Nama : Achmad Azril Auladi

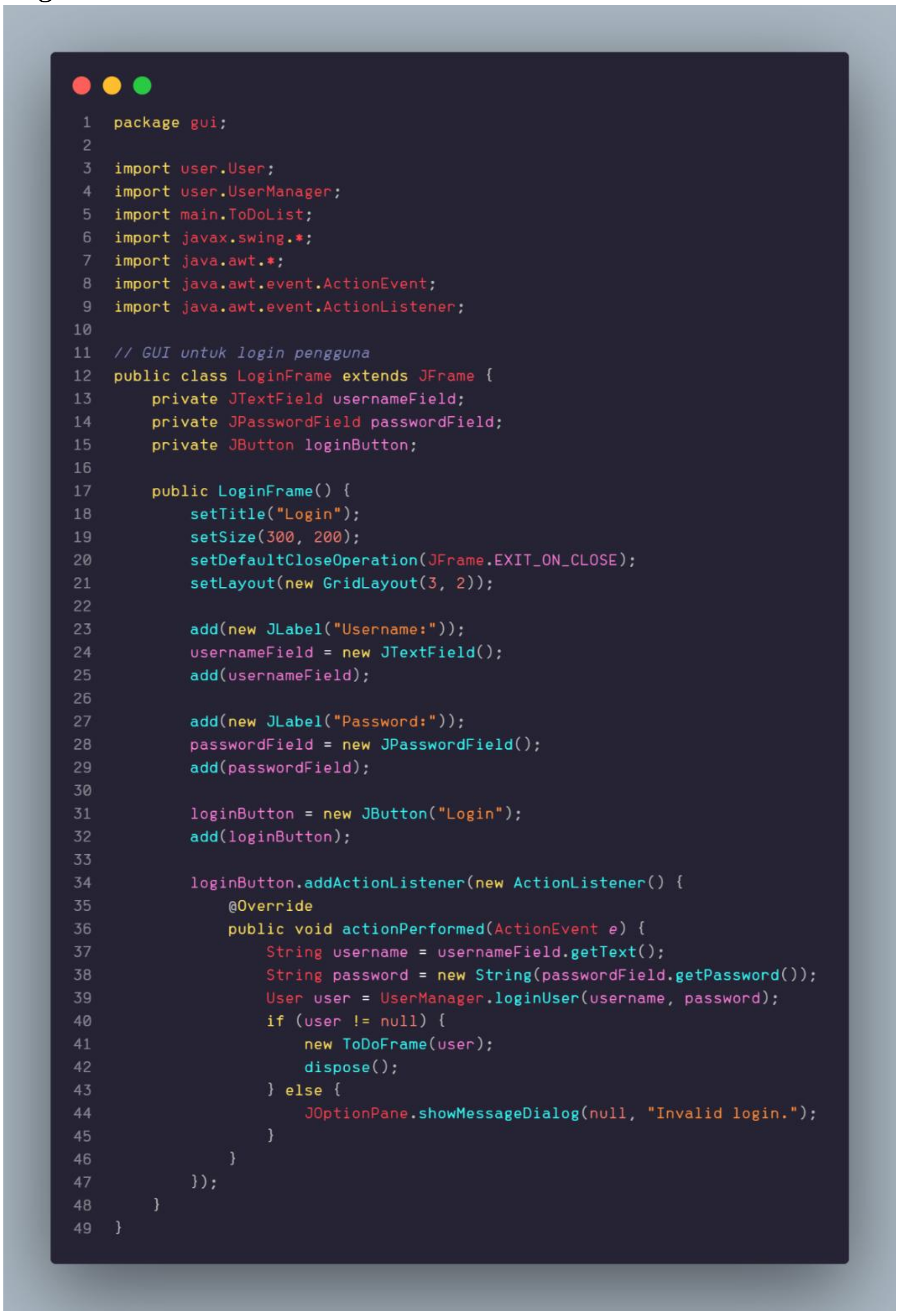
NIM : 2309106049

Kelas : Praktikum PBO B1’23

SCREENSHOOT PROGRAM POSTTEST 4

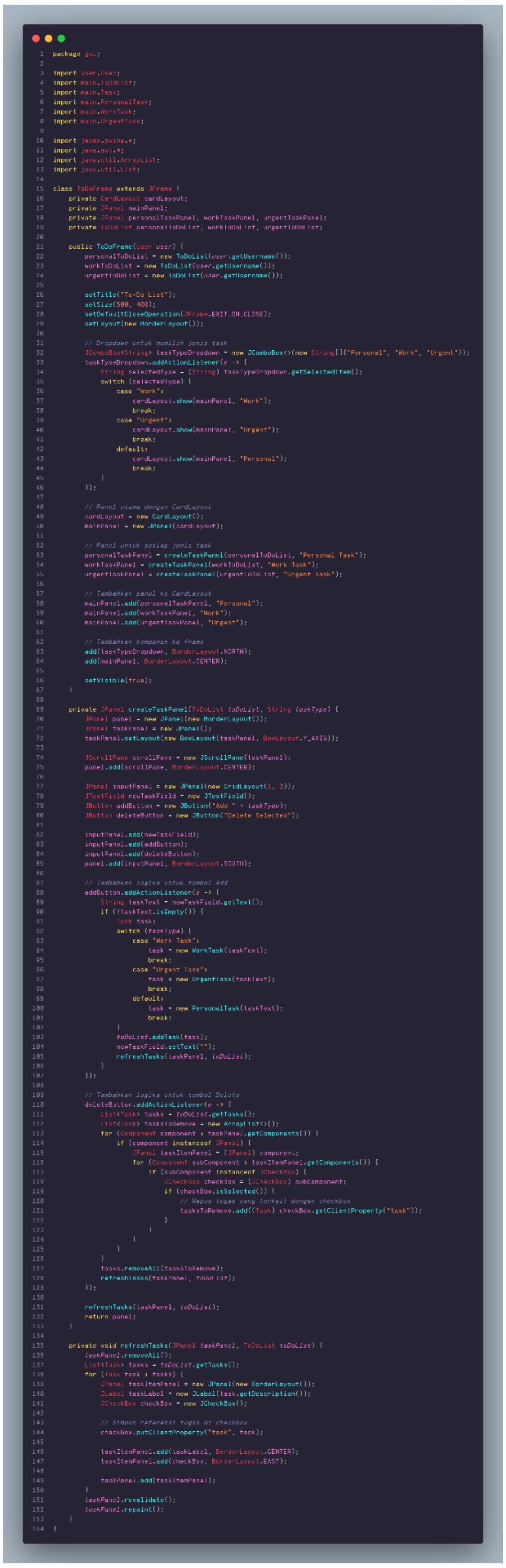
Package gui:

1. LoginFrame



```java
package gui;

import user.User;
import user.UserManager;
import main.ToDoList;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

// GUI untuk login pengguna
public class LoginFrame extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;
    private JButton loginButton;

    public LoginFrame() {
        setTitle("Login");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(3, 2));

        add(new JLabel("Username:"));
        usernameField = new JTextField();
        add(usernameField);

        add(new JLabel("Password:"));
        passwordField = new JPasswordField();
        add(passwordField);

        loginButton = new JButton("Login");
        add(loginButton);

        loginButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String username = usernameField.getText();
                String password = new String(passwordField.getPassword());
                User user = UserManager.loginUser(username, password);
                if (user != null) {
                    new ToDoFrame(user);
                    dispose();
                } else {
                    JOptionPane.showMessageDialog(null, "Invalid login.");
                }
            }
        });
    }
}
```

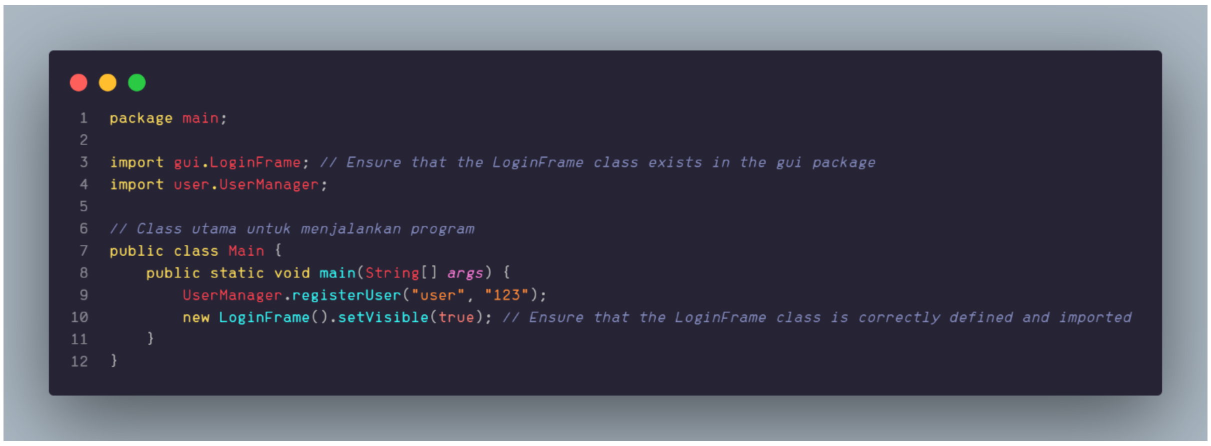
2. ToDoFrame



```java
package gui;

import user.User;
import main.ToDoList;
import main.Task;
import main.PersonalTask;
import main.WorkTask;
import main.UrgentTask;

import javax.swing.*;
import java.awt.*;
import java.util.ArrayList;
import java.util.List;

class ToDoFrame extends JFrame {
    private CardLayout cardLayout;
    private JPanel mainPanel;
    private JPanel personalTaskPanel, workTaskPanel, urgentTaskPanel;
    private ToDoList personalToDoList, workToDoList, urgentToDoList;

    public ToDoFrame(User user) {
        personalToDoList = new ToDoList(user.getUsername());
        workToDoList = new ToDoList(user.getUsername());
        urgentToDoList = new ToDoList(user.getUsername());

        setTitle("To-Do List");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        // Dropdown untuk memilih jenis task
        JComboBox<String> taskTypeDropdown = new JComboBox<>(new String[]{"Personal", "Work", "Urgent"});
        taskTypeDropdown.addActionListener(e -> {
            String selectedType = (String) taskTypeDropdown.getSelectedItem();
            switch (selectedType) {
                case "Work":
                    cardLayout.show(mainPanel, "Work");
                    break;
                case "Urgent":
                    cardLayout.show(mainPanel, "Urgent");
                    break;
                default:
                    cardLayout.show(mainPanel, "Personal");
                    break;
            }
        });

        // Panel utama dengan CardLayout
        cardLayout = new CardLayout();
        mainPanel = new JPanel(cardLayout);

        // Panel untuk setiap jenis task
        personalTaskPanel = createTaskPanel(personalToDoList, "Personal Task");
        workTaskPanel = createTaskPanel(workToDoList, "Work Task");
        urgentTaskPanel = createTaskPanel(urgentToDoList, "Urgent Task");

        // Tambahkan panel ke CardLayout
        mainPanel.add(personalTaskPanel, "Personal");
        mainPanel.add(workTaskPanel, "Work");
        mainPanel.add(urgentTaskPanel, "Urgent");

        // Tambahkan komponen ke frame
        add(taskTypeDropdown, BorderLayout.NORTH);
        add(mainPanel, BorderLayout.CENTER);

        setVisible(true);
    }

    private JPanel createTaskPanel(ToDoList toDoList, String taskType) {
        JPanel panel = new JPanel(new BorderLayout());
        JPanel taskPanel = new JPanel();
        taskPanel.setLayout(new BoxLayout(taskPanel, BoxLayout.Y_AXIS));

        JScrollPane scrollPane = new JScrollPane(taskPanel);
        panel.add(scrollPane, BorderLayout.CENTER);

        JPanel inputPanel = new JPanel(new GridLayout(1, 3));
        JTextField newTaskField = new JTextField();
        JButton addButton = new JButton("Add " + taskType);
        JButton deleteButton = new JButton("Delete Selected");

        inputPanel.add(newTaskField);
        inputPanel.add(addButton);
        inputPanel.add(deleteButton);
        panel.add(inputPanel, BorderLayout.SOUTH);

        // Tambahkan logika untuk tombol Add
        addButton.addActionListener(e -> {
            String taskText = newTaskField.getText();
            if (!taskText.isEmpty()) {
                Task task;
                switch (taskType) {
                    case "Work Task":
                        task = new WorkTask(taskText);
                        break;
                    case "Urgent Task":
                        task = new UrgentTask(taskText);
                        break;
                    default:
                        task = new PersonalTask(taskText);
                        break;
                }
                toDoList.addTask(task);
                newTaskField.setText("");
                refreshTasks(taskPanel, toDoList);
            }
        });

        // Tambahkan logika untuk tombol Delete
        deleteButton.addActionListener(e -> {
            List<Task> tasks = toDoList.getTasks();
            List<Task> tasksToRemove = new ArrayList<>();
            for (Component component : taskPanel.getComponents()) {
                if (component instanceof JPanel) {
                    JPanel taskItemPanel = (JPanel) component;
                    for (Component subComponent : taskItemPanel.getComponents()) {
                        if (subComponent instanceof JCheckBox) {
                            JCheckBox checkBox = (JCheckBox) subComponent;
                            if (checkBox.isSelected()) {
                                // Hapus tugas yang terkait dengan checkbox
                                tasksToRemove.add((Task) checkBox.getClientProperty("task"));
                            }
                        }
                    }
                }
            }
            tasks.removeAll(tasksToRemove);
            refreshTasks(taskPanel, toDoList);
        });

        refreshTasks(taskPanel, toDoList);
        return panel;
    }

    private void refreshTasks(JPanel taskPanel, ToDoList toDoList) {
        taskPanel.removeAll();
        List<Task> tasks = toDoList.getTasks();
        for (Task task : tasks) {
            JPanel taskItemPanel = new JPanel(new BorderLayout());
            JLabel taskLabel = new JLabel(task.getDescription());
            JCheckBox checkBox = new JCheckBox();

            // Simpan referensi tugas di checkbox
            checkBox.putClientProperty("task", task);

            taskItemPanel.add(taskLabel, BorderLayout.CENTER);
            taskItemPanel.add(checkBox, BorderLayout.EAST);

            taskPanel.add(taskItemPanel);
        }
        taskPanel.revalidate();
        taskPanel.repaint();
    }
}
```

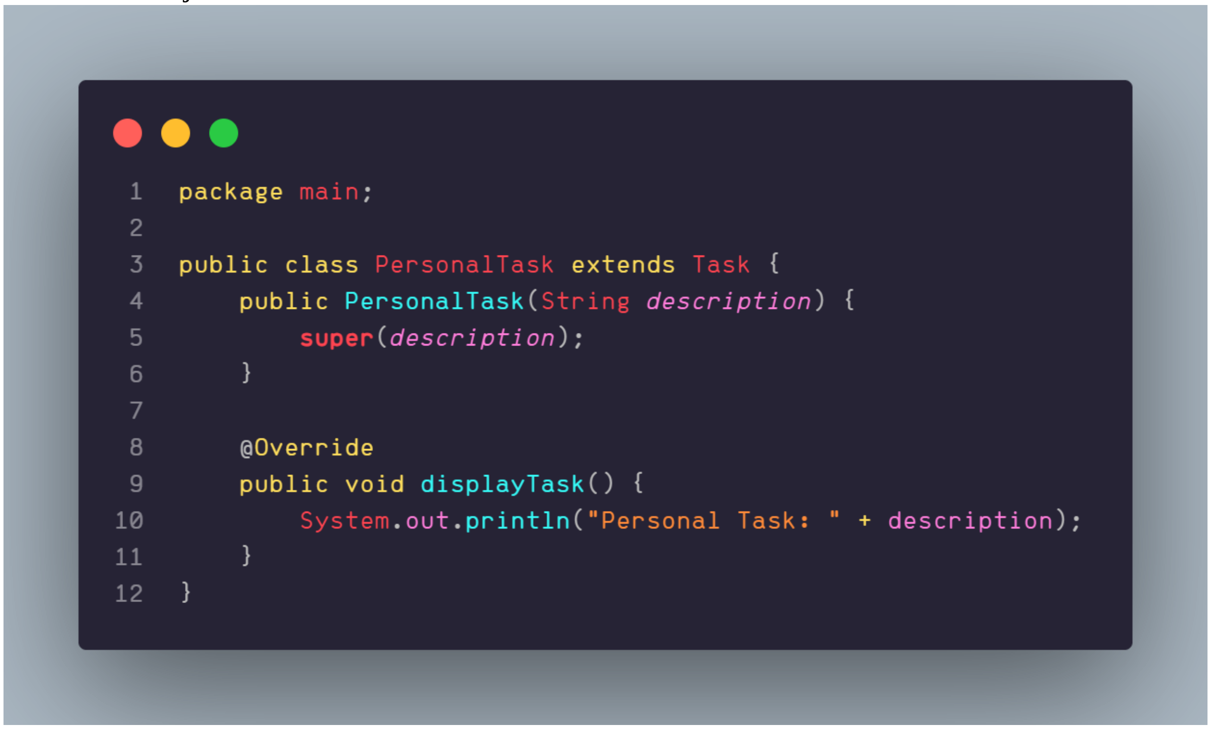
Package Main:

1. Main.Java



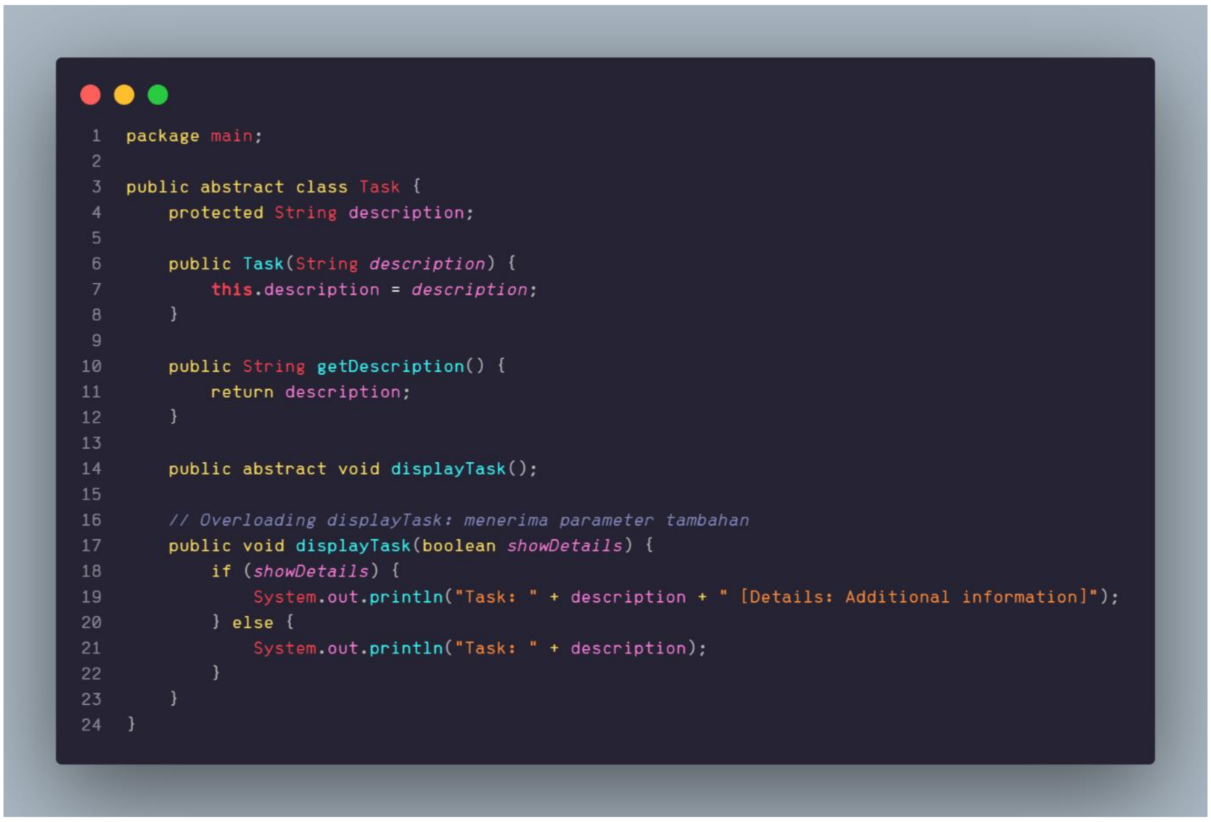
```java
package main;

import gui.LoginFrame; // Ensure that the LoginFrame class exists in the gui package
import user.UserManager;

// Class utama untuk menjalankan program
public class Main {
    public static void main(String[] args) {
        UserManager.registerUser("user", "123");
        new LoginFrame().setVisible(true); // Ensure that the LoginFrame class is correctly defined and imported
    }
}
```

2. PersonalTask.java



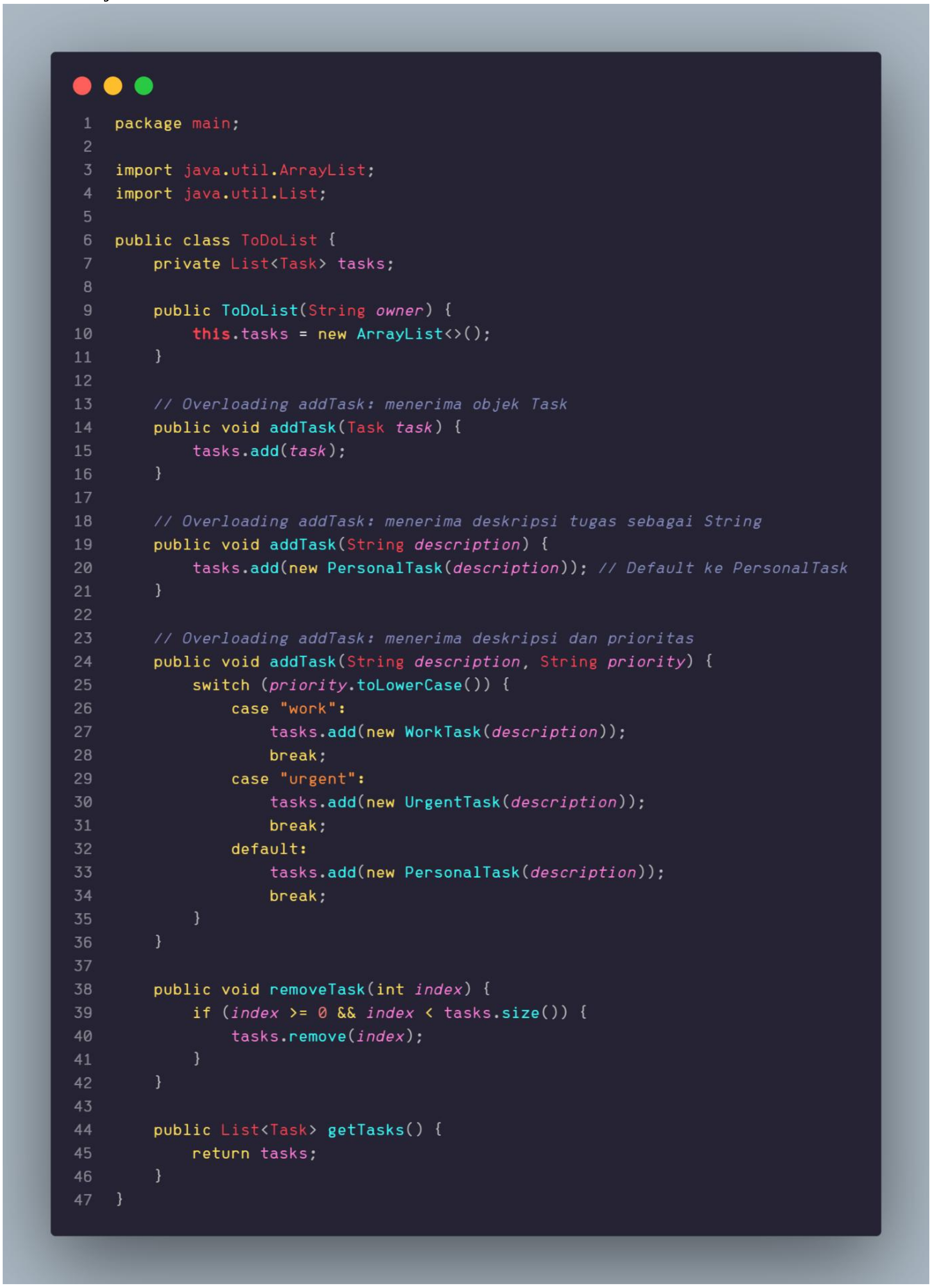
```java
package main;

public class PersonalTask extends Task {
    public PersonalTask(String description) {
        super(description);
    }

    @Override
    public void displayTask() {
        System.out.println("Personal Task: " + description);
    }
}
```

3. Task.Java



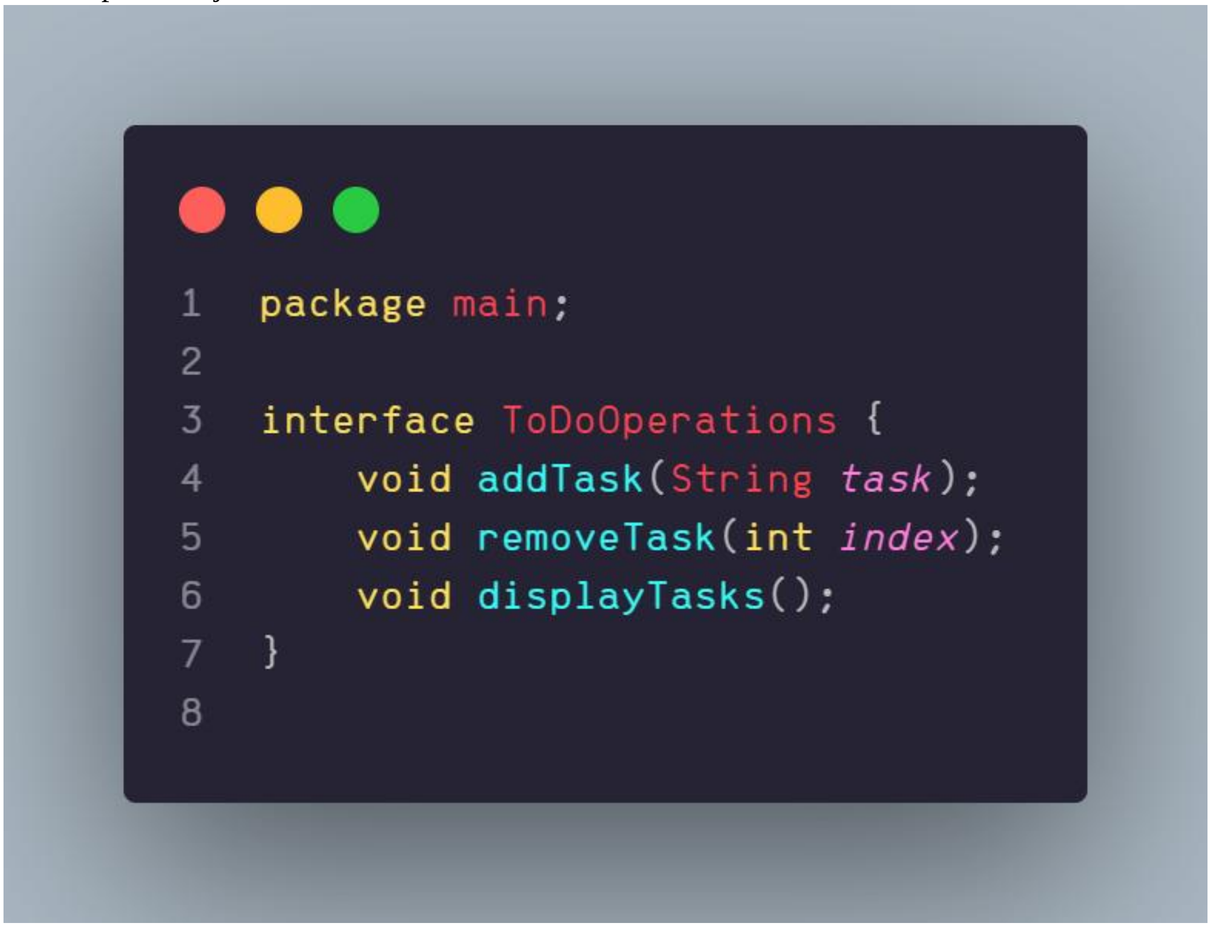
```java
package main;

public abstract class Task {
    protected String description;

    public Task(String description) {
        this.description = description;
    }

    public String getDescription() {
        return description;
    }

    public abstract void displayTask();

    // Overloading displayTask: menerima parameter tambahan
    public void displayTask(boolean showDetails) {
        if (showDetails) {
            System.out.println("Task: " + description + " [Details: Additional information]");
        } else {
            System.out.println("Task: " + description);
        }
    }
}
```

4. ToDoList.java



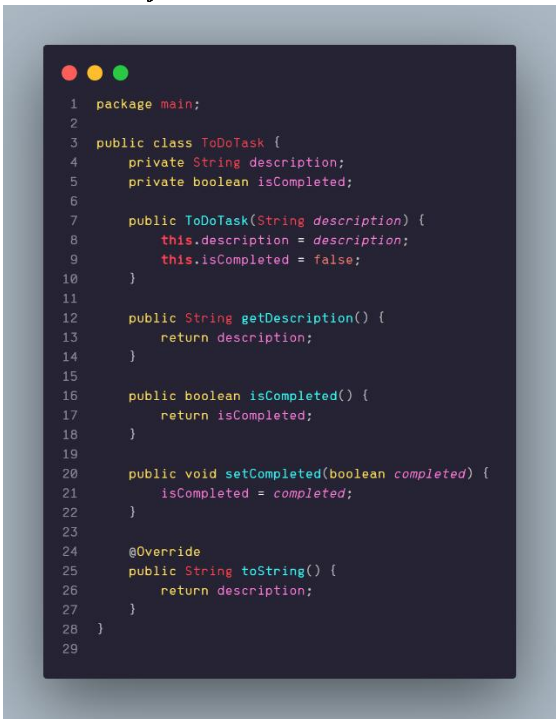
```java
package main;

import java.util.ArrayList;
import java.util.List;

public class ToDoList {
    private List<Task> tasks;

    public ToDoList(String owner) {
        this.tasks = new ArrayList<>();
    }

    // Overloading addTask: menerima objek Task
    public void addTask(Task task) {
        tasks.add(task);
    }

    // Overloading addTask: menerima deskripsi tugas sebagai String
    public void addTask(String description) {
        tasks.add(new PersonalTask(description)); // Default ke PersonalTask
    }

    // Overloading addTask: menerima deskripsi dan prioritas
    public void addTask(String description, String priority) {
        switch (priority.toLowerCase()) {
            case "work":
                tasks.add(new WorkTask(description));
                break;
            case "urgent":
                tasks.add(new UrgentTask(description));
                break;
            default:
                tasks.add(new PersonalTask(description));
                break;
        }
    }

    public void removeTask(int index) {
        if (index >= 0 && index < tasks.size()) {
            tasks.remove(index);
        }
    }

    public List<Task> getTasks() {
        return tasks;
    }
}
```

5. ToDoOperation.java



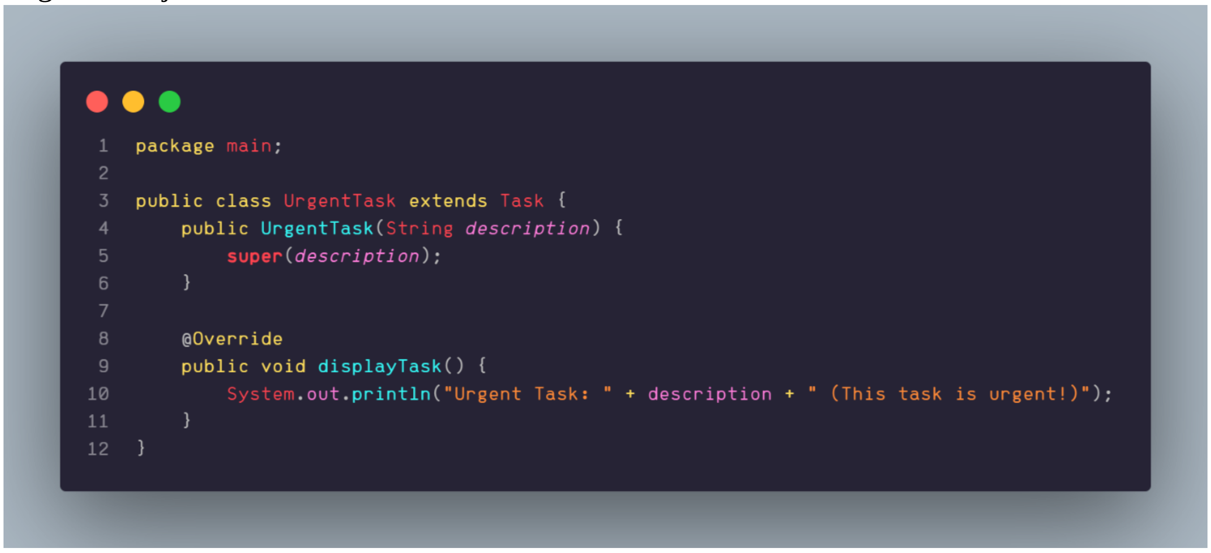
```java
package main;

interface ToDoOperations {
        void addTask(String task);
        void removeTask(int index);
        void displayTasks();
}

```

6. ToDoTask.java



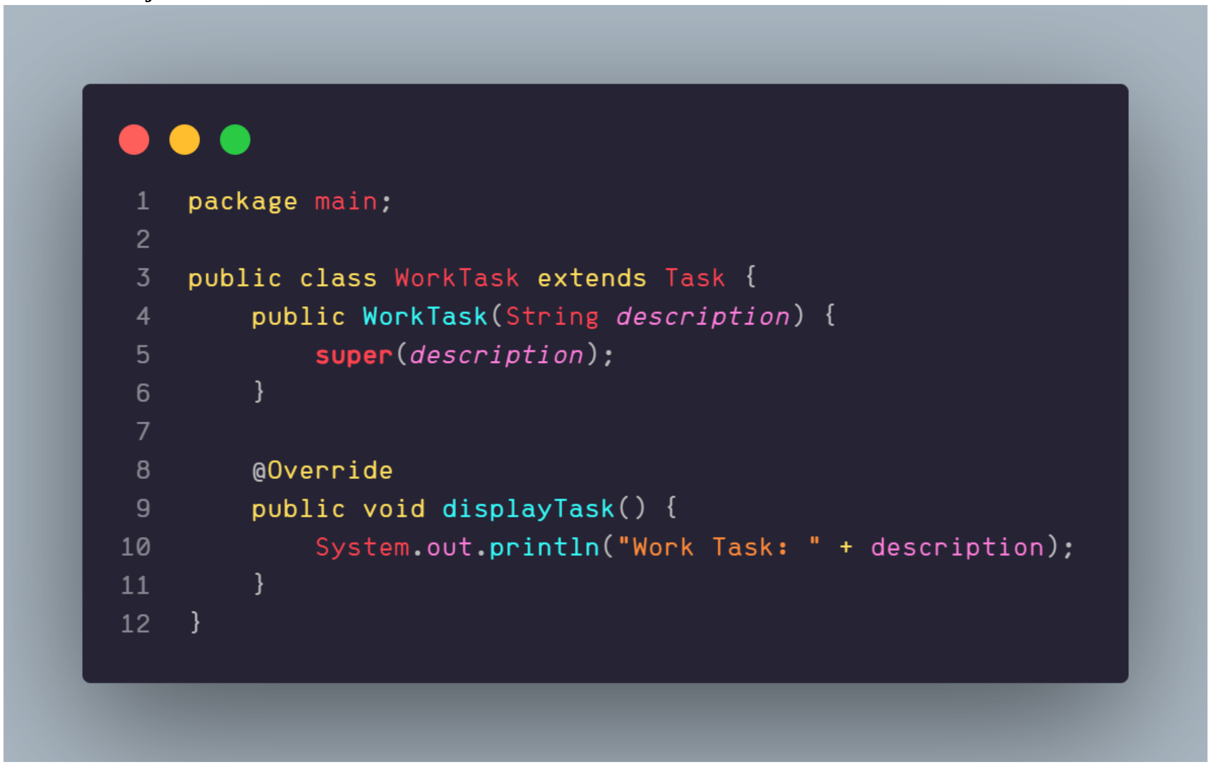
```java
package main;

public class ToDoTask {
    private String description;
    private boolean isCompleted;

    public ToDoTask(String description) {
        this.description = description;
        this.isCompleted = false;
    }

    public String getDescription() {
        return description;
    }

    public boolean isCompleted() {
        return isCompleted;
    }

    public void setCompleted(boolean completed) {
        isCompleted = completed;
    }

    @Override
    public String toString() {
        return description;
    }
}

```

7. UrgentTask.java



```java
package main;

public class UrgentTask extends Task {
    public UrgentTask(String description) {
        super(description);
    }

    @Override
    public void displayTask() {
        System.out.println("Urgent Task: " + description + " (This task is urgent!)");
    }
}
```

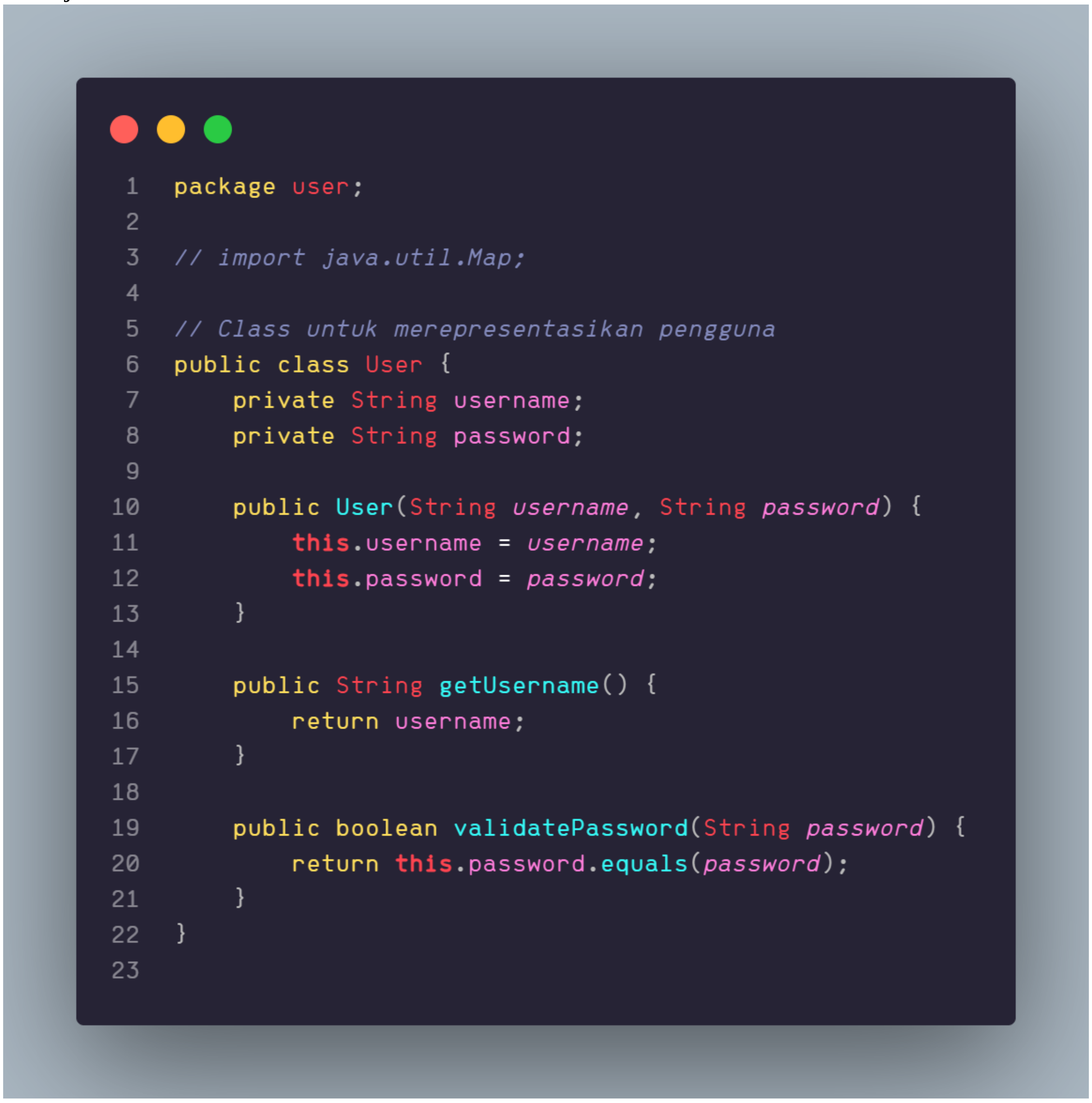
8. WorkTask.java



```java
package main;

public class WorkTask extends Task {
    public WorkTask(String description) {
        super(description);
    }

    @Override
    public void displayTask() {
        System.out.println("Work Task: " + description);
    }
}
```

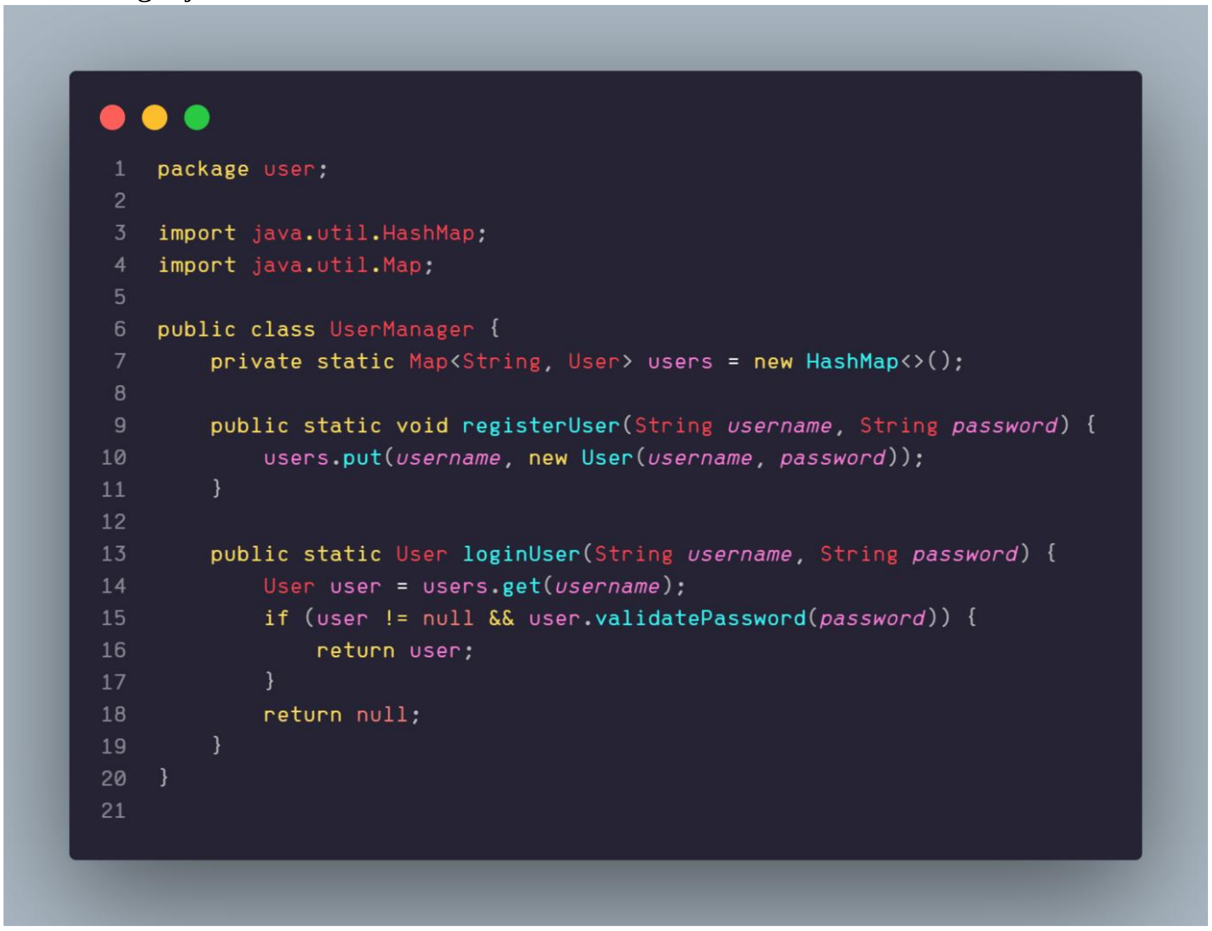
Package user:

1. User.java



```java
package user;

// import java.util.Map;

// Class untuk merepresentasikan pengguna
public class User {
    private String username;
    private String password;

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

    public String getUsername() {
        return username;
    }

    public boolean validatePassword(String password) {
        return this.password.equals(password);
    }
}

```

2. UserManager.java



```java
package user;

import java.util.HashMap;
import java.util.Map;

public class UserManager {
    private static Map<String, User> users = new HashMap<>();

    public static void registerUser(String username, String password) {
        users.put(username, new User(username, password));
    }

    public static User loginUser(String username, String password) {
        User user = users.get(username);
        if (user != null && user.validatePassword(password)) {
            return user;
        }
        return null;
    }
}

```