Laporan Tugas 2 Pemrograman Mobile

Dosen Pengampu:

I Gede Agung Sidhimantra

Disusun Oleh:

Allan Juli Sanjaya (22091397052)

Rachmadan Nauval Hidayat Akbar (22091397072)

Andika Rafael Sitorus (232139001)

Program Studi D4 Manajemen Informatika

Fakultas Vokasi

Universitas Negeri Surabaya

Tahun Ajaran 2023/2024

Source Code halaman_quiz.dart

```dart
import 'package:flutter/material.dart';
import 'pengacakan.dart'; // Mengimpor file pengacakan.dart yang berisi fungsi acakSoal

class HalamanQuiz extends StatefulWidget {
  const HalamanQuiz({Key? key}) : super(key: key); // Deklarasi constructor HalamanQuiz dengan parameter key

  @override
  State<HalamanQuiz> createState() => _HalamanQuizState(); // Membuat state baru untuk HalamanQuiz
}
```

Berikut adalah penjelasan untuk kode di atas:

1. import 'package:flutter/material.dart
Baris ini mengimpor package Flutter yang diperlukan untuk membangun antarmuka pengguna (UI) menggunakan Material Design.

2. import 'pengacakan.dart
Baris ini mengimpor file `pengacakan.dart`, yang mungkin berisi fungsi “acakSoal” atau data terkait yang diperlukan untuk mengelola pertanyaan-pertanyaan dalam kuis.

3. class HalamanQuiz extends StatefulWidget { ... }
Ini adalah definisi dari kelas `HalamanQuiz` yang merupakan StatefulWidget. Widget-stateful digunakan ketika bagian dari UI kita memerlukan perubahan berdasarkan keadaan tertentu (state). Dalam hal ini, keadaan kuis (misalnya, nomor pertanyaan, skor, dll.) akan berubah selama pengguna menjawab pertanyaan-pertanyaan.

4. const HalamanQuiz({Key? key}) : super(key: key)
Ini adalah konstruktor untuk kelas `HalamanQuiz` yang mengambil parameter opsional `key`. Parameter `key` digunakan untuk mengidentifikasi widget ini dalam widget tree dan penting dalam pengaturan stateful widget.

5. @override State<HalamanQuiz> createState() => _HalamanQuizState()
Ini adalah metode yang digunakan untuk membuat state baru untuk widget `HalamanQuiz`. Metode ini mengembalikan objek `_HalamanQuizState` yang akan mengelola keadaan kuis.

```dart
class _HalamanQuizState extends State<HalamanQuiz> {
  List<Widget> skorMaba = []; // List untuk menyimpan skor Maba (Mahasiswa Baru)
  int nomorPertanyaan = 0; // Variabel untuk menyimpan nomor pertanyaan yang sedang ditampilkan
  int jawabanBenar = 0; // Variabel untuk menyimpan jumlah jawaban benar
  int jawabanSalah = 0; // Variabel untuk menyimpan jumlah jawaban salah
  int totalSkor = 0; // Variabel untuk menyimpan total skor

  @override
  void initState() {
    super.initState();
    acakSoal(); // Panggil fungsi acakSoal saat initState dipanggil pertama kali
  }
```

1. `class _HalamanQuizState extends State<HalamanQuiz> { ... }`
Ini adalah kelas _state_ yang terkait dengan “HalamanQuiz”. Kelas ini mengimplementasikan logika kuis dan mengatur perilaku widget berdasarkan perubahan keadaan.

2. `List<Widget> skorMaba = []`
Variabel ini digunakan untuk menyimpan widget-widget yang akan menampilkan skor setiap pertanyaan yang dijawab oleh pengguna.

3. `int nomorPertanyaan = 0`
Variabel ini menyimpan nomor pertanyaan yang sedang ditampilkan kepada pengguna.

4. `int jawabanBenar = 0`
Variabel ini menyimpan jumlah jawaban yang benar yang telah diberikan oleh pengguna.

5. `int jawabanSalah = 0`
Variabel ini menyimpan jumlah jawaban yang salah yang telah diberikan oleh pengguna.

6. `int totalSkor = 0`
Variabel ini menyimpan total skor yang diperoleh pengguna dalam kuis.

7. `@override void initState() { ... }`
Metode ini dipanggil ketika objek state pertama kali dibuat. Dalam kasus ini, `initState` digunakan untuk memanggil fungsi “acakSoal()” dari file `pengacakan.dart` untuk menginisialisasi kuis.

8. `void pilihJawaban(String jawaban) { ... }`
Metode ini digunakan untuk memproses jawaban yang dipilih oleh pengguna. Ini memeriksa apakah jawaban yang dipilih benar atau salah, menghitung skor, dan menampilkan pertanyaan berikutnya atau menampilkan skor akhir jika semua pertanyaan telah dijawab.

9. `void _tampilkanSkor() { ... }`
Metode ini menampilkan dialog AlertDialog yang menunjukkan skor akhir pengguna setelah menyelesaikan kuis. Dialog ini juga memberikan opsi untuk mengulangi kuis.

10. `@override Widget build(BuildContext context) { ... }`
Metode ini mengatur tampilan (UI) kuis. Ini membangun antarmuka pengguna dengan menampilkan pertanyaan, opsi jawaban, dan elemen-elemen lain seperti skor dan gambar.

```dart
void pilihJawaban(String jawaban) {
  setState(() {
    if (jawaban == bankSoal[nomorPertanyaan].kunciJawaban) { // Memeriksa jawaban dengan kunci jawaban yang benar
      skorMaba.add(
        const Icon(
          Icons.check, // Menambahkan icon tanda centang jika jawaban benar
          color: Colors.green,
        ),
      );
      jawabanBenar++; // Menambah jumlah jawaban benar
    } else {
      skorMaba.add(
        const Icon(
          Icons.close, // Menambahkan icon tanda silang jika jawaban salah
          color: Colors.red,
        ),
      );
      jawabanSalah++; // Menambah jumlah jawaban salah
    }

    totalSkor = jawabanBenar * 10; // Menghitung total skor berdasarkan jumlah jawaban benar

    nomorPertanyaan++; // Pindah ke pertanyaan berikutnya
    if (nomorPertanyaan < bankSoal.length) { // Jika masih ada pertanyaan tersisa
      // Kosong, karena pertanyaan selanjutnya akan ditampilkan oleh widget build
    } else {
      _tampilkanSkor(); // Memanggil fungsi tampilkanSkor jika sudah semua pertanyaan terjawab
    }
  });
}
```

Kode di atas adalah implementasi dari metode “pilihJawaban” dalam kelas “HalamanQuizState”. Berikut adalah penjelasan dari kodingan diatas:

1. void pilihJawaban(String jawaban)
Metode “pilihJawaban” digunakan untuk mengelola jawaban yang dipilih oleh pengguna.

2. setState(()
digunakan untuk memberitahu Flutter bahwa ada perubahan pada state objek dan UI perlu diperbarui.

3. if (jawaban == bankSoal[nomorPertanyaan].kunciJawaban)
Ini memeriksa apakah jawaban yang dipilih oleh pengguna sama dengan kunci jawaban yang benar. “bankSoal” mungkin merupakan variabel atau objek yang menyimpan daftar pertanyaan dan jawaban di kuis.

4. skorMaba.add(
const Icon(
Icons.check,
color: Colors.green,
),
);

Jika jawaban benar, maka sebuah ikon centang (checkmark) akan ditambahkan ke dalam “skorMaba”. Ikon ini akan ditampilkan di UI untuk menunjukkan bahwa jawaban benar.

5. `jawabanBenar++; // Menambah jumlah jawaban benar`
Jumlah jawaban benar akan ditambahkan satu jika jawaban yang dipilih benar.

6. `} else {`
Jika jawaban yang dipilih salah, eksekusi akan masuk ke bagian ini.

7.
```
skorMaba.add(
  const Icon(
    Icons.close,
    color: Colors.red,
  ),
);
```
Jika jawaban salah, maka sebuah ikon silang (cross) akan ditambahkan ke dalam “skorMaba”. Ikon ini akan ditampilkan di UI untuk menunjukkan bahwa jawaban salah.

8. `jawabanSalah++; // Menambah jumlah jawaban salah`
Jumlah jawaban salah akan ditambahkan satu jika jawaban yang dipilih salah.

9. `totalSkor = jawabanBenar * 10; // Menghitung total skor berdasarkan jumlah jawaban benar`
Total skor dihitung berdasarkan jumlah jawaban benar yang telah dijawab oleh pengguna. Dalam contoh ini, setiap jawaban benar bernilai 10 poin.

10. `nomorPertanyaan++; // Pindah ke pertanyaan berikutnya`
Setelah mengelola jawaban, nomor pertanyaan akan diinkrementasi untuk menunjuk ke pertanyaan berikutnya dalam daftar pertanyaan.

11.
```
if (nomorPertanyaan < bankSoal.length) {
  // Jika masih ada pertanyaan tersisa, kode tidak melakukan apa-apa di sini
} else {
  _tampilkanSkor(); // Memanggil fungsi tampilkanSkor jika sudah semua pertanyaan terjawab
}
```
Kode ini memeriksa apakah masih ada pertanyaan yang tersisa dalam kuis. Jika semua pertanyaan telah dijawab, maka fungsi “_tampilkanSkor()” akan dipanggil untuk menampilkan skor akhir kuis.

Dengan demikian, kode “pilihJawaban” ini bertanggung jawab untuk mengelola logika jawaban benar atau salah dari pengguna dalam kuis dan memperbarui UI sesuai dengan hasil jawaban yang diberikan.

```dart
void _tampilkanSkor() {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        backgroundColor: Colors.yellowAccent, // Warna latar belakang dialog
        title: const Text('Skor Akhir'), // Judul dialog
        content: Column(
          mainAxisSize: MainAxisSize.min,
          children: [
            Text('Benar: $jawabanBenar', style: const TextStyle(color: Colors.black)), // Menampilkan jumlah jawaban benar
            Text('Salah: $jawabanSalah', style: const TextStyle(color: Colors.black)), // Menampilkan jumlah jawaban salah
            Text('Total Skor: $totalSkor', style: const TextStyle(color: Colors.black)), // Menampilkan total skor
          ],
        ),
        actions: [
          TextButton(
            onPressed: () {
              setState(() {
                nomorPertanyaan = 0; // Reset nomor pertanyaan
                jawabanBenar = 0; // Reset jumlah jawaban benar
                jawabanSalah = 0; // Reset jumlah jawaban salah
                skorMaba.clear(); // Menghapus skor Maba
                totalSkor = 0; // Reset total skor
                acakSoal(); // Panggil fungsi acakSoal lagi untuk mengacak urutan pertanyaan berikutnya
              });
              Navigator.of(context).pop(); // Tutup dialog
            },
            child: const Text('Ulangi'), // Tombol untuk mengulangi kuis
          ),
        ],
      );
    },
  );
}
```

Kode tersebut adalah implementasi dari metode “_tampilkanSkor” dalam kelas “_HalamanQuizState”. Berikut adalah penjelasan per bagiannya:

1. showDialog(...)
   Ini adalah metode untuk menampilkan dialog di atas konten aplikasi saat ini. Dialog ini adalah AlertDialog, yang merupakan dialog standar dengan judul, konten, dan tombol aksi.

2. context: context
   Parameter pertama dalam “showDialog” menentukan konteks di mana dialog akan ditampilkan. Konteks ini menyediakan akses ke berbagai sumber daya aplikasi dan pengaturan.

3. builder: (BuildContext context) { ... }
   Ini adalah fungsi pembangun (builder) yang digunakan untuk membangun tampilan dialog. Fungsi ini menerima konteks sebagai parameter dan mengembalikan widget AlertDialog.

4. AlertDialog(...)
   Ini adalah widget AlertDialog yang menampilkan informasi skor akhir setelah pengguna menyelesaikan kuis.

5. backgroundColor: Colors.yellowAccent
   Mengatur warna latar belakang dialog menjadi kuning muda.

6. `title: const Text('Skor Akhir')`
   Menentukan judul dialog sebagai “Skor Akhir”.

7. `content: Column(...)`
   Bagian ini berisi konten dialog, yang terdiri dari tiga teks: jumlah jawaban benar, jumlah jawaban salah, dan total skor.

8. `actions: [...]`
   Ini adalah daftar aksi yang ditampilkan di bagian bawah dialog. Di sini, kita hanya memiliki satu aksi, yaitu tombol 'Ulangi' yang akan mengulangi kuis.

9. `onPressed: () { ... }`
   Ini adalah fungsi yang akan dipanggil ketika tombol “Ulangi” ditekan. Di dalamnya, kita melakukan reset nilai-nilai terkait kuis seperti nomor pertanyaan, jumlah jawaban benar, jumlah jawaban salah, dan total skor. Kemudian, kita memanggil fungsi `acakSoal()` untuk mengacak urutan pertanyaan berikutnya.

10. `Navigator.of(context).pop()`
    Kode ini menutup dialog setelah pengguna menekan tombol “Ulangi”, sehingga kembali ke konten utama aplikasi.

Dengan demikian, “_tampilkanSkor” bertanggung jawab untuk menampilkan dialog AlertDialog dengan informasi skor akhir kuis, memberikan opsi untuk mengulangi kuis, dan melakukan reset terkait kuis jika pengguna memilih untuk mengulang kuis.

```dart
@override
Widget build(BuildContext context) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: <Widget>[
      Container(
        color: Colors.white,
        padding: const EdgeInsets.all(25.0),
        child: Column(
          children: [
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: [
                Text(
                  'Benar: $jawabanBenar', // Menampilkan jumlah jawaban benar
                  style: const TextStyle(
                    fontSize: 20.0,
                    color: Colors.green,
                  ),
                ),
                Text(
                  'Salah: $jawabanSalah', // Menampilkan jumlah jawaban salah
                  style: const TextStyle(
                    fontSize: 20.0,
                    color: Colors.red,
                  ),
                ),
              ],
            ),
            const SizedBox(height: 10),
            Text(
              bankSoal[nomorPertanyaan].pertanyaan, // Menampilkan pertanyaan
              textAlign: TextAlign.center,
              style: const TextStyle(
                fontSize: 25.0,
                color: Colors.black87,
              ),
            ),
```

1. @override Widget build(BuildContext context) { ... }

Ini adalah metode `build` yang merupakan bagian dari Flutter Framework dan digunakan untuk membangun tampilan (UI) widget. Metode ini akan dipanggil ketika widget perlu dirender atau diperbarui di layar.

2. return Column(...)

Metode `build` mengembalikan widget Column yang merupakan widget tata letak untuk menempatkan widget-child secara vertikal.

3. crossAxisAlignment: CrossAxisAlignment.stretch

Properti `crossAxisAlignment` pada Column digunakan untuk mengatur cara widget-child dalam Column akan ditempatkan secara horizontal. `CrossAxisAlignment.stretch` akan memperluas widget-child secara horizontal agar sesuai dengan lebar Column.

4. children: <Widget>[...]
Properti `children` dari Column berisi daftar widget-child yang akan ditampilkan dalam Column.

5. Container(...)
Ini adalah widget Container yang digunakan untuk mengelompokkan widget lain dan memberikan dekorasi seperti warna latar belakang dan padding.

6. color: Colors.white
Properti `color` pada Container digunakan untuk menentukan warna latar belakangnya, dalam hal ini adalah putih.

7. padding: const EdgeInsets.all(25.0)
Properti `padding` pada Container menentukan jarak (padding) di sekitar konten dalam Container. `EdgeInsets.all(25.0)` memberikan padding sebesar 25.0 pada semua sisi (atas, bawah, kiri, kanan).

8. child: Column(...)
Widget `child` dari Container adalah Column lain yang digunakan untuk menempatkan widget-child secara vertikal di dalam Container.

9. children: [...]
Properti `children` dari Column dalam Container berisi daftar widget-child yang akan ditampilkan dalam Column ini.

10. Row(...)
Ini adalah widget Row yang digunakan untuk menempatkan widget-child secara horizontal.

11. mainAxisAlignment: MainAxisAlignment.spaceBetween
Properti `mainAxisAlignment` pada Row menentukan cara widget-child diatur secara horizontal. `MainAxisAlignment.spaceBetween` akan membuat ruang kosong antara widget-child sehingga mereka tersebar merata di sepanjang Row.

12. children: [...]
Properti `children` dari Row berisi daftar widget-child yang akan ditampilkan secara horizontal.

13. Text(...)
Ini adalah widget Text yang digunakan untuk menampilkan teks.

14. style: const TextStyle(...)
Properti `style` pada Text digunakan untuk menentukan gaya teks seperti ukuran font, warna, dll.

15. 'Benar: $jawabanBenar'
Ini adalah teks yang akan ditampilkan di dalam widget Text. `$jawabanBenar` adalah ekspresi interpolasi string yang menggabungkan nilai variabel `jawabanBenar` ke dalam teks.

16. bankSoal[nomorPertanyaan].pertanyaan
Ini adalah ekspresi untuk mengakses pertanyaan dari daftar `bankSoal` berdasarkan `nomorPertanyaan` yang sedang ditampilkan.

17. textAlign: TextAlign.center
Properti `textAlign` pada widget Text digunakan untuk mengatur posisi teks dalam widget, dalam hal ini ditengah (center).

18. style: const TextStyle(...)
Ini adalah properti `style` pada widget Text yang digunakan untuk menentukan gaya teks seperti ukuran font, warna, dll.

19. fontSize: 25.0
Properti `fontSize` pada TextStyle digunakan untuk menentukan ukuran font teks, dalam hal ini adalah 25.0.

20. color: Colors.black87
Properti `color` pada TextStyle digunakan untuk menentukan warna teks, dalam hal ini adalah hitam (#000000) dengan opasitas 87%.

Keseluruhan, kode tersebut digunakan untuk membangun tampilan UI untuk menampilkan informasi pertanyaan, jumlah jawaban benar, dan jumlah jawaban salah dalam kuis. Widget Column, Row, dan Text digunakan untuk mengatur tata letak dan menampilkan teks dengan gaya tertentu.

```
            const SizedBox(height: 6),
            Image.asset(
              bankSoal[nomorPertanyaan].gambar, // Menampilkan gambar terkait pertanyaan
              width: 150,
              height: 150,
              fit: BoxFit.contain,
            ),
          ],
        ),
      ),
      const SizedBox(height: 10),
      Column(
        children: bankSoal[nomorPertanyaan].jawaban.map((jawaban) {
          return Padding(
            padding: const EdgeInsets.symmetric(vertical: 5.0),
            child: ElevatedButton(
              onPressed: () {
                pilihJawaban(jawaban); // Memilih jawaban
              },
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.lightBlue, // Warna tombol jawaban
                fixedSize: const Size(400, 50), // Ukuran tombol jawaban
              ),
```

1. const SizedBox(height: 6)
Widget `SizedBox` digunakan untuk memberikan ruang vertikal sebesar 6 piksel antara widget sebelumnya (Text) dan widget berikutnya (Image). Properti `height: 6` menentukan tinggi widget `SizedBox`.

2. Image.asset(
```
    bankSoal[nomorPertanyaan].gambar, // Menampilkan gambar terkait pertanyaan
    width: 150,
    height: 150,
    fit: BoxFit.contain,
  ),
```
Widget `Image.asset` digunakan untuk menampilkan gambar yang diambil dari lokasi yang diberikan oleh `bankSoal[nomorPertanyaan].gambar`. Properti `width` dan `height` menentukan lebar dan tinggi gambar. `fit: BoxFit.contain` digunakan untuk memastikan gambar sesuai dengan ukuran yang ditentukan (150x150) tanpa merubah proporsi gambar.

3. const SizedBox(height: 10)
Widget `SizedBox` kedua digunakan untuk memberikan ruang vertikal sebesar 10 piksel antara gambar dan widget Column berikutnya.

4. Column(
```
    children: bankSoal[nomorPertanyaan].jawaban.map((jawaban) {
      return Padding(
        padding: const EdgeInsets.symmetric(vertical: 5.0),
        child: ElevatedButton(
          onPressed: () {
            pilihJawaban(jawaban); // Memilih jawaban
          },
          style: ElevatedButton.styleFrom(
            backgroundColor: Colors.lightBlue, // Warna tombol jawaban
            fixedSize: const Size(400, 50), // Ukuran tombol jawaban
          ),
        ),
      );
    }).toList(),
  ),
```
- Widget “Column” ini berisi daftar jawaban untuk pertanyaan yang ditampilkan pada nomor tertentu. Widget “children” pada Column menggunakan “map” untuk mengonversi daftar jawaban (“bankSoal[nomorPertanyaan].jawaban”) menjadi daftar widget “ElevatedButton”. Setiap jawaban akan ditampilkan dalam “ElevatedButton” dengan konfigurasi tertentu seperti fungsi yang dipanggil saat tombol ditekan (“onPressed”), warna latar belakang (“backgroundColor”), dan ukuran tetap (“fixedSize”). “ElevatedButton” digunakan untuk membuat tombol responsif dengan efek visual naik ketika ditekan.

- “Padding” digunakan untuk memberikan ruang di sekitar tombol jawaban. Properti symmetric(vertical: 5.0) memberikan ruang vertikal sebesar 5 piksel di atas dan di bawah tombol.
- “onPressed: () { pilihJawaban(jawaban); }” digunakan untuk memanggil fungsi “pilihJawaban” dengan parameter jawaban yang dipilih saat tombol jawaban ditekan.
- “style: ElevatedButton.styleFrom(...)” digunakan untuk mengatur gaya tombol seperti warna latar belakang menggunakan “Colors.lightBlue” dan ukuran tetap tombol menggunakan `fixedSize`.
- “toList()” digunakan untuk mengonversi hasil dari “map” ke dalam bentuk list karena `children` dari Column memerlukan daftar widget.

Keseluruhan, kode tersebut digunakan untuk menampilkan gambar terkait pertanyaan dan daftar jawaban dalam bentuk tombol “ElevatedButton”. Setiap tombol jawaban akan merespons saat ditekan dengan memanggil fungsi “pilihJawaban” dan mengubah warna latar belakang tombol.

```
                child: Row(
                  mainAxisAlignment: MainAxisAlignment.spaceBetween,
                  children: [
                    Text(
                      jawaban, // Teks jawaban
                      style: const TextStyle(
                        fontSize: 20.0,
                        color: Colors.black,
                      ),
                    ),
                    if (skorMaba.length > nomorPertanyaan) skorMaba[nomorPertanyaan], // Menampilkan ikon tanda benar/salah jika sudah dipilih
                  ],
                ),
              ),
            );
          }).toList(),
        ),
        const SizedBox(height: 10),
      ],
    );
  }
}
```

1. child: Row(
   mainAxisAlignment: MainAxisAlignment.spaceBetween,
   children: [
     Text(
       jawaban, // Teks jawaban
       style: const TextStyle(
         fontSize: 20.0,
         color: Colors.black,
       ),
     ),
     if (skorMaba.length > nomorPertanyaan) skorMaba[nomorPertanyaan], // Menampilkan ikon tanda benar/salah jika sudah dipilih
   ],
 ),

Widget “Row” ini digunakan untuk menampilkan teks jawaban bersama dengan ikon tanda benar/salah jika jawaban tersebut sudah dipilih.

- “mainAxisAlignment: MainAxisAlignment.spaceBetween” digunakan untuk mengatur tata letak widget di dalam Row. Dalam hal ini, widget-child akan tersusun merata di sepanjang ruang yang tersedia.

- “children” dari Row berisi dua widget: “Text” untuk menampilkan teks jawaban dan ekspresi kondisional untuk menampilkan ikon tanda benar/salah jika jawaban sudah dipilih.

- “Text” menampilkan teks jawaban dari variabel `jawaban` dengan gaya tertentu seperti ukuran font 20 dan warna hitam.

- Ekspresi kondisional “if (skorMaba.length > nomorPertanyaan)” digunakan untuk memeriksa apakah sudah ada jawaban dipilih untuk pertanyaan ini. Jika ya, maka widget `skorMaba[nomorPertanyaan]` akan ditampilkan. Ini akan menampilkan ikon tanda centang (Icons.check) jika jawaban benar atau ikon tanda silang (Icons.close) jika jawaban salah.

2. }).toList(),
“toList()” digunakan untuk mengonversi hasil dari “map” ke dalam bentuk list karena `children` dari Column memerlukan daftar widget.

3. const SizedBox(height: 10),Widget `SizedBox
digunakan untuk memberikan ruang vertikal sebesar 10 piksel antara widget `Row` dan widget `Column` berikutnya.

Keseluruhan, kode tersebut bertanggung jawab untuk menampilkan daftar jawaban dalam bentuk tombol responsif (“ElevatedButton”) di dalam widget “Column”. Setiap tombol jawaban memiliki teks jawaban dan ikon tanda benar/salah jika jawaban tersebut sudah dipilih. Jika jawaban belum dipilih, maka hanya teks jawaban yang akan ditampilkan. Widget ini memberikan interaksi kepada pengguna untuk memilih jawaban dan menampilkan status jawaban yang benar atau salah setelah dipilih.

Source code untuk main.dart

```dart
import 'package:flutter/material.dart';
import 'halaman_quiz.dart'; // Mengimpor file halaman_quiz.dart yang berisi widget HalamanQuiz

Run | Debug
void main() {
  runApp(const MyApp()); // Menjalankan aplikasi Flutter dengan widget MyApp sebagai root widget
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}); // Deklarasi constructor MyApp

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Quiz App'), // Judul aplikasi di app bar
        ),
        body: HalamanQuiz(), // Menampilkan HalamanQuiz sebagai isi body aplikasi
      ),
    );
  }
}
```

Tentu, mari kita bahas kode di atas secara lengkap dan rinci:

1. `import 'package:flutter/material.dart';`
`import 'halaman_quiz.dart'; // Mengimpor file halaman_quiz.dart yang berisi widget HalamanQuiz`
Ini adalah bagian import, yang digunakan untuk mengimpor paket “flutter/material.dart” untuk mengakses widget Flutter dan mengimpor file “halaman_quiz.dart” yang berisi widget “HalamanQuiz” yang akan digunakan dalam aplikasi.

2. `void main() {`
`runApp(const MyApp()); // Menjalankan aplikasi Flutter dengan widget MyApp sebagai root widget`
Ini adalah fungsi “main”, yang merupakan titik masuk utama untuk aplikasi Flutter. Di dalamnya, “runApp” dipanggil dengan parameter “MyApp()” untuk memulai aplikasi dengan “MyApp” sebagai root widget.

3. `class MyApp extends StatelessWidget {`
`const MyApp({Key? key}); // Deklarasi constructor MyApp`
Ini adalah kelas “MyApp” yang merupakan turunan dari “StatelessWidget”. Kelas ini digunakan untuk membuat aplikasi Flutter dengan menggunakan widget stateless (tanpa keadaan).

4.
```
@override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Quiz App'), // Judul aplikasi di app bar
        ),
        body: HalamanQuiz(), // Menampilkan HalamanQuiz sebagai isi body aplikasi
      ),
    );
  }
```

Coding diatas adalah metode “build” dari kelas “MyApp”, yang akan membuat dan mengembalikan tampilan (UI) aplikasi. Di dalamnya, kita menggunakan “MaterialApp” sebagai root widget untuk menyediakan struktur aplikasi berbasis Material Design.

- “MaterialApp” memiliki properti `home` yang menunjukkan widget utama yang akan ditampilkan saat aplikasi dijalankan. Di sini, kita menggunakan `Scaffold` sebagai widget utama dengan “AppBar” dan “body”.

- “Scaffold” digunakan untuk membuat kerangka aplikasi yang memiliki fitur-fitur umum seperti AppBar, Drawer, dan lainnya. Di dalam “Scaffold”, kita menggunakan “AppBar” dengan judul "Quiz App" sebagai judul aplikasi di bagian atas.

- Properti “body” dari “Scaffold” menunjukkan widget yang akan ditampilkan sebagai konten utama aplikasi. Di sini, kita menggunakan widget “HalamanQuiz()” yang diimpor dari file “halaman_quiz.dart”. “HalamanQuiz()” merupakan widget yang berisi logika dan tampilan untuk menjalankan kuis.

Dengan demikian, kode di atas digunakan untuk membuat dan menjalankan aplikasi Flutter sederhana yang memiliki tampilan berbasis Material Design dengan judul "Quiz App" di AppBar dan konten kuis di dalam widget “HalamanQuiz”.

Source code pengacakan.dart

```dart
import 'dart:math'; // Mengimpor pustaka math untuk menggunakan class Random
import 'question.dart'; // Mengimpor file question.dart yang berisi definisi kelas Pertanyaan

// List bankSoal berisi daftar objek Pertanyaan yang akan digunakan dalam quiz
List<Pertanyaan> bankSoal = [
  Pertanyaan(
    'Berapa jumlah prodi di fakultas vokasi?', // Pertanyaan 1
    ['A. 10', 'B. 20', 'C. 30', 'D. 40'], // Opsi jawaban
    'A. 10', // Kunci jawaban
    'images/vokasi.jpg', // Lokasi gambar terkait pertanyaan
  ),
  Pertanyaan(
    'Di kota mana Unesa berada?', // Pertanyaan 2
    ['A. Surabaya', 'B. Mojokerto', 'C. Malang', 'D. Sidoarjo'], // Opsi jawaban
    'A. Surabaya', // Kunci jawaban
    'images/kampus.png', // Lokasi gambar terkait pertanyaan
  ),
  Pertanyaan(
    'Manajemen Informatika berada di Fakultas', // Pertanyaan 3
    ['A. Teknik', 'B. Vokasi', 'C. Bahasa', 'D. Sidoarjo'], // Opsi jawaban
    'B. Vokasi', // Kunci jawaban
    ('images/himti.jpg'),
  ),
  Pertanyaan(
    'Berapa Jumlah Fakultas yang berada di Kampus Lidah Wetan?', // Pertanyaan 4
    ['A. Sembilan', 'B. Dua', 'C. Tiga', 'D. Empat'], // Opsi jawaban
    'C. Tiga', // Kunci jawaban
    ('images/futsal.jpg'),
  ),
  Pertanyaan(
    'Dimana Kampus Cabang Unesa?', // Pertanyaan 5
    ['A. Jombang', 'B. Singapura', 'C. Hongkong', 'D. Magetan'], // Opsi jawaban
    'D. Magetan', // Kunci jawaban
    ('images/magetan.jpeg'),
  ),
```

```
    Pertanyaan(
      'Berapa Lama 1 Jam Kuliah saat bulan puasa?', // Pertanyaan 6
      [
        'A. 50 menit',
        'B. 40 menit',
        'C. 30 menit',
        'D. 60 menit'
      ], // Opsi jawaban
      'B. 40 menit', // Kunci jawaban
      ('images/ramadhan.png'),
    ),
    Pertanyaan(
      'Berapa total sks D4 Manajemen Informatika?', // Pertanyaan 7
      ['A. 144', 'B. 120', 'C. 150', 'D. 18'], // Opsi jawaban
      'A. 144', // Kunci jawaban
      ('images/manajemeninformatika.png'),
    ),
    Pertanyaan(
      'Gedung mana yang sering digunakan praktik mengajar untuk mahasiswa Manajemen Informatika?', // Pertanyaan 8
      ['A. K2', 'B. K10', 'C. A10', 'D. K4'], // Opsi jawaban
      'A. K2', // Kunci jawaban
      ('images/gedung.png'),
    ),
    Pertanyaan(
      'Dimana lokasi Fakultas Ilmu Pendidikan?', // Pertanyaan 9
      [
        'A. Lidah Wetan',
        'B. Surabaya',
        'C. Ketintang',
        'D. Ngagel'
      ], // Opsi jawaban
      'C. Ketintang', // Kunci jawaban
      ('images/fip.jpg'),
    ),
```

```
    Pertanyaan(
      'Bapak Dosen I Gde Agung Sri Sidhimantra, S.Kom., M.Kom. di Manajemen Informatika mengampu mata kuliah apa?', // Pertanyaan 10
      [
        'A. Kewarganegaraan',
        'B. Basis Data',
        'C. Kewirausahaan',
        'D. PemWe'
      ], // Opsi jawaban
      'D. PemWeb', // Kunci jawaban
      ('images/dosen.jpg'),
    ),
  ];

  // Fungsi acakSoal digunakan untuk mengacak urutan pertanyaan dalam list bankSoal
  void acakSoal() {
    bankSoal.shuffle(
        Random()); // Menggunakan fungsi shuffle dari list untuk mengacak urutan elemen
  }

```

1. Import Pustaka dan File Eksternal:

```
import 'dart:math'; // Mengimpor pustaka math untuk menggunakan class Random
import 'question.dart'; // Mengimpor file question.dart yang berisi definisi kelas Pertanyaan
```

Pada bagian ini, pustaka dart:math diimpor untuk menggunakan class Random, yang akan digunakan untuk mengacak urutan pertanyaan dalam list. Kemudian, file eksternal question.dart diimpor untuk mengakses definisi kelas Pertanyaan.

2. Deklarasi List bankSoal:

```
List<Pertanyaan> bankSoal = [
  // Daftar objek Pertanyaan
];
```

List bankSoal dideklarasikan sebagai sebuah list yang berisi objek-objek Pertanyaan. Setiap objek Pertanyaan merepresentasikan satu pertanyaan dalam kuis. Setiap objek Pertanyaan memiliki teks pertanyaan, opsi jawaban, kunci jawaban, dan lokasi gambar terkait.

3. Fungsi acakSoal():

```
void acakSoal() {
  bankSoal.shuffle(Random()); // Menggunakan fungsi shuffle dari list untuk mengacak urutan elemen
}
```

Fungsi acakSoal() digunakan untuk mengacak urutan pertanyaan dalam list bankSoal. Ini dilakukan dengan menggunakan metode shuffle() dari list, dengan menggunakan objek Random() sebagai argumen. Ini akan mengacak urutan elemen dalam list bankSoal setiap kali fungsi ini dipanggil.

Dengan demikian, potongan kode di atas menggambarkan proses impor pustaka dan file eksternal, deklarasi list bankSoal yang berisi pertanyaan-pertanyaan untuk kuis, dan fungsi acakSoal() untuk mengacak urutan pertanyaan.

Source code question.dart

```dart
// Deklarasi kelas Pertanyaan untuk merepresentasikan pertanyaan dalam quiz
class Pertanyaan {
  String pertanyaan; // Variabel untuk menyimpan teks pertanyaan
  List<String>
      jawaban; // Variabel untuk menyimpan opsi jawaban dalam bentuk list
  String kunciJawaban; // Variabel untuk menyimpan kunci jawaban
  String gambar; // Variabel untuk menyimpan lokasi gambar terkait pertanyaan

  // Konstruktor dengan parameter untuk menginisialisasi nilai variabel saat objek Pertanyaan dibuat
  Pertanyaan(this.pertanyaan, this.jawaban, this.kunciJawaban, this.gambar);
}

```

Kelas `Pertanyaan` adalah sebuah blueprint untuk merepresentasikan pertanyaan dalam kuis. Berikut penjelasan singkat mengenai deklarasinya:

1. Variabel Anggota:
   - pertanyaan: Variabel bertipe String untuk menyimpan teks pertanyaan.
   - jawaban : Variabel bertipe List <String> untuk menyimpan opsi jawaban dalam bentuk list.
   - kunciJawaban`: Variabel bertipe String untuk menyimpan kunci jawaban.
   - gambar : Variabel bertipe String untuk menyimpan lokasi gambar terkait pertanyaan.

2. Konstruktor:
Pertanyaan(this.pertanyaan, this.jawaban, this.kunciJawaban, this.gambar);

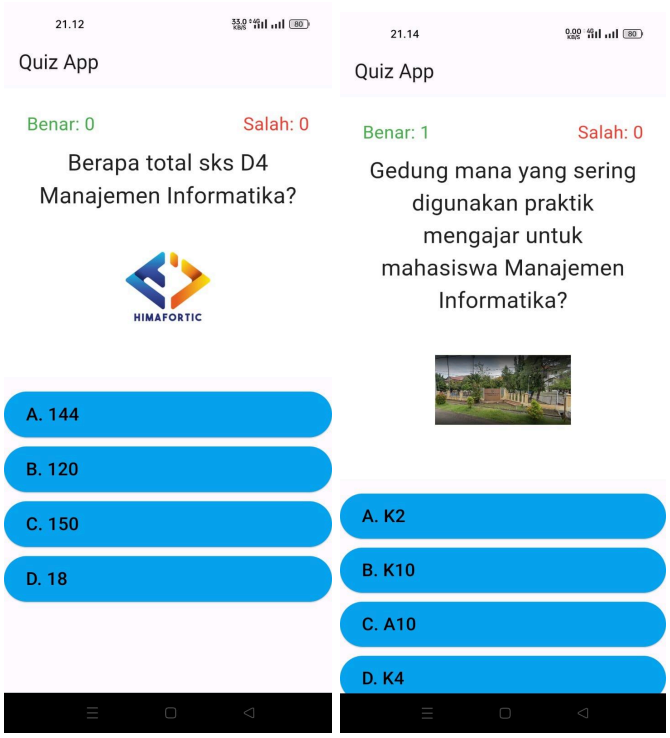
Konstruktor ini menerima empat parameter yang digunakan untuk menginisialisasi nilai variabel saat objek `Pertanyaan` dibuat.

Dengan deklarasi-deklarasi diatas, kelas `Pertanyaan` memungkinkan pembuatan objek pertanyaan dengan menyertakan teks pertanyaan, opsi jawaban, kunci jawaban, dan lokasi gambar terkait. Ini akan mempermudah pengelolaan dan penggunaan data pertanyaan dalam kuis.

Link Github:

https://github.com/AllanJuliSanjaya/utspemmobile2024

Berikut ini adalah hasil dari dokumentasi tampilan aplikasi quiz yang kami buat:



21:13 ···

Quiz App

Benar: 1 Salah: 1

Di kota mana Unesa berada?

A. Surabaya

B. Mojokerto

C. Malang

D. Sidoarjo

21:13 ···

Quiz App

Benar: 2 Salah: 1

Dimana lokasi Fakultas Ilmu Pendidikan?

A. Lidah Wetan

B. Surabaya

C. Ketintang

D. Ngagel

21:13 ···

Quiz App

Benar: 3 Salah: 2

Bapak Dosen I Gde Agung Sri Sidhimantra, S.Kom., M.Kom. di Manajemen Informatika mengampu mata kuliah apa?

A. Kewarganegaraan

B. Basis Data

C. Kewirausahaan

D. PemWeb

21:13 ···

Quiz App

Benar: 4 Salah: 2

Dimana Kampus Cabang Unesa?

A. Jombang

B. Singapura

C. Hongkong

D. Magetan

21:13

Quiz App

Benar: 6 Salah: 2

Manajemen Informatika berada di Fakultas

A. Teknik

B. Vokasi

C. Bahasa

D. Sidoarjo

21:15

Quiz App

Benar: 7 Salah: 2

Berapa Jumlah Fakultas yang berada di Kampus Lidah Wetan?

A. Sembilan

B. Dua

C. Tiga

D. Empat

23.57

Quiz App

Benar: 1 Salah: 0

Berapa Lama 1 Jam Kuliah saat bulan puasa?

A. 50 menit

B. 40 menit

C. 30 menit

D. 60 menit

21.17

Quiz App

Benar: 6 Salah: 2

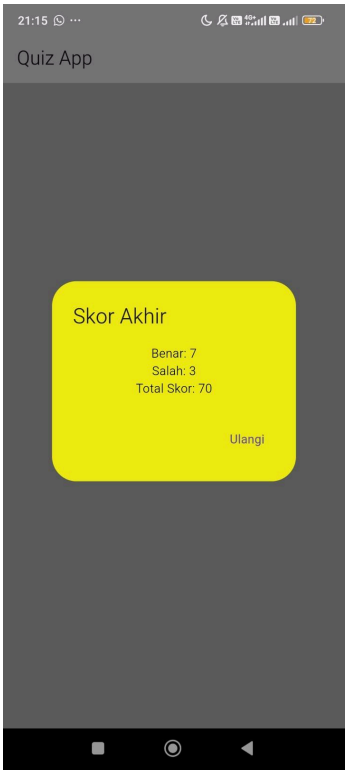
Berapa jumlah prodi di fakultas vokasi?

A. 10

B. 20

C. 30

D. 40


21:15
Quiz App
Skor Akhir
Benar: 7
Salah: 3
Total Skor: 70
Ulangi