

Nama : Muhamad Handika Mawardi
Kelas : 2TI03
Nim : 2222105245

Dosen Pengampu : Novan Zulkarnain, S.T.,M.Kom
Mata Kuliah : PEMROGRAMAN BERORIENTASI OBJEK

PENJELASAN PROGRAM PERMAINAN KUIS TEBAK TEBAK NGESELIN DEH!

1. Deklarasi Kelas dan Variabel

- ``Question` Class:`
 - Mendefinisikan objek pertanyaan dengan atribut pertanyaan, opsi jawaban, jawaban yang benar, dan hint.
- ``QuizGame` Class:`
 - Kelas utama yang mewarisi ``JFrame`` untuk membuat GUI.
 - Menyimpan pertanyaan dalam array ``questions``.
 - Menyimpan status permainan seperti indeks pertanyaan saat ini (``currentQuestionIndex``) dan skor (``score``).

2. Inisialisasi GUI

- Membuat tampilan awal menggunakan ``CardLayout`` untuk memungkinkan transisi antar panel.
- Membuat ``startPanel`` untuk halaman awal yang berisi pesan selamat datang dan tombol mulai.
- Membuat ``quizPanel`` untuk halaman kuis yang berisi pertanyaan dan opsi jawaban.

3. Setup Tampilan Awal

- ``startPanel`` menggunakan ``BoxLayout`` untuk menempatkan label selamat datang di atas dan tombol mulai di tengah.
- ``startButton`` ketika diklik akan menampilkan ``quizPanel`` dan memulai pertanyaan pertama.

4. Setup Halaman Kuis

- ``quizPanel`` menggunakan ``BorderLayout`` untuk menempatkan label pertanyaan di atas, tombol jawaban di tengah, dan tombol hint di bawah.
- Tombol-tombol jawaban diatur dalam ``GridLayout`` dengan ukuran yang tetap agar tidak memenuhi layar.

5. Fungsi Menampilkan Pertanyaan (``showQuestion``)

- Menampilkan pertanyaan dan opsi jawaban sesuai dengan indeks pertanyaan saat ini.
- Jika semua pertanyaan telah dijawab, menampilkan pesan selesai dan meminta pengguna untuk memulai kembali atau keluar.

6. Fungsi Mengecek Jawaban (``checkAnswer``)

- Memeriksa jawaban yang dipilih pengguna dan menampilkan pesan yang sesuai.

- Menambah skor jika jawaban benar, menampilkan hint jika jawaban salah.
- Meningkatkan indeks pertanyaan dan memanggil `showQuestion` untuk pertanyaan berikutnya.

7. Fungsi Reset Permainan (`resetGame`)

- Mengatur ulang status permainan ke kondisi awal (indeks pertanyaan dan skor diatur ulang).
- Mengosongkan dan mengaktifkan kembali tombol jawaban.
- Mengosongkan label pertanyaan.

8. Event Handling (`actionPerformed`)

- Menangani klik tombol hint dengan menampilkan hint.
- Menangani klik tombol jawaban dengan memeriksa jawaban yang dipilih.

Step-by-Step Penjelasan

1. Membuat Kelas Pertanyaan (`Question`)

```
```java
class Question {
 private String question;
 private String[] options;
 private String correctAnswer;
 private String hint;

 public Question(String question, String[] options, String correctAnswer, String hint) {
 this.question = question;
 this.options = options;
 this.correctAnswer = correctAnswer;
 this.hint = hint;
 }

 public String getQuestion() {
 return question;
 }
}
```

```

public String[] getOptions() {
 return options;
}

public String getCorrectAnswer() {
 return correctAnswer;
}

public String getHint() {
 return hint;
}
}
...

```

- Kelas ini mendefinisikan struktur untuk pertanyaan kuis, termasuk pertanyaan, opsi jawaban, jawaban benar, dan hint.

## 2. Membuat Kelas Utama (`QuizGame`)

```

```java

public class QuizGame extends JFrame implements ActionListener {

    private static final long serialVersionUID = 1L;

    private static final Question[] questions = {

        new Question("Apa warna langit di pagi hari?", new String[]{"Biru", "Merah", "Hijau",
"Kuning"}, "Biru", "Benda berwarna biru"),

        new Question("Siapaakah presiden pertama Indonesia?", new String[]{"Soekarno", "Soeharto",
"Jokowi", "Megawati"}, "Soekarno", "Bapak Bangsa"),

    };

    private int currentQuestionIndex;

    private int score;

    private JLabel questionLabel;

    private JButton[] optionButtons;

```

```

private JButton hintButton;

private JPanel quizPanel;

private CardLayout cardLayout;

private JPanel mainPanel;


public QuizGame() {
    super("Kuis Tebak-Tebakan Ngeselin");
    setSize(400, 300);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


    cardLayout = new CardLayout();
    mainPanel = new JPanel(cardLayout);


    JPanel startPanel = new JPanel();
    startPanel.setLayout(new BoxLayout(startPanel, BoxLayout.Y_AXIS));
    startPanel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));


    JLabel welcomeLabel = new JLabel("<html><div style='text-align: center;'>Selamat Datang di  
Kuis Tebak-Tebakan Ngeselin!</div></html>", JLabel.CENTER);
    welcomeLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
    startPanel.add(welcomeLabel);
    startPanel.add(Box.createRigidArea(new Dimension(0, 20)));


    JButton startButton = new JButton("Mulai Kuis");
    startButton.setAlignmentX(Component.CENTER_ALIGNMENT);
    startButton.setPreferredSize(new Dimension(150, 50));
    startButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            cardLayout.show(mainPanel, "quizPanel");
            showQuestion();
        }
    }

```

```

});

startPanel.add(startButton);


quizPanel = new JPanel();
quizPanel.setLayout(new BorderLayout(10, 10));


questionLabel = new JLabel("", JLabel.CENTER);
questionLabel.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
quizPanel.add(questionLabel, BorderLayout.NORTH);


JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new GridLayout(2, 2, 10, 10));
buttonPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
optionButtons = new JButton[4];
for (int i = 0; i < optionButtons.length; i++) {
    optionButtons[i] = new JButton();
    optionButtons[i].setFont(new Font("Arial", Font.PLAIN, 14));
    optionButtons[i].setPreferredSize(new Dimension(150, 50));
    optionButtons[i].setMargin(new Insets(10, 10, 10, 10));
    optionButtons[i].addActionListener(this);
    buttonPanel.add(optionButtons[i]);
}
quizPanel.add(buttonPanel, BorderLayout.CENTER);


hintButton = new JButton("Hint");
hintButton.setFont(new Font("Arial", Font.PLAIN, 14));
hintButton.addActionListener(this);
quizPanel.add(hintButton, BorderLayout.SOUTH);


mainPanel.add(startPanel, "startPanel");
mainPanel.add(quizPanel, "quizPanel");

```

```

add(mainPanel);

currentQuestionIndex = 0;
score = 0;
}

private void showQuestion() {
    if (currentQuestionIndex < questions.length) {
        Question currentQuestion = questions[currentQuestionIndex];
        questionLabel.setText("<html><div style='text-align: center;'>" +
currentQuestion.getQuestion() + "</div></html>");

        String[] options = currentQuestion.getOptions();
        for (int i = 0; i < optionButtons.length; i++) {
            optionButtons[i].setText(options[i]);
            optionButtons[i].setEnabled(true);
        }
    } else {
        JOptionPane.showMessageDialog(this, "Permainan selesai! Skor Anda: " + score + "/" +
questions.length);

        int choice = JOptionPane.showConfirmDialog(this, "Main lagi?", "Kuis Selesai",
JOptionPane.YES_NO_OPTION);

        if (choice == JOptionPane.YES_OPTION) {
            resetGame();
            cardLayout.show(mainPanel, "startPanel");
        } else {
            System.exit(0);
        }
    }
}

private void resetGame() {

```

```

currentQuestionIndex = 0;
score = 0;
for (JButton button : optionButtons) {
    button.setText("");
    button.setEnabled(true);
}
questionLabel.setText("");
}

```

```

private void checkAnswer(String answer) {
    Question currentQuestion = questions[currentQuestionIndex];
    if (answer.equalsIgnoreCase(currentQuestion.getCorrectAnswer())) {
        score++;
        JOptionPane.showMessageDialog(this, "Jawaban Anda benar!");
    } else {
        JOptionPane.showMessageDialog(this, "Jawaban Anda salah! Hint: " +
currentQuestion.getHint());
    }
    currentQuestionIndex++;
    showQuestion();
}

```

@Override

```

public void actionPerformed(ActionEvent e) {
    Object source = e.getSource();
    if (source == hintButton) {
        Question currentQuestion = questions[currentQuestionIndex];
        JOptionPane.showMessageDialog(this, "Hint: " + currentQuestion.getHint());
    } else {
        for (int i = 0; i < optionButtons.length; i++) {
            if (source == optionButtons[i]) {

```

```

        checkAnswer(optionButtons[i].getText());
        break;
    }
}
}
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            QuizGame game = new QuizGame();
            game.setVisible(true

);

            game.showQuestion();
        }
    });
}
}
...

```

Penjelasan Langkah demi Langkah

1. Deklarasi dan Inisialisasi Kelas Pertanyaan (`Question`)

- Kelas `Question` digunakan untuk menyimpan detail setiap pertanyaan.
- Terdapat konstruktor untuk inisialisasi pertanyaan, opsi jawaban, jawaban benar, dan hint.

2. Inisialisasi Komponen GUI

- Kelas `QuizGame` mendeklarasikan beberapa variabel penting seperti `currentQuestionIndex`, `score`, `questionLabel`, `optionButtons`, `hintButton`, `quizPanel`, `cardLayout`, dan `mainPanel`.
- Konstruktor `QuizGame` inisialisasi frame utama dan mengatur layout menggunakan `CardLayout` untuk memungkinkan transisi antar panel.

3. Membuat Panel Start

- ``startPanel`` diatur dengan ``BoxLayout`` untuk menempatkan komponen secara vertikal.
- Label selamat datang ditempatkan di tengah dengan menggunakan ``Box.createRigidArea`` untuk memberikan ruang kosong.
- Tombol "Mulai Kuis" ditempatkan di tengah dan diberi ukuran tetap.

4. Membuat Panel Kuis

- ``quizPanel`` diatur dengan ``BorderLayout``.
- Label pertanyaan ditempatkan di bagian atas.
- Tombol opsi jawaban ditempatkan dalam ``GridLayout`` agar tertata rapi.
- Tombol hint ditempatkan di bagian bawah.

5. Menampilkan Pertanyaan (``showQuestion``)

- Mengecek apakah indeks pertanyaan saat ini masih dalam batas array pertanyaan.
- Menampilkan pertanyaan dan opsi jawaban yang sesuai.
- Jika semua pertanyaan telah dijawab, menampilkan pesan selesai dan menawarkan opsi untuk mengulang atau keluar.

6. Memeriksa Jawaban (``checkAnswer``)

- Mengecek apakah jawaban yang dipilih pengguna benar.
- Menambah skor jika jawaban benar, menampilkan hint jika jawaban salah.
- Meningkatkan indeks pertanyaan dan menampilkan pertanyaan berikutnya.

7. Reset Permainan (``resetGame``)

- Mengatur ulang indeks pertanyaan dan skor.
- Mengosongkan dan mengaktifkan kembali tombol jawaban.
- Mengosongkan label pertanyaan.

8. Menangani Event (``actionPerformed``)

- Menangani klik tombol hint dengan menampilkan hint dari pertanyaan saat ini.
- Menangani klik tombol jawaban dengan memeriksa jawaban yang dipilih.

9. Main Method

- Menggunakan `SwingUtilities.invokeLater` untuk memastikan GUI dibangun dan ditampilkan pada thread Event Dispatch.
- Memanggil `showQuestion` untuk menampilkan pertanyaan pertama setelah UI ditampilkan.

Kesimpulan

Dengan pendekatan ini, kita memastikan bahwa setiap kali permainan diulang, semua komponen kembali ke kondisi awal tanpa ada "bekas" dari permainan sebelumnya. Setiap tindakan pengguna seperti memilih jawaban atau meminta hint akan ditangani secara efisien oleh event handler.

Hasil

