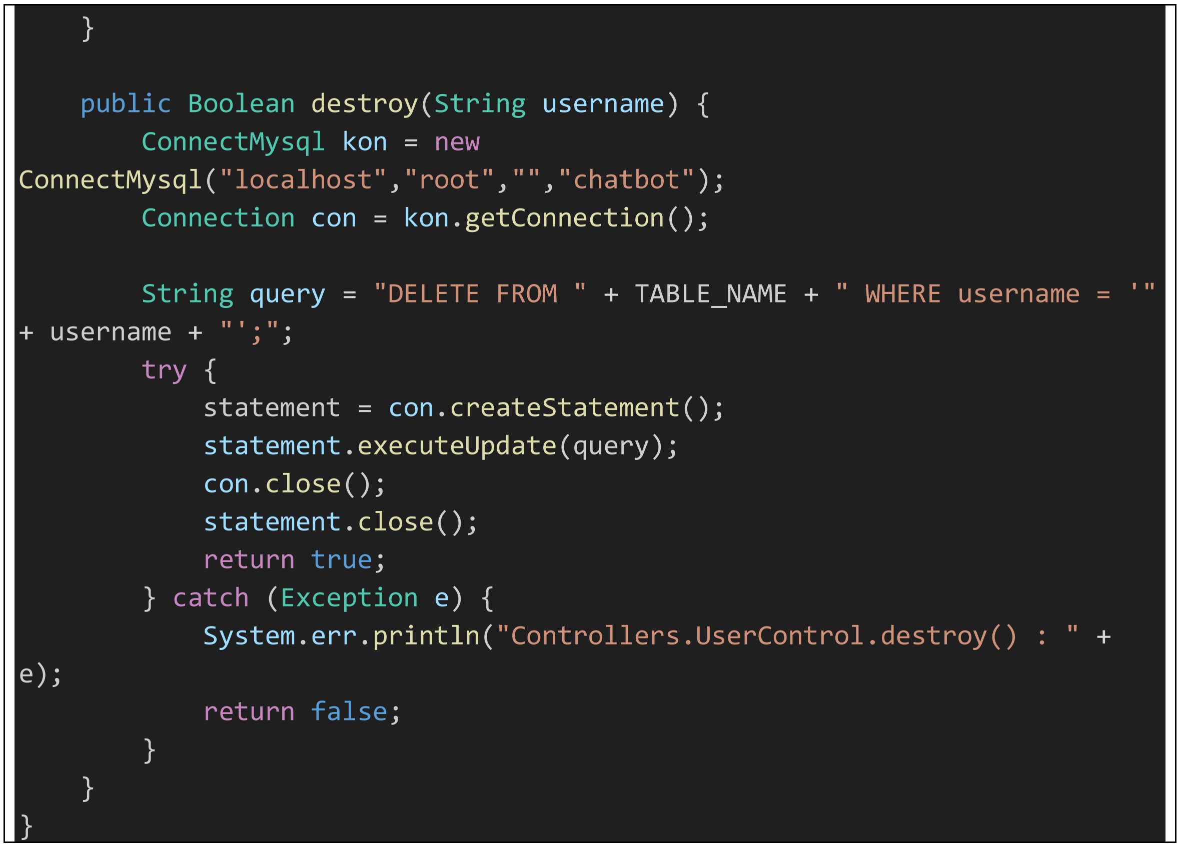
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package Controllers;

import Model.User;
import com.wahyubagus.project.chatbot.ConnectMysql;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author user
 */
public class UserControl {
    final static String TABLE_NAME = "users";
```



```
    // statement = interface untuk menggunakan query
    static Statement statement;
    static ResultSet result;

    public UserControl() {
    }

    public List<User> index() {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();


        String query = "SELECT * FROM " + TABLE_NAME + ";";
        List<User> listUser = new ArrayList<User>();
        try {
            statement = con.createStatement();
            result = statement.executeQuery(query);
            while(result.next()) {
                User user = new User(result.getString("id"),
result.getString("name"), result.getString("username"),
result.getString("password"));
                listUser.add(user);
            }
            statement.close();
        } catch (SQLException e) {
            System.err.println("Controllers.UserControl.getAll() : " + e);
        }
        try {
            statement.close();
            con.close();
        } catch (Exception e) {
            System.err.println("Controllers.UserControl.index() : " + e);
        }

        return listUser;
    }

    public void store(User user) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();
```

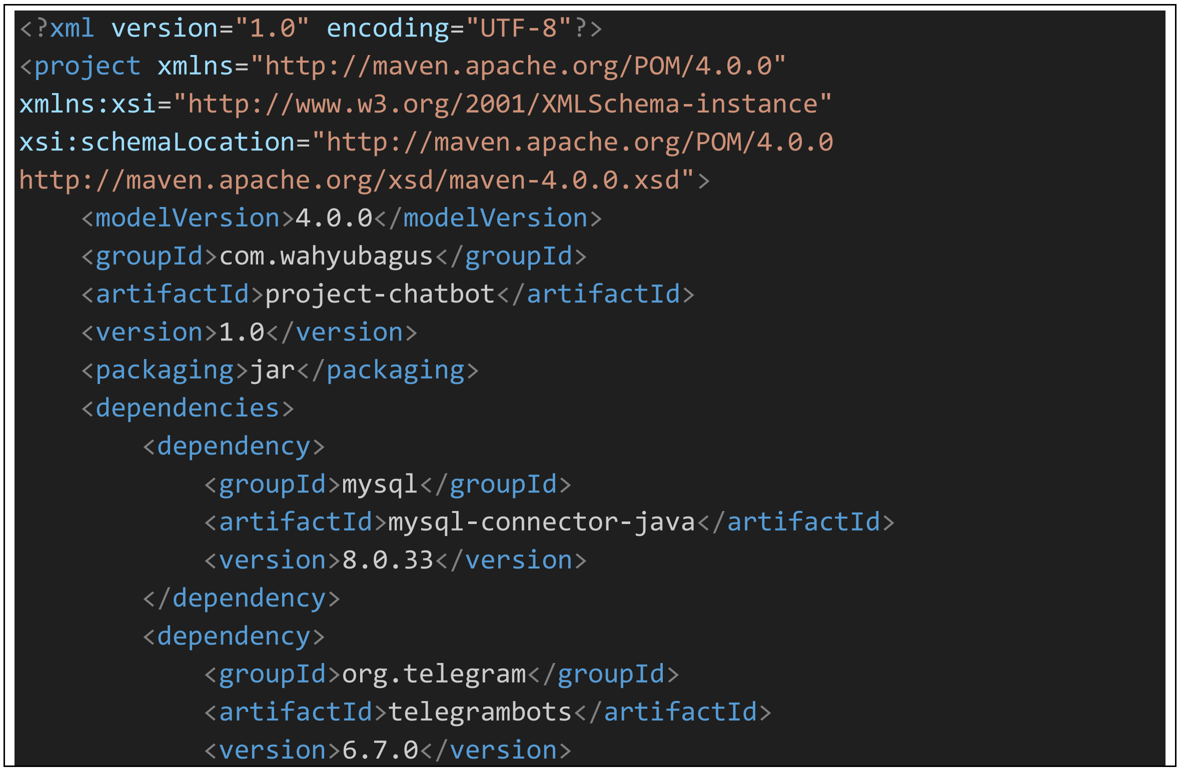


```
        String query = "INSERT INTO " + TABLE_NAME+ " VALUES(UPPER('" +
user.getId() + "'), '" + user.getName() + "', '" + user.getUsername() +
"','" + user.getPassword() + "')";

        try {
            statement = con.createStatement();
            statement.executeUpdate(query);
            statement.close();
        } catch (SQLException e) {
            System.err.println("Controllers.UserControl.store() : " + e);
        }

        try {
            statement.close();
            con.close();
        } catch (Exception e) {
            System.err.println("Controllers.UserControl.index() : " + e);
        }
    }

    public User show(String username) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();

        String query = "SELECT * FROM " + TABLE_NAME + " WHERE username =
'" + username + "';";
        try {
            statement = con.createStatement();
            result = statement.executeQuery(query);
            while(result.next()) {
                User user = new User(result.getString("id"),
result.getString("name"), result.getString("username"),
result.getString("password"));
                con.close();
                statement.close();
                return user;
            }
            System.err.println("Controllers.UserControl.show() : User
tidak ditemukan");
            return null;
        } catch (SQLException e) {
            System.err.println("Controllers.UserControl.show() : "+ e);
            return null;
        }
```



```
    }

    public Boolean destroy(String username) {
        ConnectMysql kon = new
ConnectMysql("localhost","root","","chatbot");
        Connection con = kon.getConnection();

        String query = "DELETE FROM " + TABLE_NAME + " WHERE username = '"
+ username + "';";
        try {
            statement = con.createStatement();
            statement.executeUpdate(query);
            con.close();
            statement.close();
            return true;
        } catch (Exception e) {
            System.err.println("Controllers.UserControl.destroy() : " +
e);
            return false;
        }
    }
}
```

- pom.xml



```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.wahyubagus</groupId>
    <artifactId>project-chatbot</artifactId>
    <version>1.0</version>
    <packaging>jar</packaging>
    <dependencies>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.33</version>
        </dependency>
        <dependency>
            <groupId>org.telegram</groupId>
            <artifactId>telegrambots</artifactId>
            <version>6.7.0</version>
        </dependency>
        <dependency>
            <groupId>com.github.pengrad</groupId>
            <artifactId>java-telegram-bot-api</artifactId>
            <version>6.7.0</version>
        </dependency>
        <dependency>
            <groupId>com.google.guava</groupId>
            <artifactId>guava</artifactId>
            <version>11.0.2</version>
        </dependency>
    </dependencies>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>19</maven.compiler.source>
        <maven.compiler.target>19</maven.compiler.target>
        <exec.mainClass>com.wahyubagus.project.chatbot.ProjectChatbot</exec.mainClass>
    </properties>

    <build>
        <plugins>
            <plugin>
            <artifactId>maven-assembly-plugin</artifactId>
            <version>2.4</version>
               <configuration>
                    <descriptorRefs>
                        <descriptorRef>jar-with-dependencies</descriptorRef>
                    </descriptorRefs>
                    <archive>
                        <manifest>
                            <mainClass>com.wahyubagus.project.chatbot.Main</mainClass>
                        </manifest>
                    </archive>
                </configuration>
            <executions>
                <execution>
                <id>make-assembly</id>
                <phase>package</phase>
                    <goals>
                        <goal>single</goal>
                    </goals>
                </execution>
            </executions>
            </plugin>
        </plugins>
    </build>
</project>
```