**LAPORAN PEMROGRAMAN MOBILE**

**APLIKASI QUIZ VOKASI**

**Dosen Pengampu:**

I Gde Agung Sidhimantra, S.Kom., M. Kom.

**Disusun Oleh:**

Kelompok 7

1. Feis Aulia Fatchuriani (22091397053)
2. Berlin Marsyah Yustina (22091397062)
3. Tsamarah Mu’adzah Lubis (22091397074)

**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA**

**FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

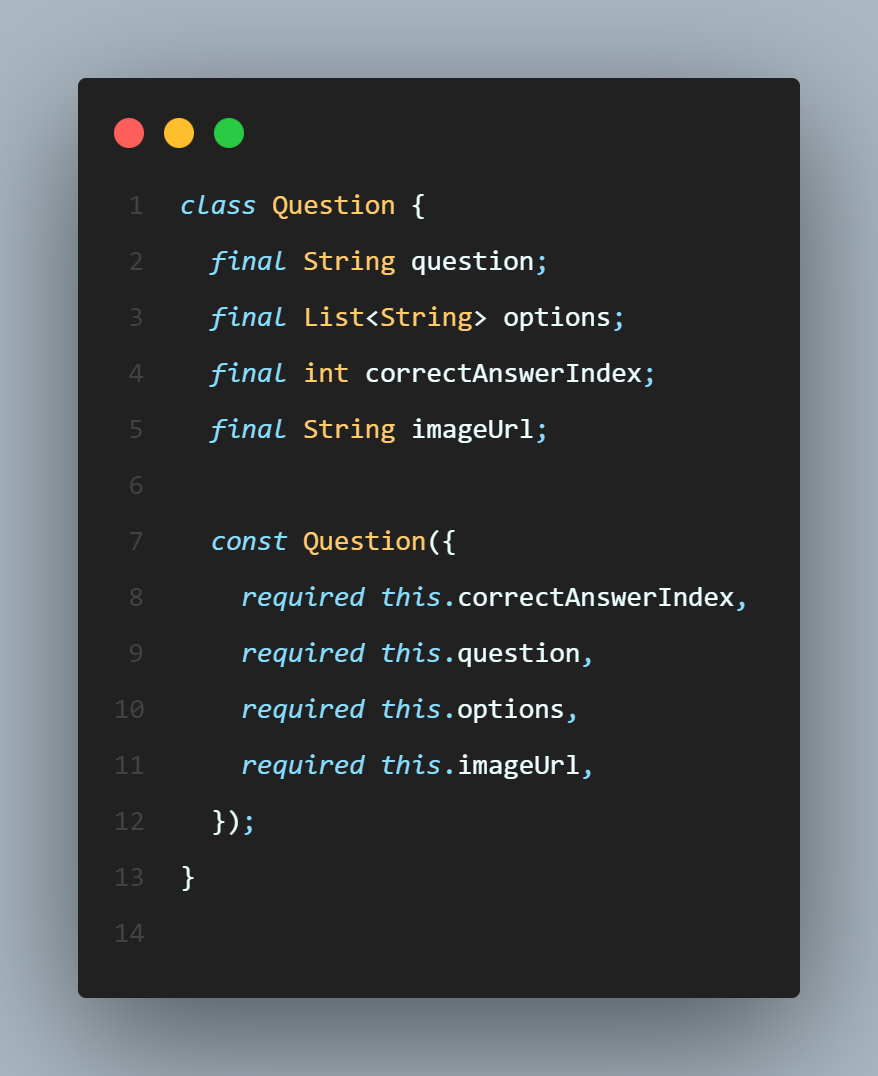
**2024**

* **Source Code Pada File ‘main.dart’**

****

Kode di atas merupakan kode untuk aplikasi Flutter sederhana yang bernama "About Vokasi". Berikut adalah penjelasannya:

* **import 'package:flutter/material.dart';** Mengimpor pustaka Flutter Material yang berisi widget-widget dasar untuk membangun UI.
* **import '/screens/quiz\_screen.dart';** Mengimpor file quiz\_screen.dart yang berisi widget QuizScreen.
* **void main** merupakan fungsi utama yang dijalankan saat aplikasi dimulai. Di dalam fungsi main, dijalankan fungsi runApp dengan argumen MainApp().
* **class MainApp** merupakan kelas **StatelessWidget** yang mendefinisikan widget utama aplikasi.
* Widget MainApp memiliki constructor default tanpa parameter.
* Method build dari kelas MainApp mengembalikan widget MaterialApp.
* Widget **MaterialApp** merupakan widget utama untuk membangun aplikasi Flutter.
* Widget MaterialApp memiliki beberapa properti:
  + **title:** Judul aplikasi.
  + **debugShowCheckedModeBanner:** Menunjukkan banner debug di sudut kanan atas.
  + **themeMode:** Mengatur mode tema (terang atau gelap).
  + **theme:** Tema yang digunakan untuk aplikasi.
  + **home:** Widget utama yang ditampilkan di aplikasi.
* **Source Code Pada Folder ‘models’**
* **question.dart**

****

Kelas ini digunakan untuk merepresentasikan sebuah pertanyaan dalam konteks suatu kuis atau aplikasi yang melibatkan kumpulan pertanyaan dan jawaban.

* **class Question { ... }** adalah deklarasi kelas Question. Kelas ini digunakan untuk merepresentasikan sebuah pertanyaan dalam konteks aplikasi.
  + **final String question;** adalah deklarasi variabel question yang bertipe String. Variabel ini akan menyimpan teks dari pertanyaan yang ingin ditanyakan.
  + **final List<String> options;** adalah deklarasi variabel options yang bertipe List<String>. Variabel ini akan menyimpan daftar opsi atau pilihan jawaban yang terkait dengan pertanyaan.
  + **final int correctAnswerIndex;**  adalah deklarasi variabel correctAnswerIndex yang bertipe int. Variabel ini akan menyimpan indeks dari opsi jawaban yang benar dalam daftar opsi.
  + **final String imageUrl;** adalah deklarasi variabel imageUrl yang bertipe String. Variabel ini akan menyimpan URL gambar terkait dengan pertanyaan. Ini bersifat opsional dan dapat diisi jika pertanyaan memiliki gambar terkait.
* **const Question({ ... });** adalah constructor dari kelas `Question`. Ini menggunakan sintaksis named parameters (`{ ... }`) yang memungkinkan pembuatan objek `Question` menyediakan nilai untuk setiap atribut dalam konstruktor secara eksplisit menggunakan nama atributnya.
  + **required this.correctAnswerIndex** Parameter ini mewajibkan penentuan indeks jawaban yang benar.
  + **required this.question** Parameter ini mewajibkan penentuan teks pertanyaan.
  + **required this.options** Parameter ini mewajibkan penentuan daftar opsi jawaban.
  + **required this.imageUrl** Parameter ini mewajibkan penentuan URL gambar terkait pertanyaan.
* **questions.dart**

****

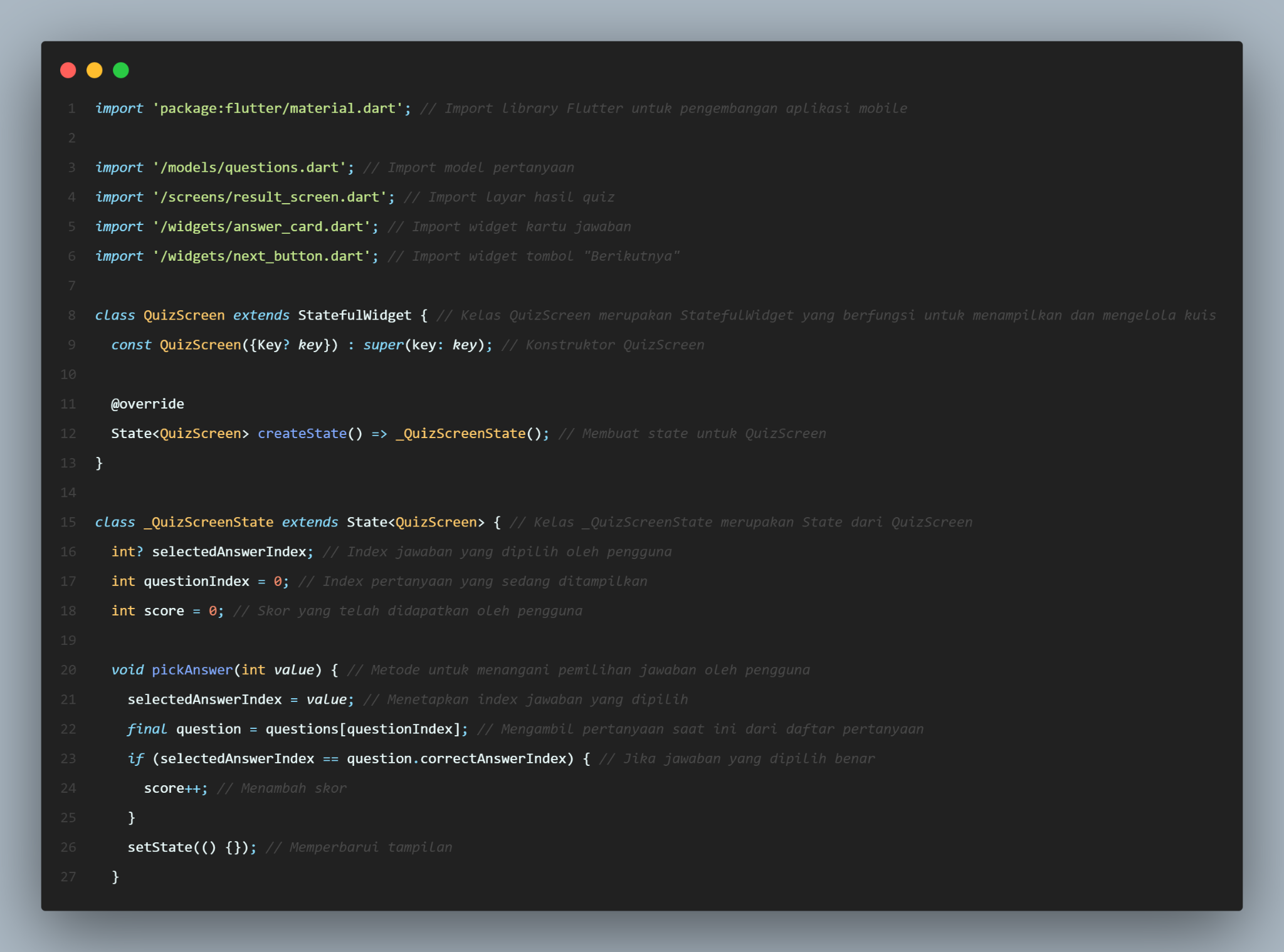
****

Kode di atas digunakan untuk membuat sebuah daftar pertanyaan untuk digunakan dalam aplikasi kuis.

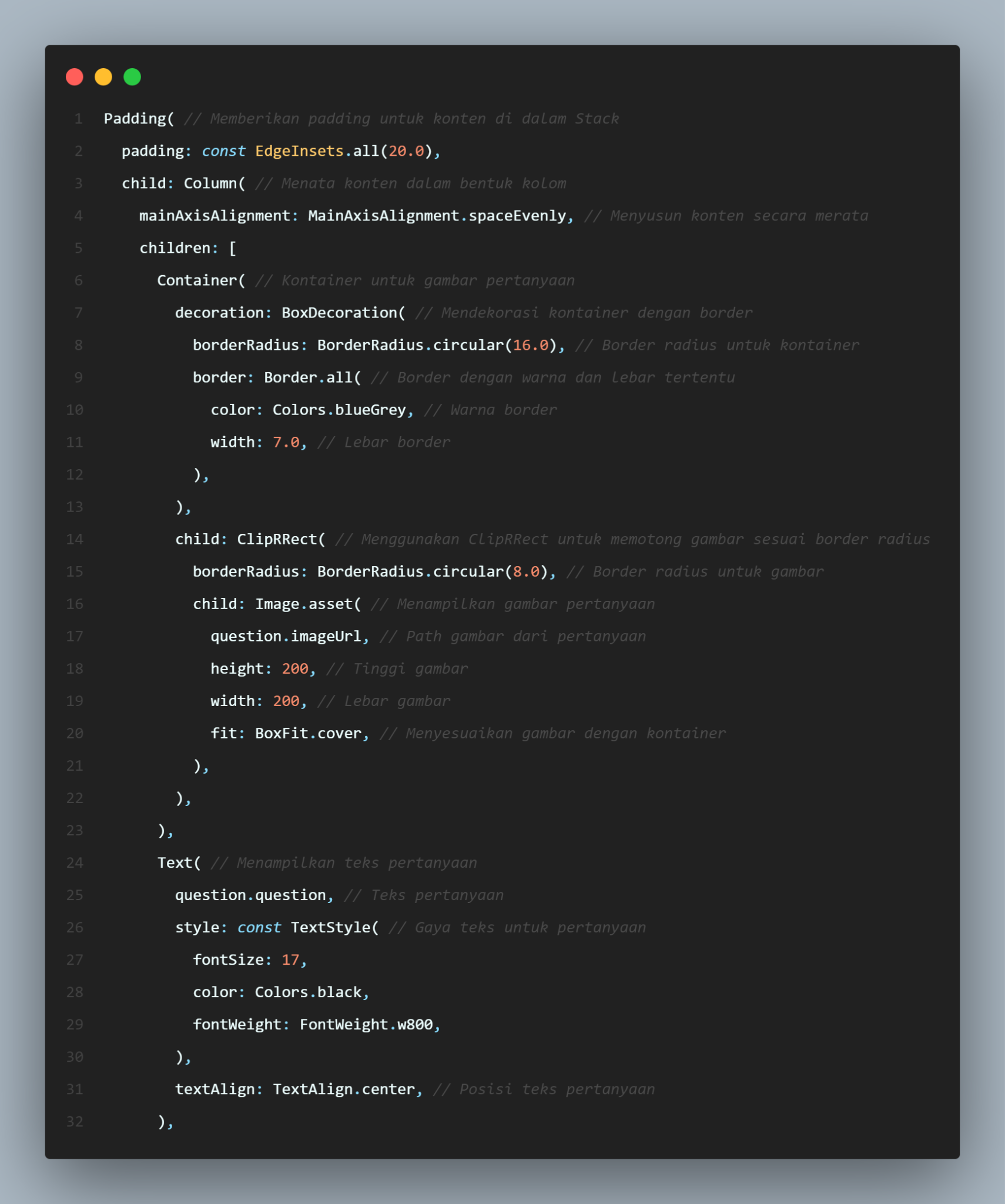
* **import 'package:quiz\_app\_tutorial/models/question.dart';** Ini adalah pernyataan untuk mengimpor kelas `Question` dari file.
* **question.dart** dalam paket **quiz\_app\_tutorial/models** digunakan untuk menghubungkan kode dengan definisi kelas Question yang digunakan untuk merepresentasikan pertanyaan dalam kuis.
* **const List<Question> questions = [ ... ];** Ini mendeklarasikan sebuah konstanta bernama **questions**, yang merupakan sebuah list dari objek-objek **Question**. Variabel ini dinyatakan sebagai **const**, yang berarti nilainya tidak dapat diubah setelah diinisialisasi.
* Setiap elemen dalam list **questions** adalah sebuah objek **Question** yang dibuat menggunakan constructor yang telah ditentukan sebelumnya dalam kelas **Question**.

Setiap objek **Question** memiliki atribut-atribut berikut:

* + **question** Ini adalah teks pertanyaan yang akan ditampilkan kepada pengguna.
  + **correctAnswerIndex** Ini adalah indeks dari opsi jawaban yang benar di dalam daftar opsi jawaban.
  + **options** Ini adalah daftar opsi jawaban yang akan ditampilkan kepada pengguna.
  + **imageUrl** Ini adalah URL gambar terkait dengan pertanyaan. Ini adalah atribut opsional, yang berarti tidak semua pertanyaan harus memiliki gambar terkait.
* **Source Code Pada Folder ‘screens’**
* **quiz\_screen.dart**

****

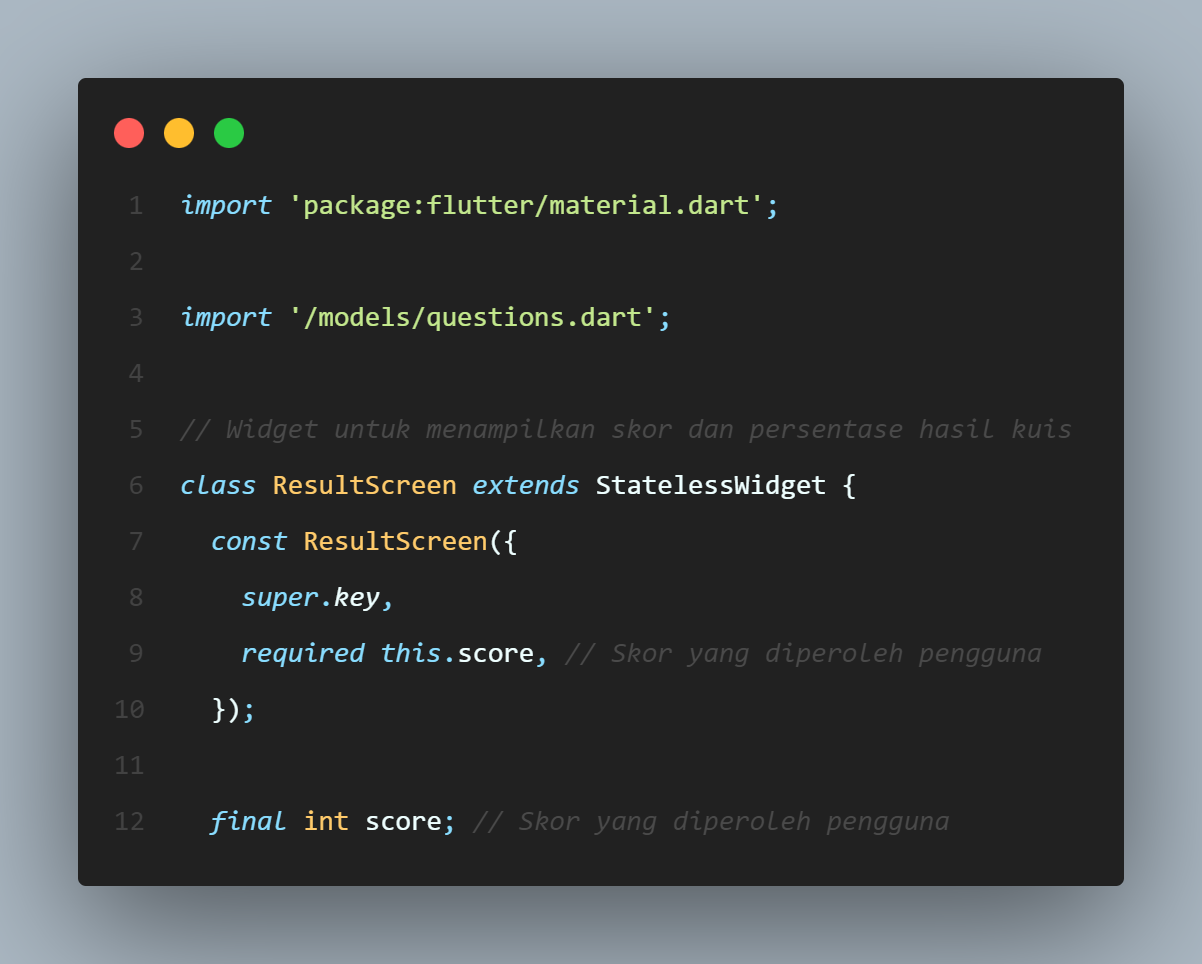
****

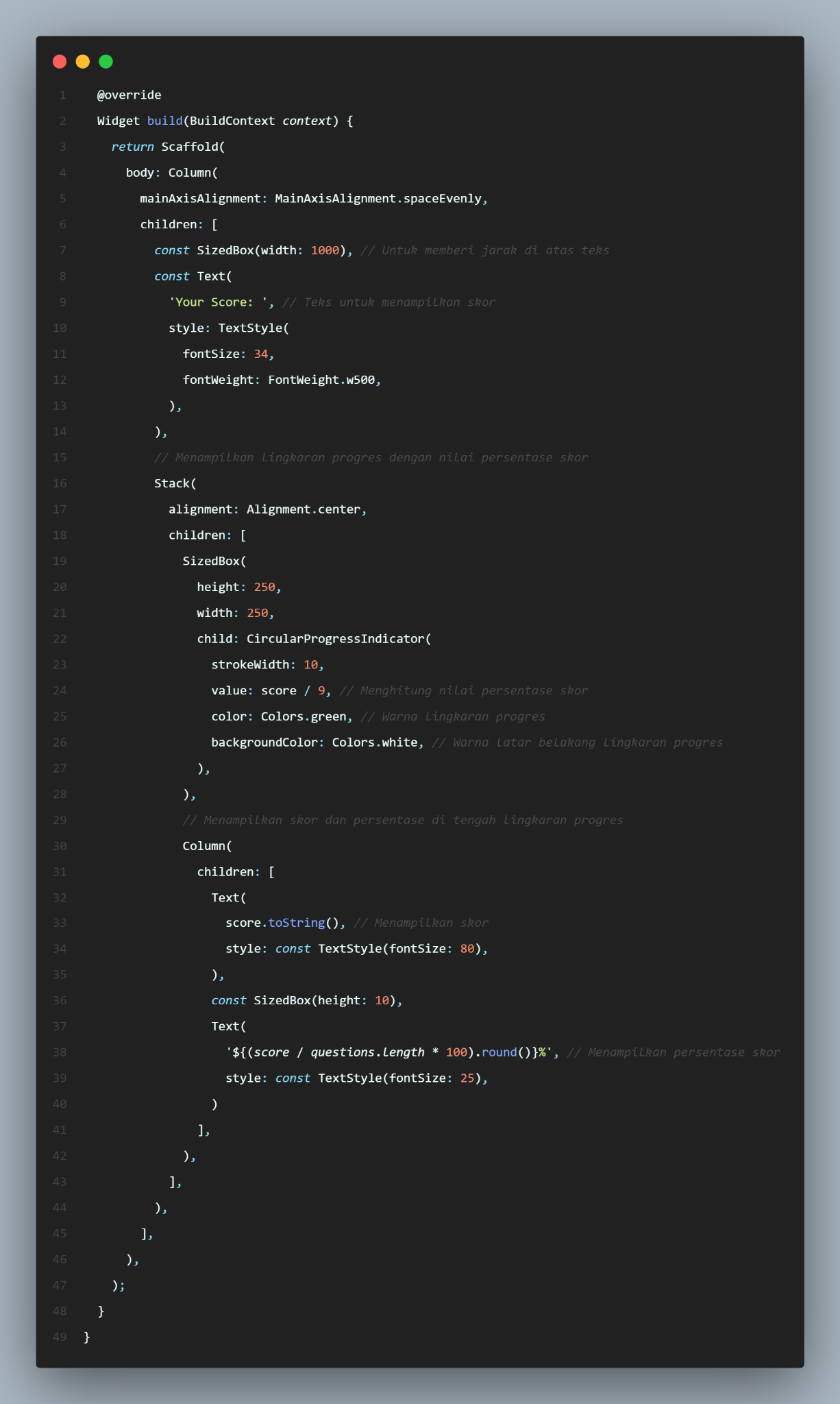
****

****

Kode di atas adalah implementasi dari layar kuis dalam aplikasi Flutter. Ini terdiri dari beberapa bagian yang mengatur tampilan dan logika dari kuis tersebut. Berikut adalah penjelasan detailnya:

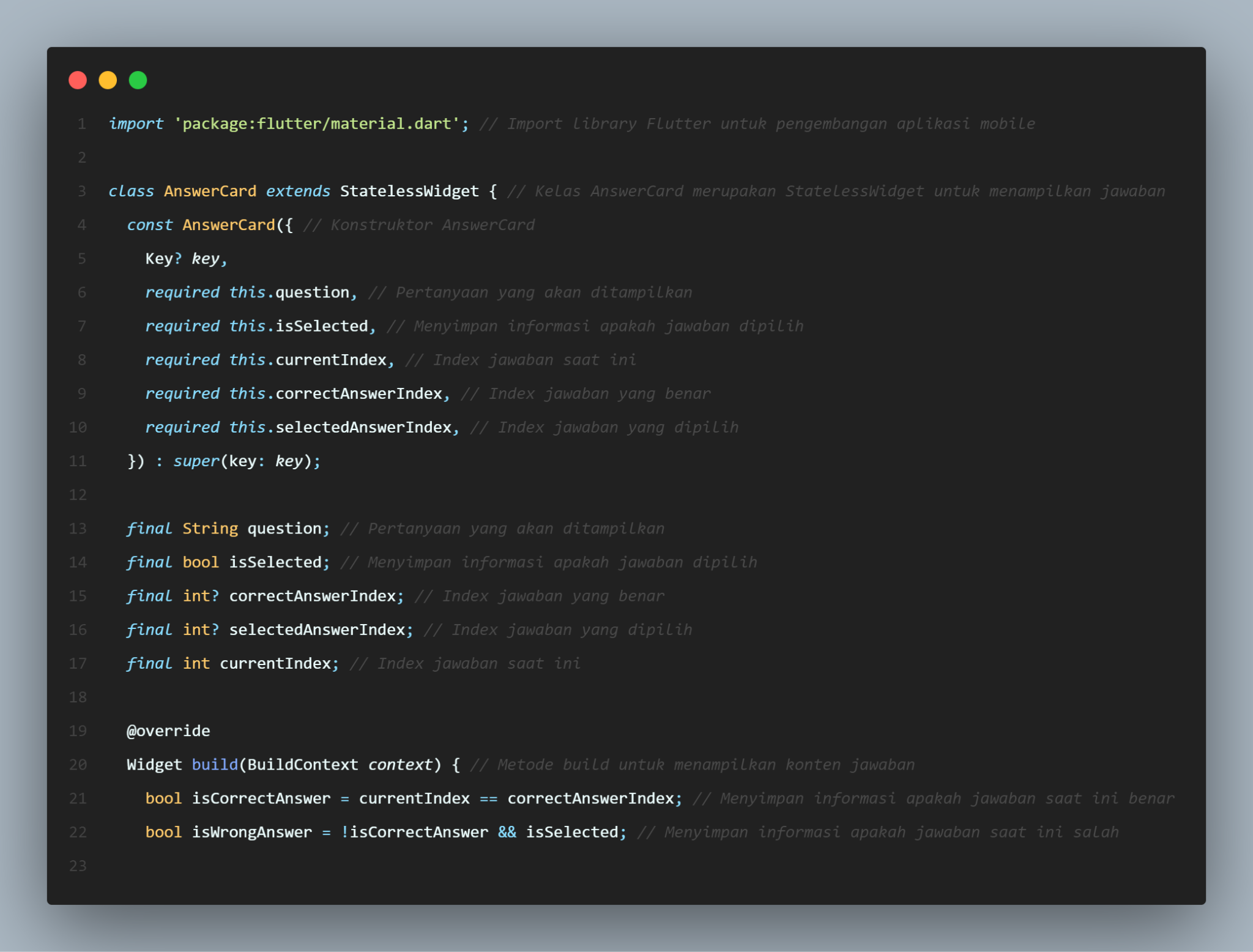
* **Import Statements:**
  + **import 'package:flutter/material.dart';** digunakan untuk mendukung penggunaan Flutter dan memberikan akses ke widget dan fungsi yang diperlukan untuk membangun antarmuka pengguna.
  + **import '/models/questions.dart';** digunakan untuk mengimpor model pertanyaan dari file **questions.dart**. Ini memberikan akses ke data pertanyaan yang akan ditampilkan dalam kuis.
  + **import '/screens/result\_screen.dart';** Mengimpor layar hasil kuis dari file **result\_screen.dart**. Ini akan digunakan untuk menampilkan hasil kuis setelah pengguna menyelesaikan semua pertanyaan.
  + **import '/widgets/answer\_card.dart';** Mengimpor widget kartu jawaban dari file `answer\_card.dart`. Widget ini akan digunakan untuk menampilkan opsi jawaban pada setiap pertanyaan.
  + **import '/widgets/next\_button.dart';** Mengimpor widget tombol "Berikutnya" dari file `next\_button.dart`. Widget ini akan digunakan untuk mengontrol navigasi antara pertanyaan dalam kuis.
* **Kelas QuizScreen:**
  + **extends StatefulWidget** Kelas ini mengambil widget stateful karena mengandung logika dinamis dan akan merender ulang ketika state berubah.
  + **const QuizScreen({Key? key}) : super(key: key);** Konstruktor untuk **QuizScreen**. Ini memungkinkan untuk mengirimkan kunci ke widget ini.
  + **@override State<QuizScreen> createState() => \_QuizScreenState();** Membuat state untuk **QuizScreen**.
* **Kelas QuizScreenState:**
* **int? selectedAnswerIndex;** Menyimpan indeks jawaban yang dipilih oleh pengguna.
* **int questionIndex = 0;** Menyimpan indeks pertanyaan yang sedang ditampilkan.
* **int score = 0;** Menyimpan skor yang telah didapatkan oleh pengguna.
  + **void pickAnswer(int value) { ... }** Metode untuk menangani pemilihan jawaban oleh pengguna. Ini memperbarui nilai **selectedAnswerIndex** dan **score** jika jawaban yang dipilih benar.
  + **void goToNextQuestion() { ... }**Metode untuk beralih ke pertanyaan berikutnya. Ini memperbarui nilai **questionIndex** dan **selectedAnswerIndex**.
  + **@override Widget build(BuildContext context) { ... }** Metode ini digunakan untuk membangun UI kuis. Ini mengandung tampilan pertanyaan, opsi jawaban, gambar, dan tombol navigasi.
* **result\_screen.dart**

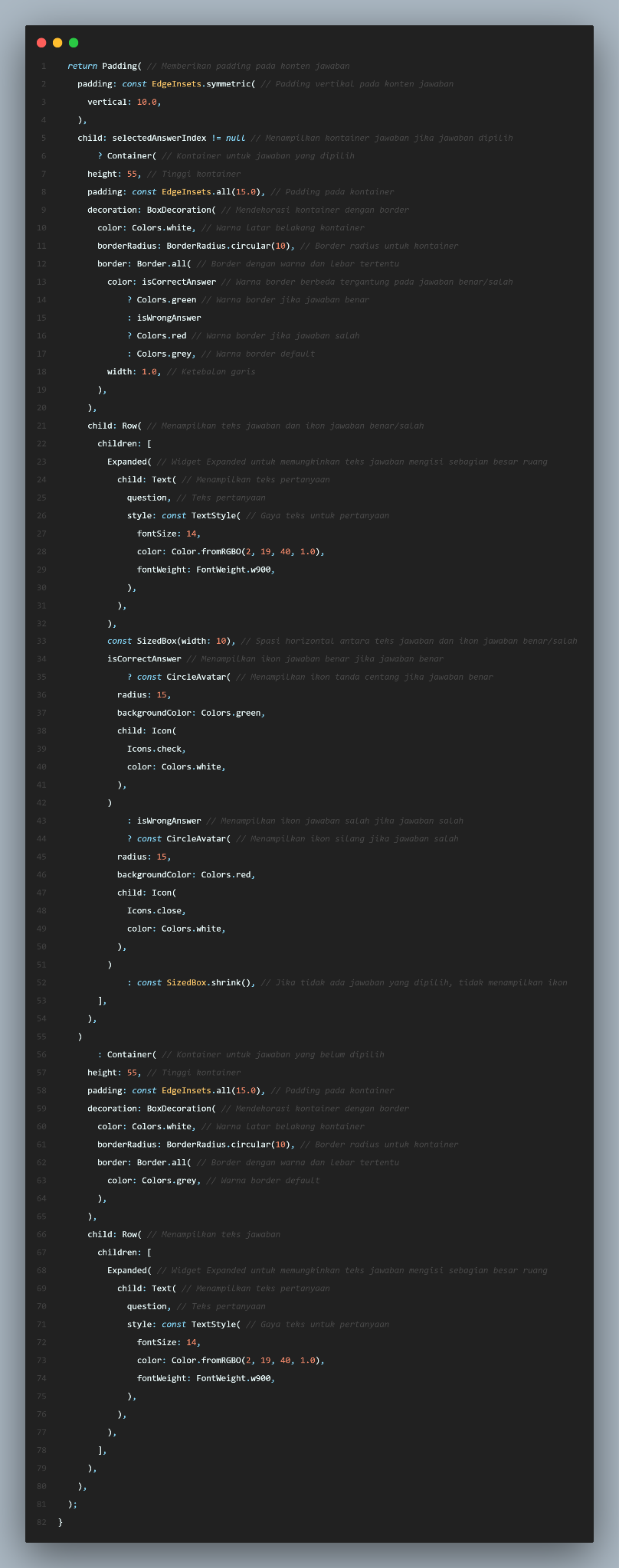
****

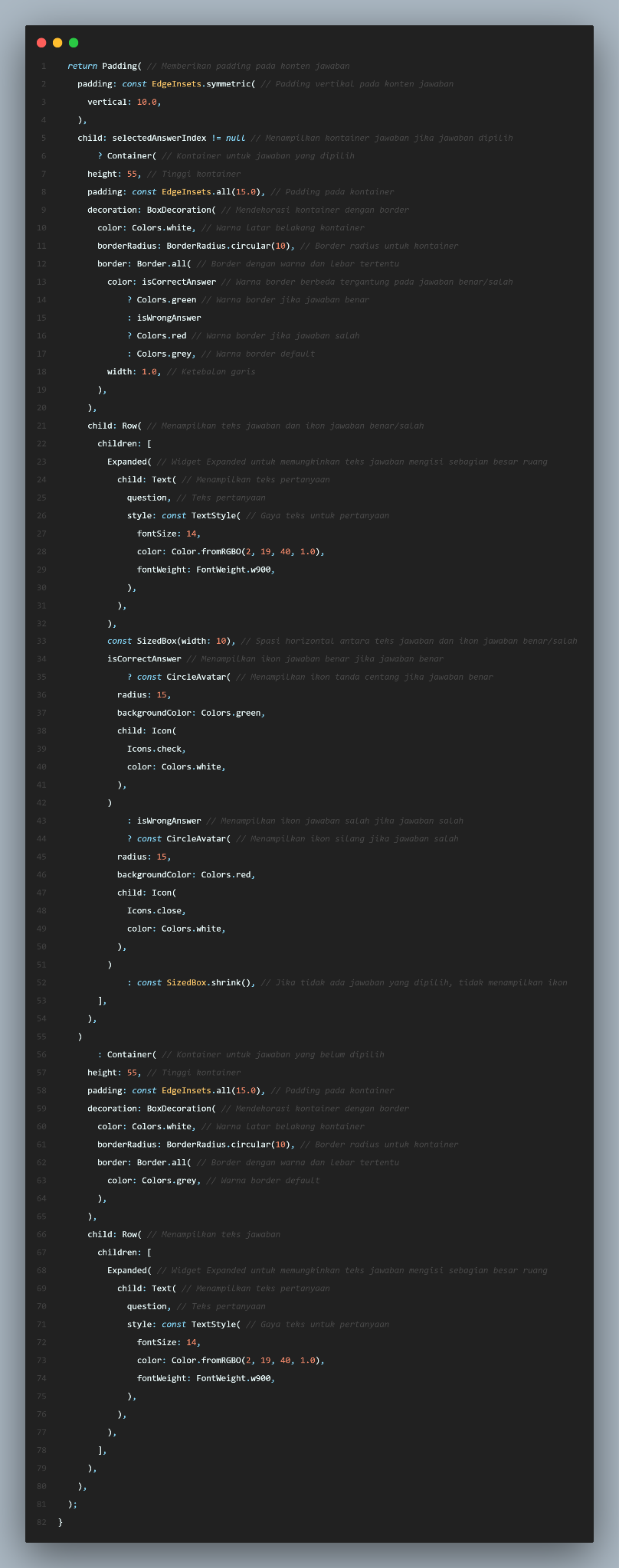
****

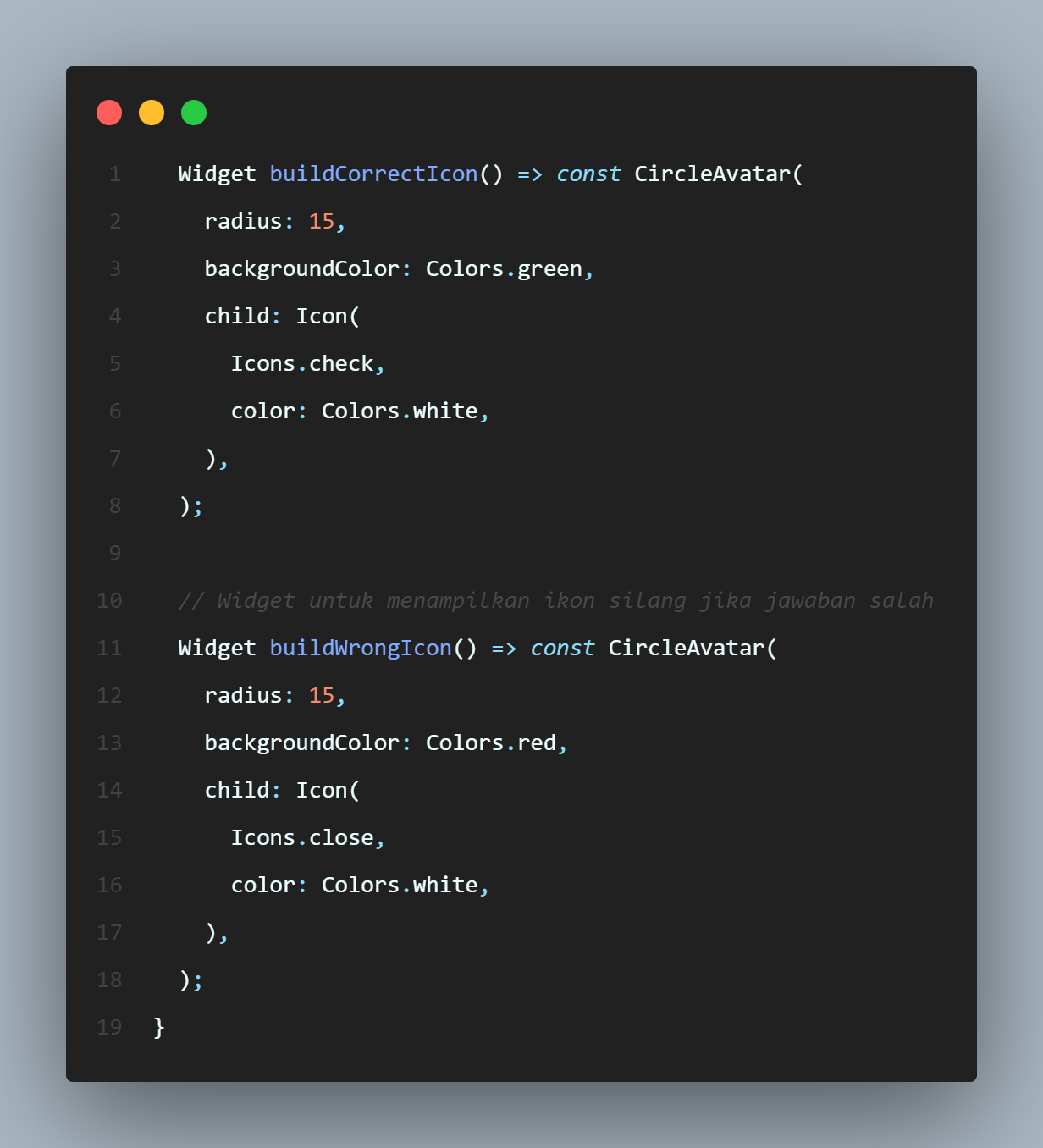
Code di atas mendemonstrasikan cara menampilkan skor dan persentase hasil kuis dalam aplikasi Flutter. Berikut penjelasannya:

* **import ‘package:flutter/material.dart’** Digunakan untuk mengakses widget-widget dasar Flutter.
* **import ’/models/questions.dart’** Digunakan untuk mengakses data pertanyaan kuis.
* **class ResultScreen** Kelas stateless yang mendefinisikan widget untuk menampilkan skor dan persentase hasil kuis.
* **conts ResultScreen({super.key, required this.score})** Konstruktor untuk kelas ResultScreen yang menerima dua parameter:
* **super.key** Key untuk widget.
* **score** Skor yang diperoleh pengguna.
* **Widget build(BuildContext context)** Metode yang membangun widget untuk menampilkan skor dan persentase hasil kuis. Berikut penjelasannya:
* **Scaffold** Widget yang menampung semua widget lain di layar.
* **Column Widget** yang mengatur widget anaknya secara vertikal.
* **SizedBox(width: 1000)** Memberi jarak di atas teks "Your Score: ".
* **Text('Your Score: ', style: TextStyle(fontSize: 34, fontWeight: FontWeight.w500))** Menampilkan teks "Your Score: " dengan style yang ditentukan.
* **Stack** Widget yang menumpuk widget anaknya di atas satu sama lain.
* **CircularProgressIndicator** Widget yang menampilkan lingkaran progres. Berikut penjelasannya:
* **strokeWidth: 10** Ketebalan lingkaran progres.
* **value: score / 9** Nilai persentase skor yang ditampilkan di lingkaran progres.
* **color: Colors.green** Warna lingkaran progres.
* **backgroundColor Colors.white** Warna latar belakang lingkaran progres.
* **Column** Widget yang mengatur widget anaknya secara vertikal.
* **Text(score.toString(), style: const TextStyle(fontSize: 80))** Menampilkan skor yang diperoleh pengguna.
* **Text('${(score / questions.length \* 100).round()}%', style: const TextStyle(fontSize: 25))** Menampilkan persentase skor yang diperoleh pengguna.
* **Source Code Pada Folder ‘widgets’**
* **answer\_card.dart**

****

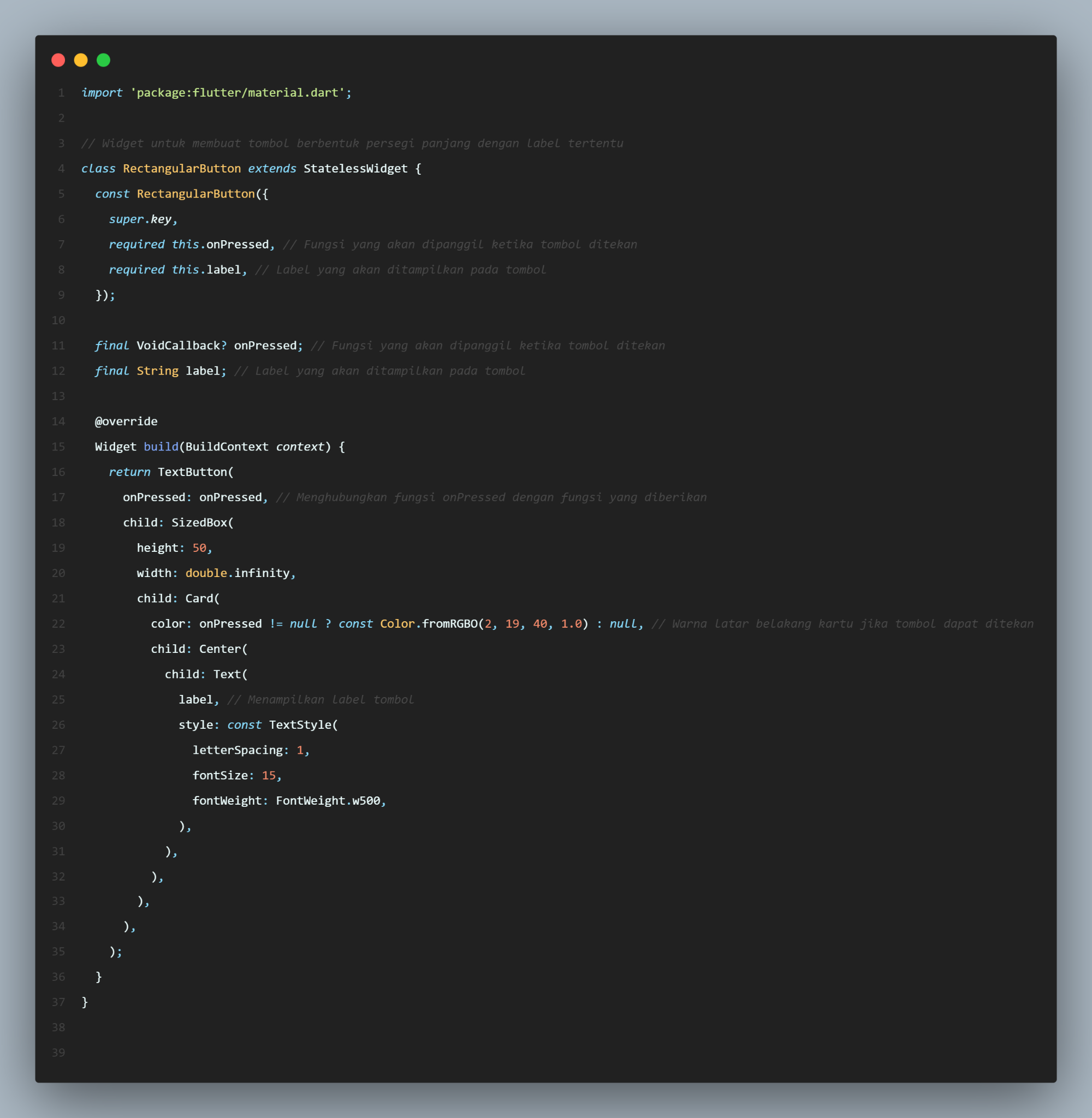
****

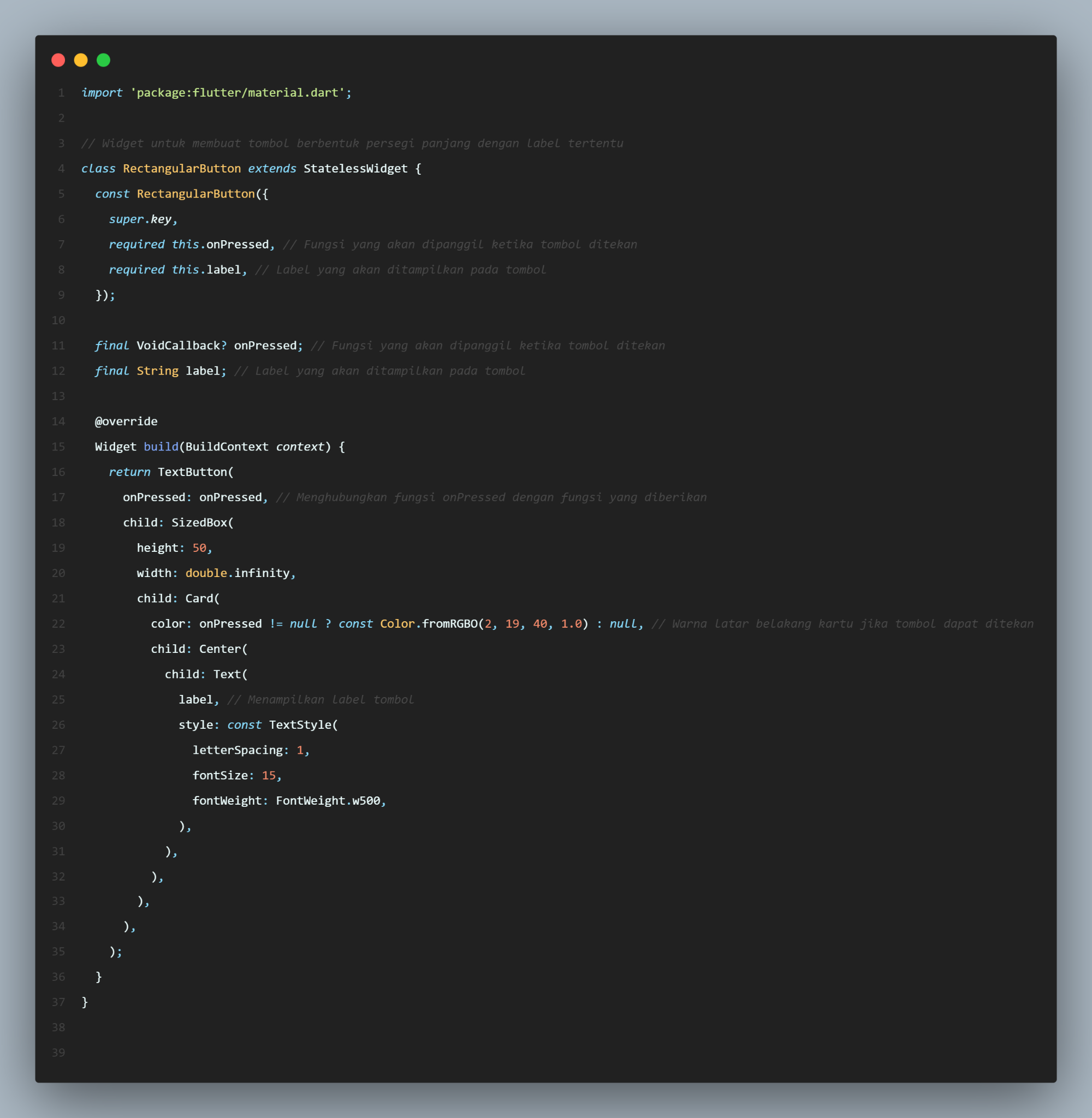
****

****

Codingan di atas adalah sebuah widget untuk menampilkan jawaban dalam aplikasi kuis Flutter. Berikut penjelasannya:

* **import “package:flutter/material.dart”** Digunakan untuk mengakses widget-widget dasar Flutter.
* **class AnswerCard** Kelas stateless yang mendefinisikan widget untuk menampilkan jawaban.
* **const AnswerCard({Key? key, required this.question, required this.isSelected, required this.currentIndex, required this.correctAnswerIndex, required this.selectedAnswerIndex})** Konstruktor untuk kelas AnswerCard yang menerima beberapa parameter:
* **Key? key:** Key untuk widget.
* **question:** Pertanyaan yang akan ditampilkan.
* **isSelected:** Menunjukkan apakah jawaban dipilih.
* **currentIndex:** Index jawaban saat ini.
* **correctAnswerIndex:** Index jawaban yang benar.
* **selectedAnswerIndex:** Index jawaban yang dipilih pengguna.
* **Widget build(BuildContext context)**: Metode yang membangun widget untuk menampilkan jawaban. Berikut penjelasannya:
* **bool isCorrectAnswer = currentIndex == correctAnswerIndex** Menentukan apakah jawaban saat ini benar.
* **bool isWrongAnswer** **= !isCorrectAnswer && isSelected** Menentukan apakah jawaban saat ini salah dan dipilih.
* **Padding** Memberikan padding pada konten jawaban.
* **selectedAnswerIndex != null** Menampilkan kontainer jawaban jika jawaban dipilih.
* **Container** untuk jawaban yang dipilih.
* **height:** Tinggi kontainer.
* **padding:** Padding pada kontainer.
* **decoration:** Dekorasi kontainer dengan border.
* **color:** Warna latar belakang kontainer.
* **borderRadius:** Border radius untuk kontainer.
* **border:** Border dengan warna dan lebar tertentu.
* **Row:** Menampilkan teks jawaban dan ikon jawaban benar/salah.
* **Expanded:** Widget Expanded untuk memungkinkan teks jawaban mengisi sebagian besar ruang.
* **Text :** Menampilkan teks pertanyaan.
* **SizedBox:** Spasi horizontal antara teks jawaban dan ikon jawaban benar/salah.
* **isCorrectAnswer:** Menampilkan ikon jawaban benar jika jawaban benar.
* **isWrongAnswer:** Menampilkan ikon jawaban salah jika jawaban salah.
* **SizedBox.shrink():** Jika tidak ada jawaban yang dipilih, tidak menampilkan ikon.
* **else:** Menampilkan kontainer untuk jawaban yang belum dipilih.
* **Container:** Kontainer untuk jawaban yang belum dipilih.
* **height:** Tinggi kontainer.
* **padding:** Padding pada kontainer.
* **decoration:** Dekorasi kontainer dengan border.
* **color:** Warna latar belakang kontainer.
* **borderRadius:** Border radius untuk kontainer.
* **border:** Border dengan warna dan lebar tertentu.
* **Row:** Menampilkan teks jawaban.
* **Expanded:** Widget Expanded untuk memungkinkan teks jawaban mengisi sebagian besar ruang.
* **Text:** Menampilkan teks pertanyaan.
* **next\_button.dart**

****

****

Codingan di atas adalah sebuah widget untuk membuat tombol berbentuk persegi panjang dengan label tertentu. Berikut penjelasannya:

* **import “package:flutter/material.dart”** Digunakan untuk mengakses widget-widget dasar Flutter.
* **class RectangularButton** Kelas stateless yang mendefinisikan widget untuk membuat tombol.
* **RectangularButton({super.key, required this.onPressed, required this.label})** Konstruktor untuk kelas RectangularButton yang menerima dua parameter:
* **super.key :** Key untuk widget.
* **onPressed :** Fungsi yang akan dipanggil ketika tombol ditekan.
* **label :** Label yang akan ditampilkan pada tombol.
* **Widget build(BuildContext context)** Metode yang membangun widget untuk menampilkan tombol. Berikut penjelasannya:
* **TextButton :** Widget tombol standar Flutter.
* **onPressed :** Menghubungkan fungsi onPressed widget dengan fungsi yang diberikan.
* **child :** Widget yang ditampilkan di dalam tombol.
* **SizedBox :** Widget untuk mengatur tinggi dan lebar tombol.
* **height :** Tinggi tombol.
* **width :** Lebar tombol (diatur menjadi double.infinity untuk mengisi seluruh lebar layar).
* **Card :** Widget untuk menampilkan latar belakang tombol.
* **color :** Warna latar belakang tombol.
* **onPressed != null:** Jika tombol dapat ditekan, atur warna latar belakang.
* **null :** Jika tombol tidak dapat ditekan, biarkan warna latar belakang transparan.
* **child :** Widget yang ditampilkan di tengah tombol.
* **Center :** Widget untuk memusatkan teks tombol.
* **Text :** Widget untuk menampilkan teks label tombol.
* **label :** Teks yang ditampilkan pada tombol.
* **style :** Gaya teks untuk label tombol.
* **Source Code Pada File ‘pubspec.yaml’**

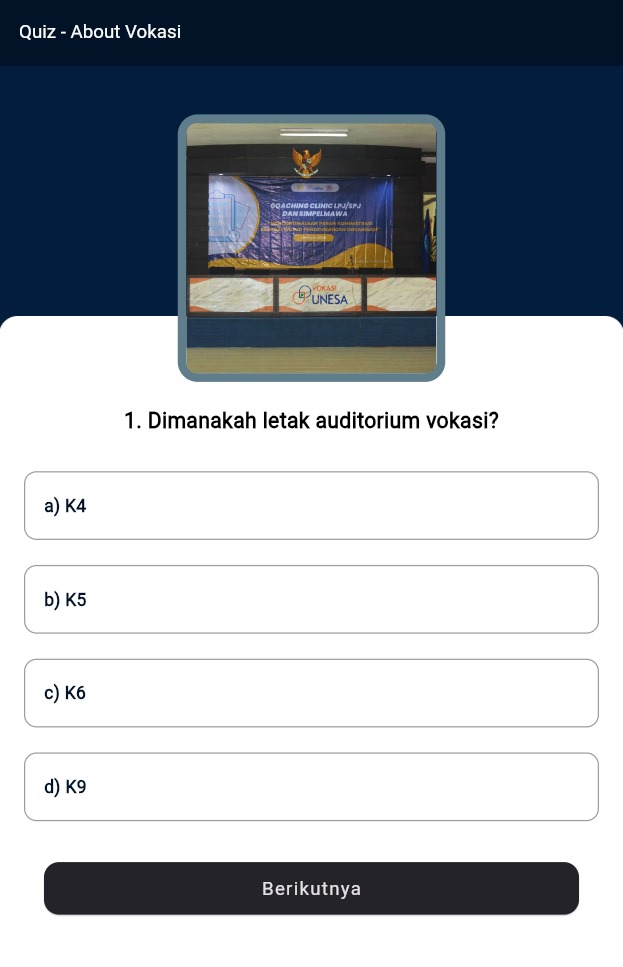
****

Kode di atas merupakan sebuah file **pubspec.yaml**, yang merupakan file konfigurasi untuk proyek Flutter. Berikut penjelasannya:

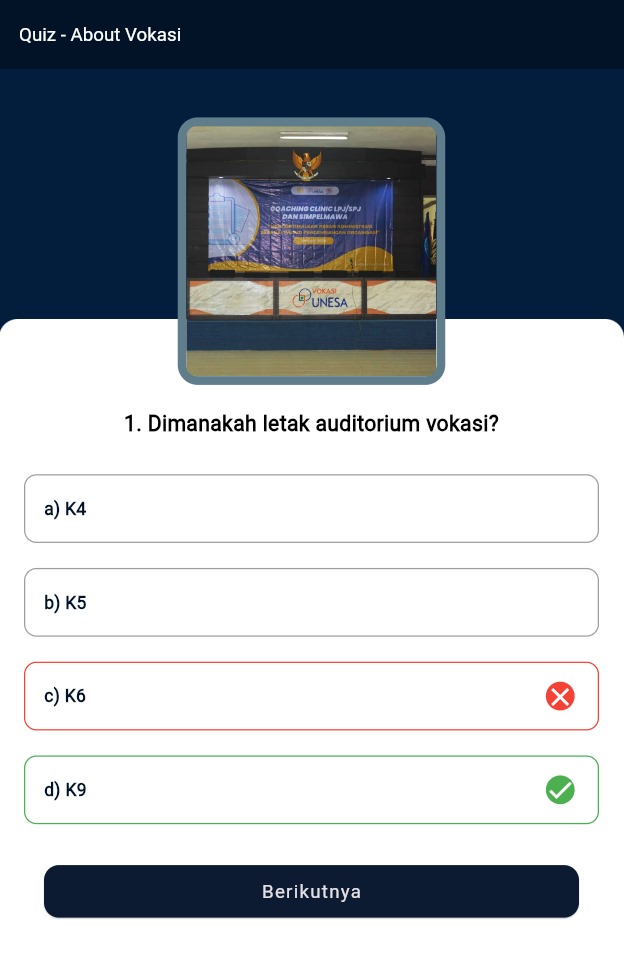
* **name: quiz\_app\_tutorial** Mendefinisikan nama proyek Flutter, dalam hal ini "quiz\_app\_tutorial".
* **description: A new Flutter project.** Deskripsi singkat tentang proyek, dalam hal ini "A new Flutter project.".
* **publish\_to: 'none'** Menunjukkan bahwa proyek ini tidak akan dipublikasikan di mana pun.
* **version: 0.1.0** Versi proyek, dalam hal ini "0.1.0".
* **environment** Menyatakan lingkungan di mana proyek dapat berjalan. Versi minimum SDK yang diperlukan adalah `3.0.2`, tetapi kurang dari `4.0.0`.
* **dependencies** Mendefinisikan dependensi proyek. Di sini, satu-satunya dependensi adalah Flutter SDK sendiri.
* **dev\_dependencies** Dependensi yang hanya diperlukan selama pengembangan, dalam hal ini **flutter\_test** dan **flutter\_lints**.
* **flutter** Konfigurasi spesifik untuk Flutter.
  + **uses-material-design: true** Menandakan bahwa proyek menggunakan desain material sebagai panduan desainnya.
  + **assets** Menentukan daftar aset yang akan dimuat bersama aplikasi, dalam hal ini direktori `assets/images/` beserta isinya.
  + **fonts** Mendefinisikan font yang akan digunakan dalam aplikasi.
  + **family: Poppins** Nama keluarga font yang digunakan, dalam hal ini "Poppins".
  + **fonts** Daftar font dalam keluarga font tersebut.
  + **asset: fonts/Poppins-Medium.ttf`** Lokasi font yang digunakan dalam proyek, dalam hal ini "fonts/Poppins-Medium.ttf".

**Tampilan Aplikasi:**

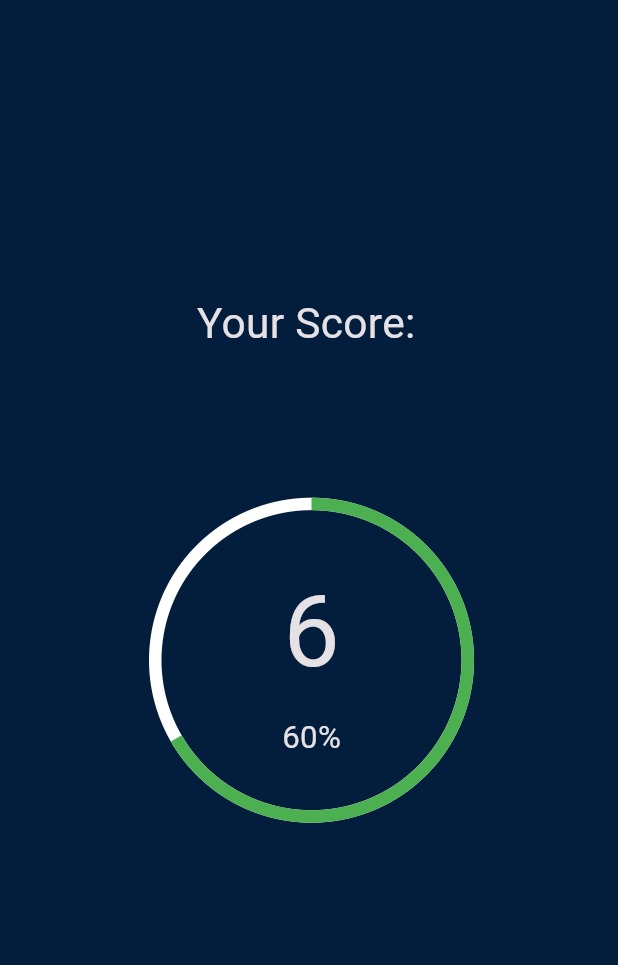
* **Tampilan Desain Quiz 1**



* **Tampilan Desain Quiz 2**



* **Tampilan Desain Skor**



**Link Github :** [**https://github.com/feisauu/flutter-quiz-vokasi**](https://github.com/feisauu/flutter-quiz-vokasi)

**Link Aplikasi :** [**https://drive.google.com/drive/folders/1E1ZHQynzJaZWEJ1M-39TOQxg8EHg9xs6**](https://drive.google.com/drive/folders/1E1ZHQynzJaZWEJ1M-39TOQxg8EHg9xs6)