TUGAS
WORKSHOP MOBILE APPLICATION FRAMEWORK

(TIFNJK140707)

SEMESTER IV

Object-Oriented Programing (OOP)

Dibuat oleh :

Natanel Putra Ananta (E41232268)

PROGRAM STUDI TEKNIK INFORMATIKA KAMPUS 3 NGANJUK

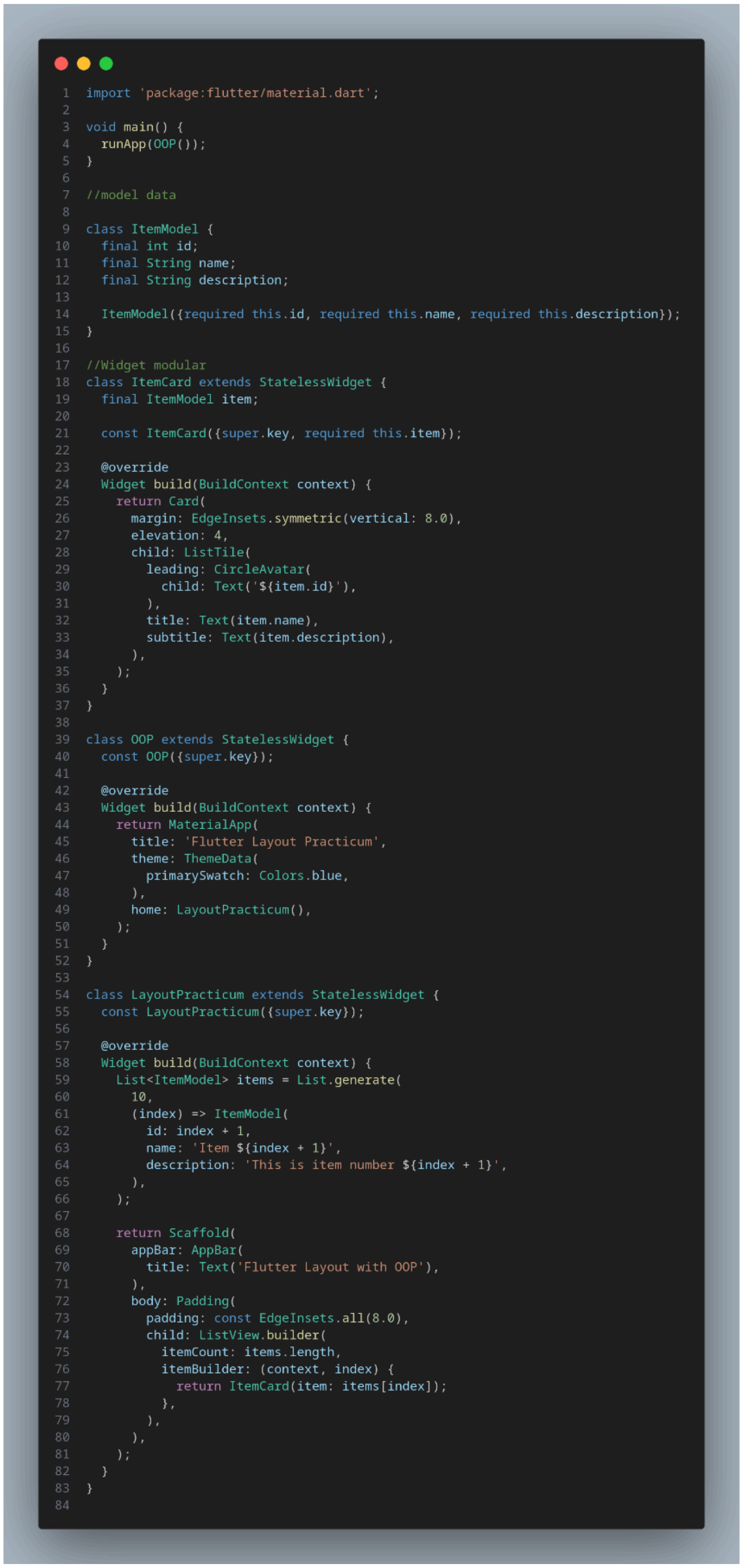
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI JEMBER

TAHUN 2025/2026

ObjectOriented Programing (OOP)

Kode Program :

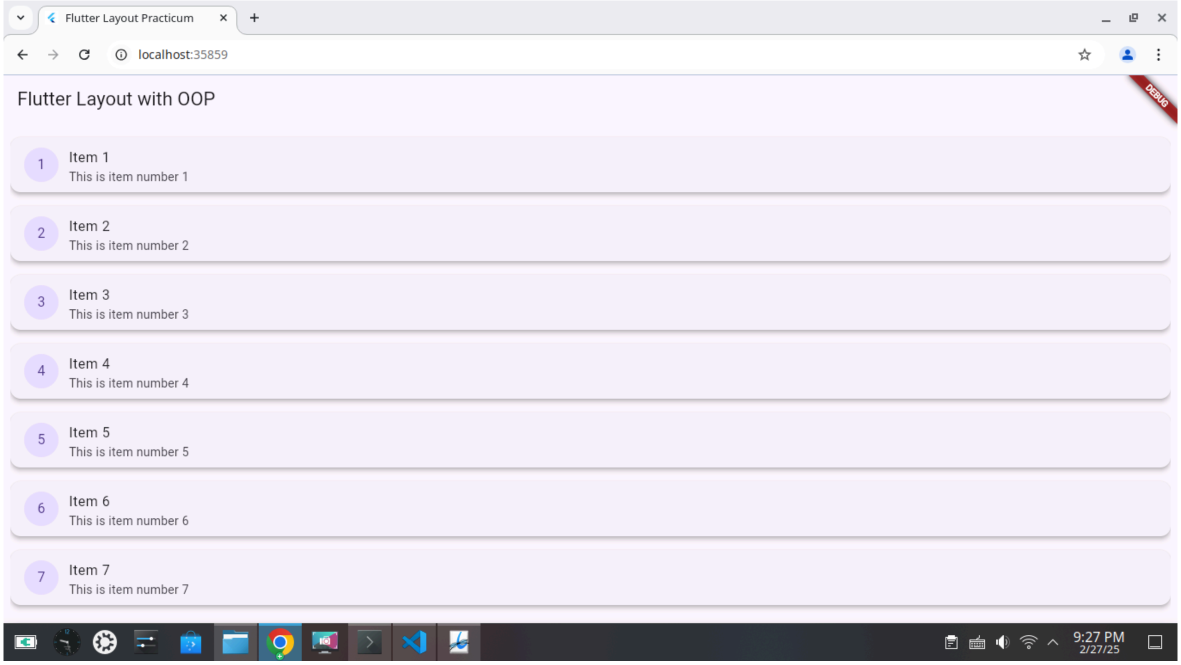


```dart
import 'package:flutter/material.dart';

void main() {
  runApp(OOP());
}

//model data

class ItemModel {
  final int id;
  final String name;
  final String description;

  ItemModel({required this.id, required this.name, required this.description});
}

//Widget modular
class ItemCard extends StatelessWidget {
  final ItemModel item;

  const ItemCard({super.key, required this.item});

  @override
  Widget build(BuildContext context) {
    return Card(
      margin: EdgeInsets.symmetric(vertical: 8.0),
      elevation: 4,
      child: ListTile(
        leading: CircleAvatar(
          child: Text('${item.id}'),
        ),
        title: Text(item.name),
        subtitle: Text(item.description),
      ),
    );
  }
}

class OOP extends StatelessWidget {
  const OOP({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Layout Practicum',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LayoutPracticum(),
    );
  }
}

class LayoutPracticum extends StatelessWidget {
  const LayoutPracticum({super.key});

  @override
  Widget build(BuildContext context) {
    List<ItemModel> items = List.generate(
      10,
      (index) => ItemModel(
        id: index + 1,
        name: 'Item ${index + 1}',
        description: 'This is item number ${index + 1}',
      ),
    );

    return Scaffold(
      appBar: AppBar(
        title: Text('Flutter Layout with OOP'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: ListView.builder(
          itemCount: items.length,
          itemBuilder: (context, index) {
            return ItemCard(item: items[index]);
          },
        ),
      ),
    );
  }
}

```

Penjelasan :

- Class OOP adalah class induk dari layout yang akan ditampilkan
- ItemModel untuk model item yang ditampilkan nomor setiap item,nama item, dan deskripsi
- ItemCard dipakai untuk menampilkan tiap item dalam bentuk kartu

Output :



Flutter Layout with OOP

1 Item 1
This is item number 1

2 Item 2
This is item number 2

3 Item 3
This is item number 3

4 Item 4
This is item number 4

5 Item 5
This is item number 5

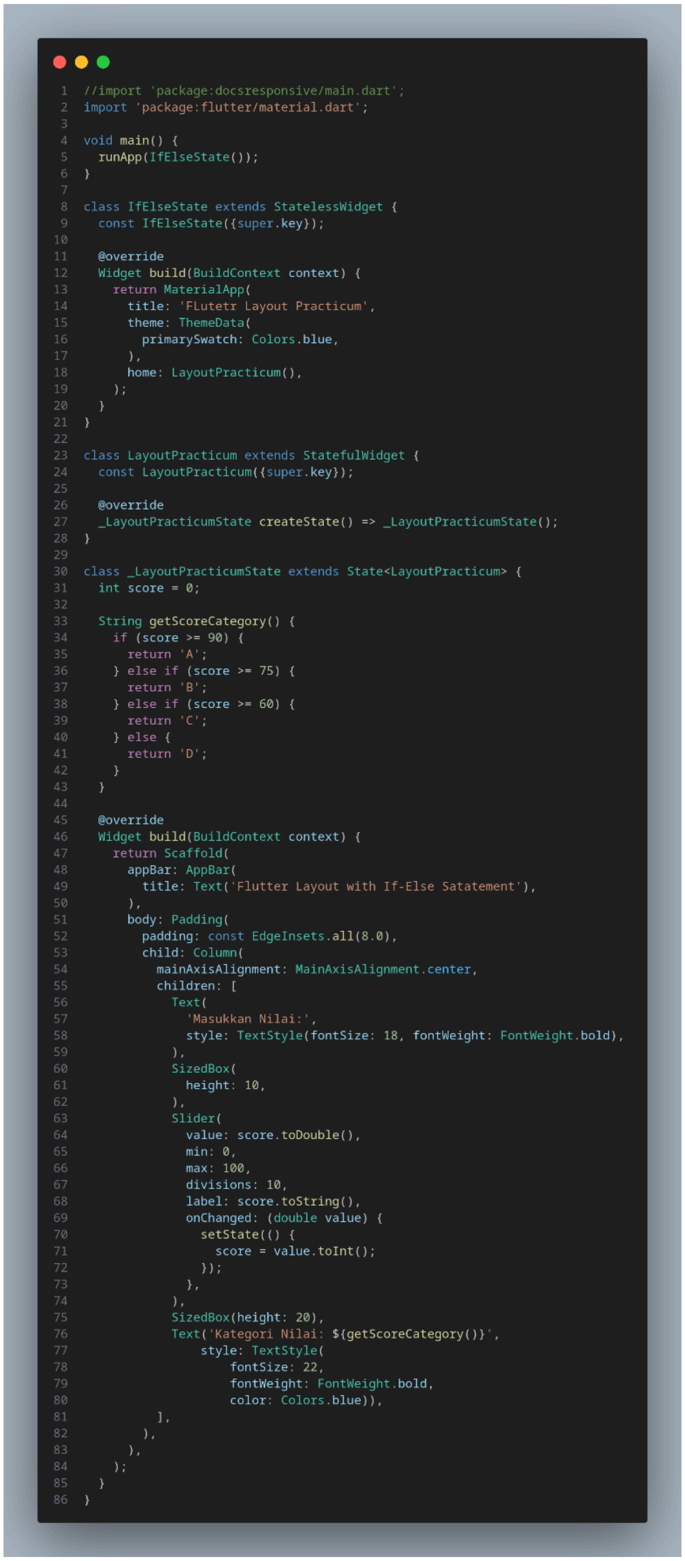
6 Item 6
This is item number 6

7 Item 7
This is item number 7

2. Implementasi Logika Pemograman pada Flutter

a. Percabangan if-else Statement

Kode Progam :

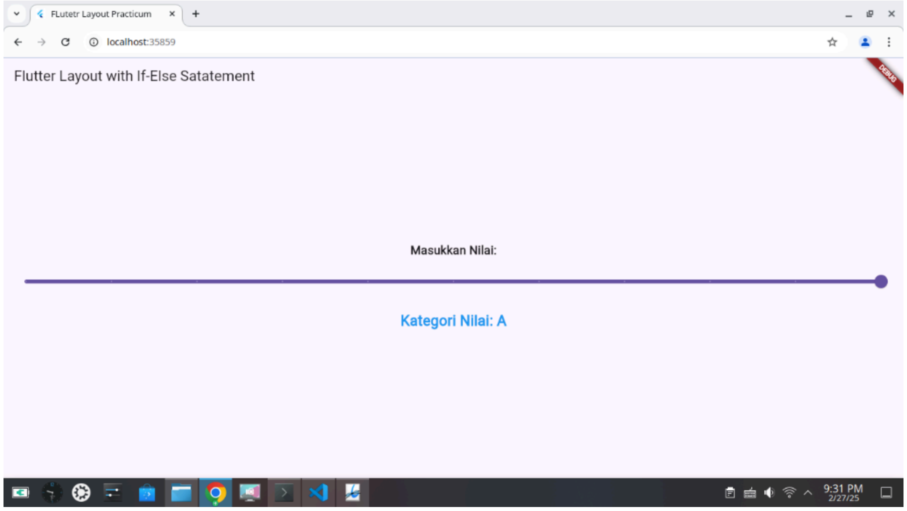


```dart
//import 'package:docsresponsive/main.dart';
import 'package:flutter/material.dart';

void main() {
  runApp(IfElseState());
}

class IfElseState extends StatelessWidget {
  const IfElseState({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'FLutetr Layout Practicum',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LayoutPracticum(),
    );
  }
}

class LayoutPracticum extends StatefulWidget {
  const LayoutPracticum({super.key});

  @override
  _LayoutPracticumState createState() => _LayoutPracticumState();
}

class _LayoutPracticumState extends State<LayoutPracticum> {
  int score = 0;

  String getScoreCategory() {
    if (score >= 90) {
      return 'A';
    } else if (score >= 75) {
      return 'B';
    } else if (score >= 60) {
      return 'C';
    } else {
      return 'D';
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Flutter Layout with If-Else Satatement'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              'Masukkan Nilai:',
              style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
            ),
            SizedBox(
              height: 10,
            ),
            Slider(
              value: score.toDouble(),
              min: 0,
              max: 100,
              divisions: 10,
              label: score.toString(),
              onChanged: (double value) {
                setState(() {
                  score = value.toInt();
                });
              },
            ),
            SizedBox(height: 20),
            Text('Kategori Nilai: ${getScoreCategory()}',
                style: TextStyle(
                    fontSize: 22,
                    fontWeight: FontWeight.bold,
                    color: Colors.blue)),
          ],
        ),
      ),
    );
  }
}
```

Penjelasan :

- Class induk yaitu IfElseState
- Menggunakan Statefull Widget karena nilai bisa diubah jika slide digeser
- If else untuk kategori nilai
- Slider untuk input nilai

Output :



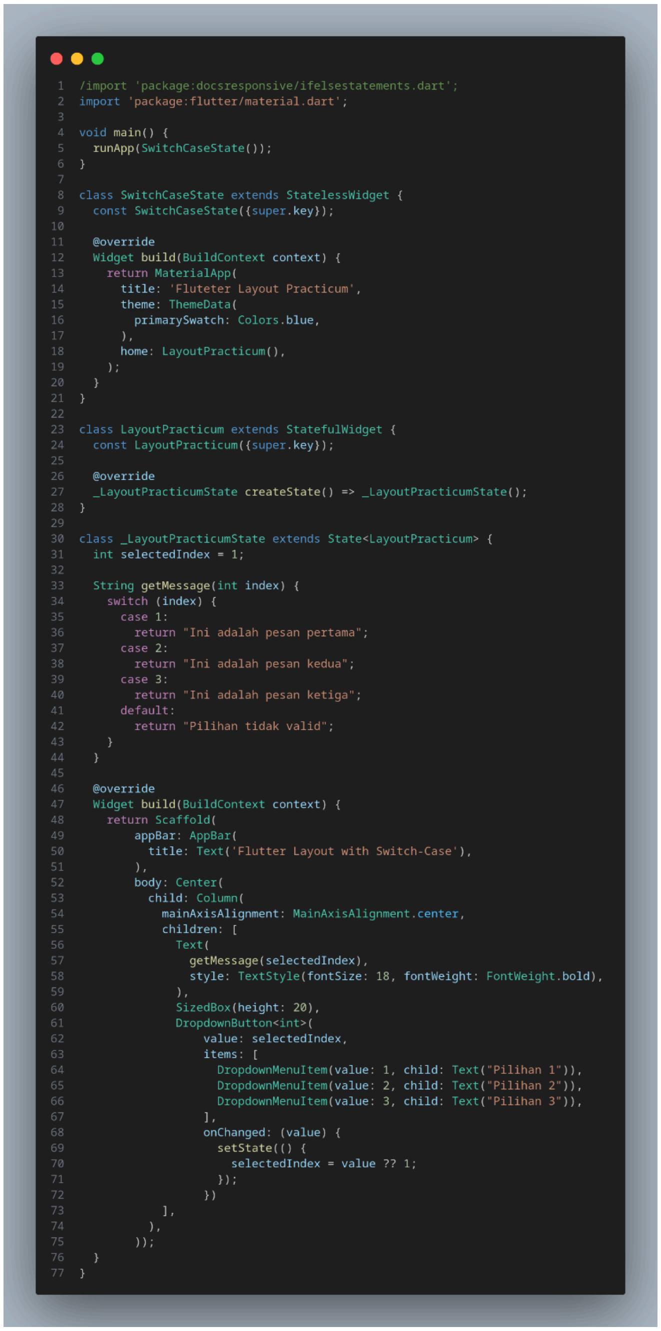
Flutter Layout with If-Else Satatement

Masukkan Nilai:

Kategori Nilai: A

b. Percabangan switch-case statement

Kode Program :

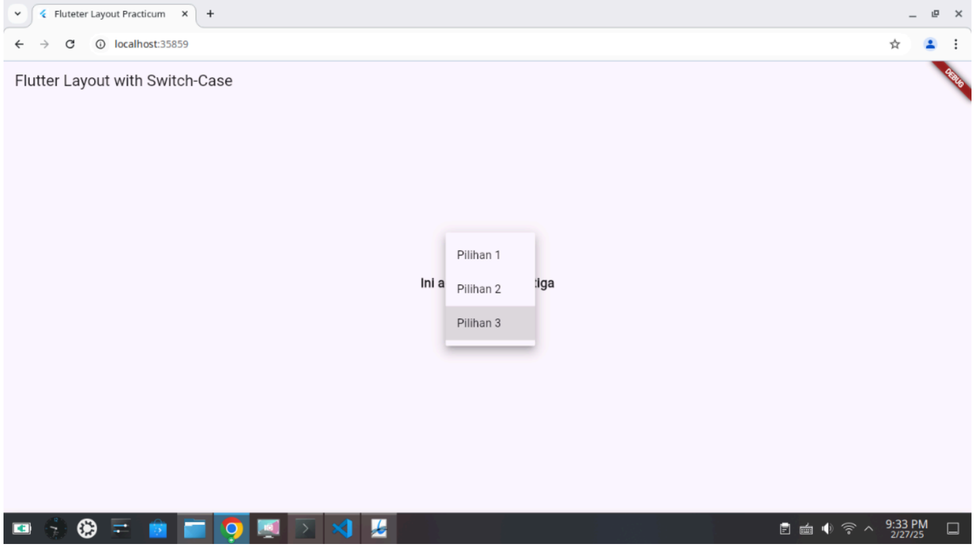


```dart
/import 'package:docsresponsive/ifelsestatements.dart';
import 'package:flutter/material.dart';

void main() {
  runApp(SwitchCaseState());
}

class SwitchCaseState extends StatelessWidget {
  const SwitchCaseState({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Fluteter Layout Practicum',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LayoutPracticum(),
    );
  }
}

class LayoutPracticum extends StatefulWidget {
  const LayoutPracticum({super.key});

  @override
  _LayoutPracticumState createState() => _LayoutPracticumState();
}

class _LayoutPracticumState extends State<LayoutPracticum> {
  int selectedIndex = 1;

  String getMessage(int index) {
    switch (index) {
      case 1:
        return "Ini adalah pesan pertama";
      case 2:
        return "Ini adalah pesan kedua";
      case 3:
        return "Ini adalah pesan ketiga";
      default:
        return "Pilihan tidak valid";
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: Text('Flutter Layout with Switch-Case'),
        ),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(
                getMessage(selectedIndex),
                style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
              ),
              SizedBox(height: 20),
              DropdownButton<int>(
                  value: selectedIndex,
                  items: [
                    DropdownMenuItem(value: 1, child: Text("Pilihan 1")),
                    DropdownMenuItem(value: 2, child: Text("Pilihan 2")),
                    DropdownMenuItem(value: 3, child: Text("Pilihan 3")),
                  ],
                  onChanged: (value) {
                    setState(() {
                      selectedIndex = value ?? 1;
                    });
                  })
            ],
          ),
        ));
  }
}
```

Penjelasan :

- Class induk yaitu SwitchCaseState
- Menggunakan StatefulWidget karena nilai pilihan bisa berubah
- SwitchCase untuk membuat pesan berdasarkan pilihan
- DropdownButton untuk memunculkan opsi dan memilih angka

Output :



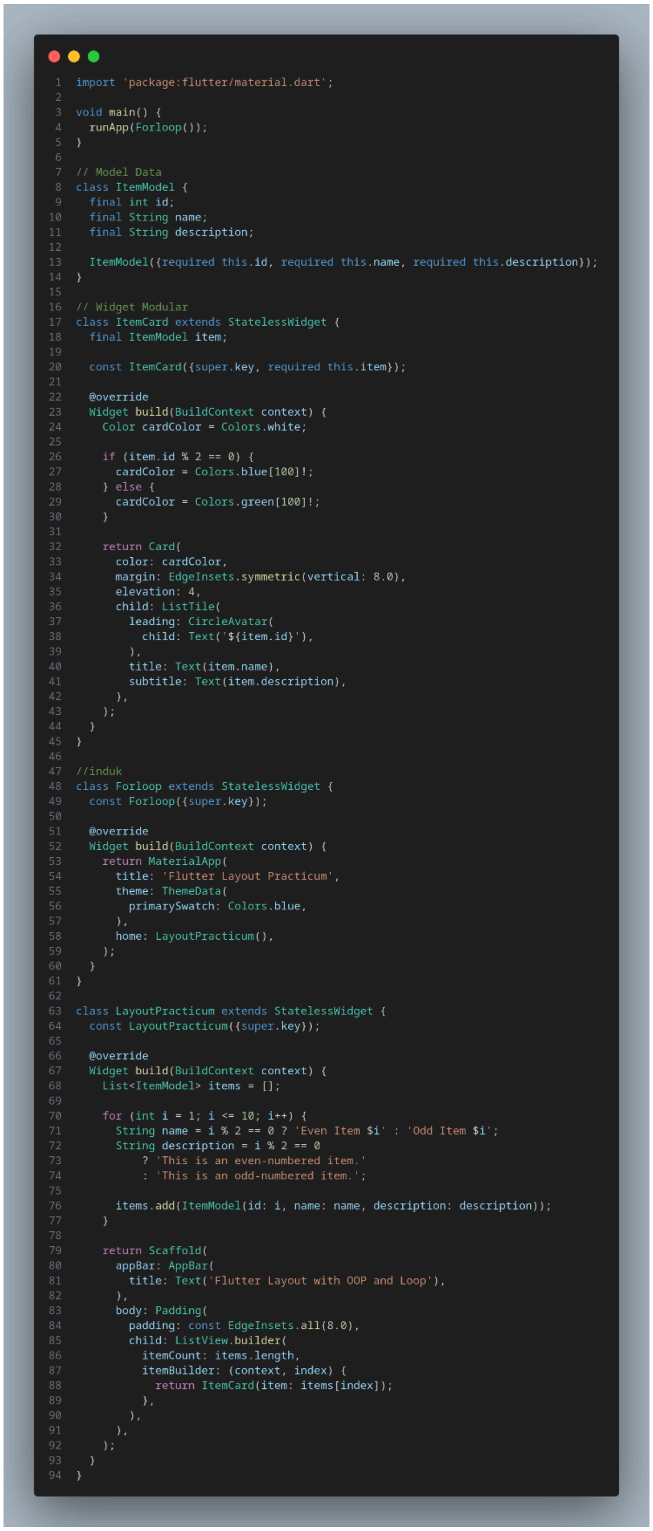
Flutter Layout with Switch-Case

Ini a

tiga

c. For loop

Kode Program :

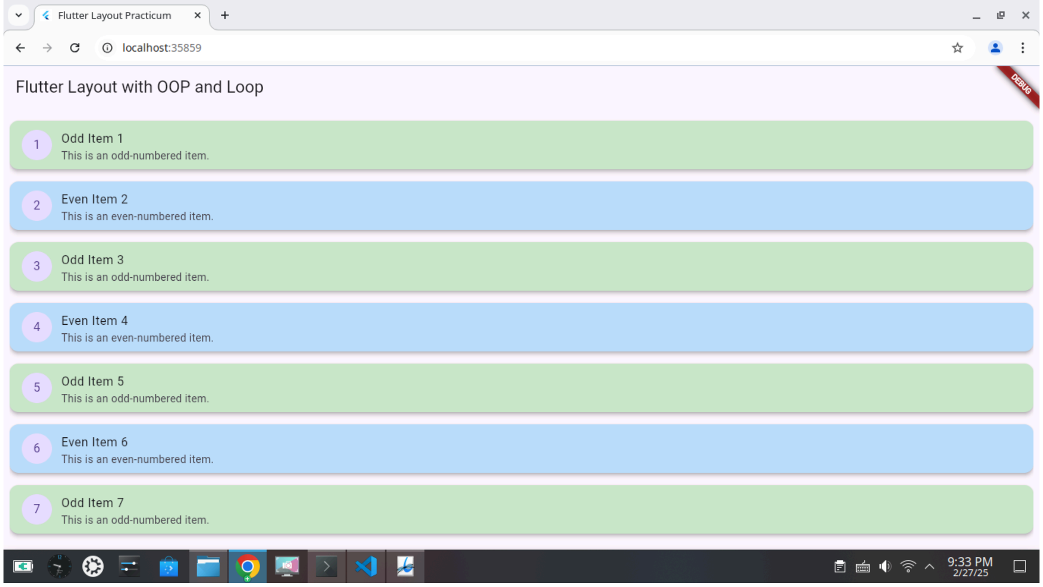


```dart
import 'package:flutter/material.dart';

void main() {
  runApp(Forloop());
}

// Model Data
class ItemModel {
  final int id;
  final String name;
  final String description;

  ItemModel({required this.id, required this.name, required this.description});
}

// Widget Modular
class ItemCard extends StatelessWidget {
  final ItemModel item;

  const ItemCard({super.key, required this.item});

  @override
  Widget build(BuildContext context) {
    Color cardColor = Colors.white;

    if (item.id % 2 == 0) {
      cardColor = Colors.blue[100]!;
    } else {
      cardColor = Colors.green[100]!;
    }

    return Card(
      color: cardColor,
      margin: EdgeInsets.symmetric(vertical: 8.0),
      elevation: 4,
      child: ListTile(
        leading: CircleAvatar(
          child: Text('${item.id}'),
        ),
        title: Text(item.name),
        subtitle: Text(item.description),
      ),
    );
  }
}

//induk
class Forloop extends StatelessWidget {
  const Forloop({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Layout Practicum',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LayoutPracticum(),
    );
  }
}

class LayoutPracticum extends StatelessWidget {
  const LayoutPracticum({super.key});

  @override
  Widget build(BuildContext context) {
    List<ItemModel> items = [];

    for (int i = 1; i <= 10; i++) {
      String name = i % 2 == 0 ? 'Even Item $i' : 'Odd Item $i';
      String description = i % 2 == 0
          ? 'This is an even-numbered item.'
          : 'This is an odd-numbered item.';

      items.add(ItemModel(id: i, name: name, description: description));
    }

    return Scaffold(
      appBar: AppBar(
        title: Text('Flutter Layout with OOP and Loop'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: ListView.builder(
          itemCount: items.length,
          itemBuilder: (context, index) {
            return ItemCard(item: items[index]);
          },
        ),
      ),
    );
  }
}
```

Penjelasan :

- Class induk yaitu ForLoop yang dijalankan
- item model untuk id,nama dan deskripsi
- Item card mengatur warna angka ganjil dan genap

Output :



Flutter Layout with OOP and Loop

1 Odd Item 1
This is an odd-numbered item.

2 Even Item 2
This is an even-numbered item.

3 Odd Item 3
This is an odd-numbered item.

4 Even Item 4
This is an even-numbered item.

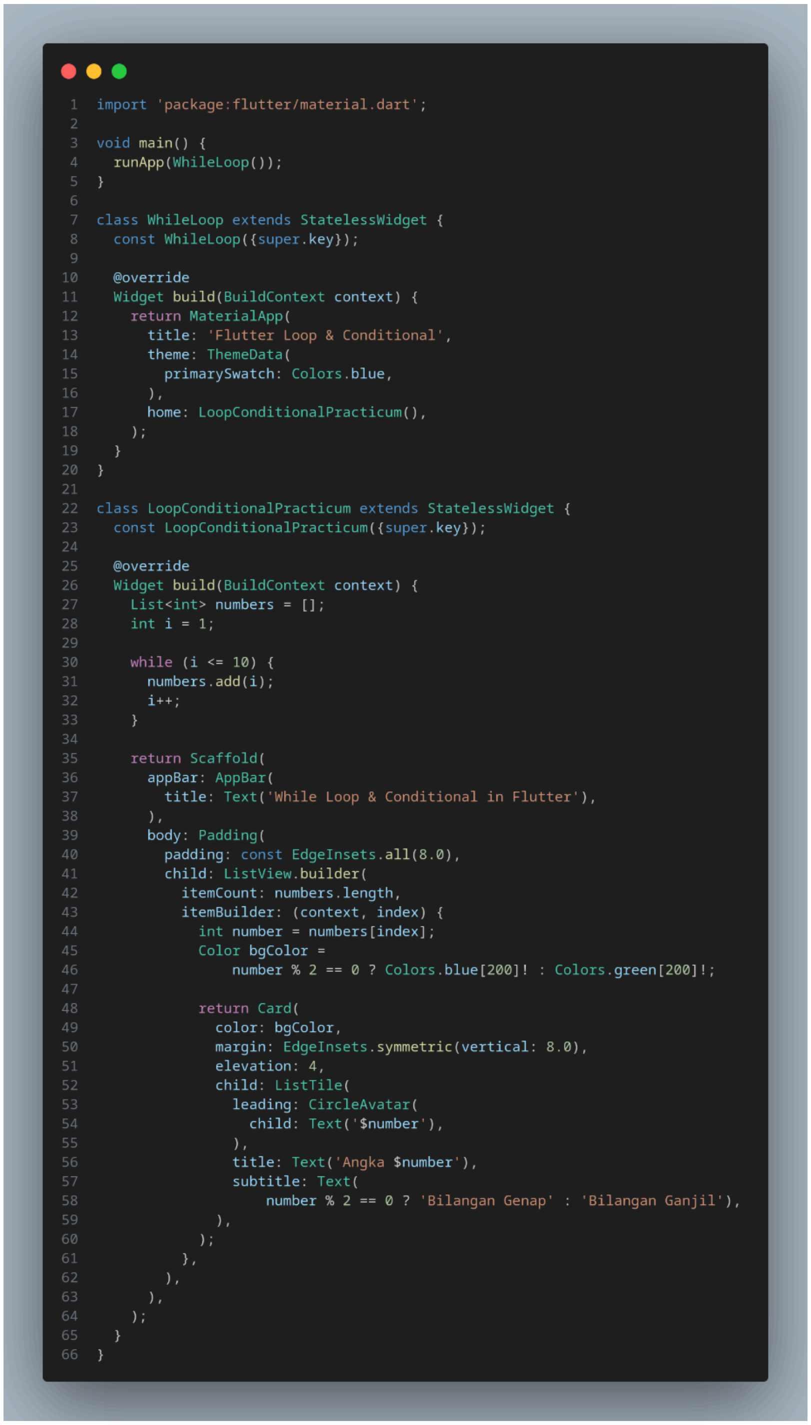
5 Odd Item 5
This is an odd-numbered item.

6 Even Item 6
This is an even-numbered item.

7 Odd Item 7
This is an odd-numbered item.

d. While loop

Kode Program :

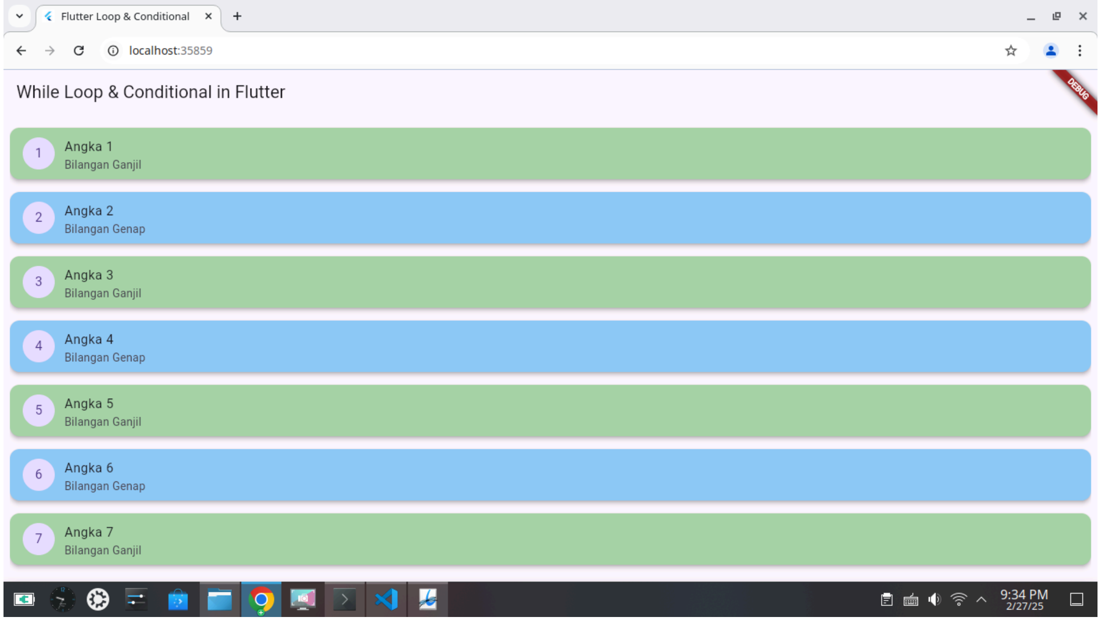


```dart
import 'package:flutter/material.dart';

void main() {
  runApp(WhileLoop());
}

class WhileLoop extends StatelessWidget {
  const WhileLoop({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Loop & Conditional',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LoopConditionalPracticum(),
    );
  }
}

class LoopConditionalPracticum extends StatelessWidget {
  const LoopConditionalPracticum({super.key});

  @override
  Widget build(BuildContext context) {
    List<int> numbers = [];
    int i = 1;

    while (i <= 10) {
      numbers.add(i);
      i++;
    }

    return Scaffold(
      appBar: AppBar(
        title: Text('While Loop & Conditional in Flutter'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: ListView.builder(
          itemCount: numbers.length,
          itemBuilder: (context, index) {
            int number = numbers[index];
            Color bgColor =
                number % 2 == 0 ? Colors.blue[200]! : Colors.green[200]!;

            return Card(
              color: bgColor,
              margin: EdgeInsets.symmetric(vertical: 8.0),
              elevation: 4,
              child: ListTile(
                leading: CircleAvatar(
                  child: Text('$number'),
                ),
                title: Text('Angka $number'),
                subtitle: Text(
                    number % 2 == 0 ? 'Bilangan Genap' : 'Bilangan Ganjil'),
              ),
            );
          },
        ),
      ),
    );
  }
}
```

Penjelasan :

- Class induk yaitu WhileLoop
- While untuk buat list angka
- ListView.builder untuk menampilkan angka

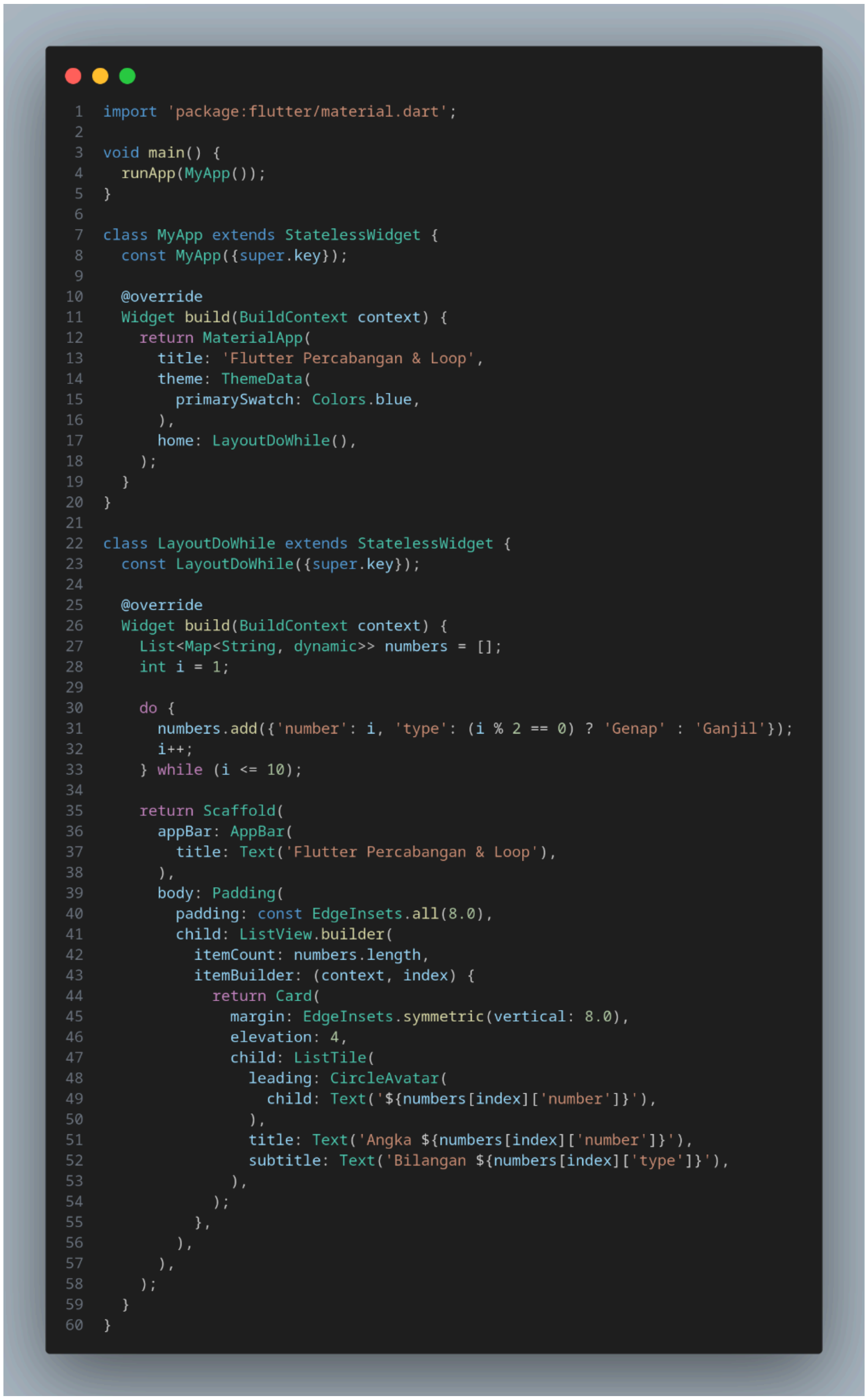
Output :



```
While Loop & Conditional in Flutter

1  Angka 1
   Bilangan Ganjil
2  Angka 2
   Bilangan Genap
3  Angka 3
   Bilangan Ganjil
4  Angka 4
   Bilangan Genap
5  Angka 5
   Bilangan Ganjil
6  Angka 6
   Bilangan Genap
7  Angka 7
   Bilangan Ganjil
```

e. **Do-while loop**

Kode Program :

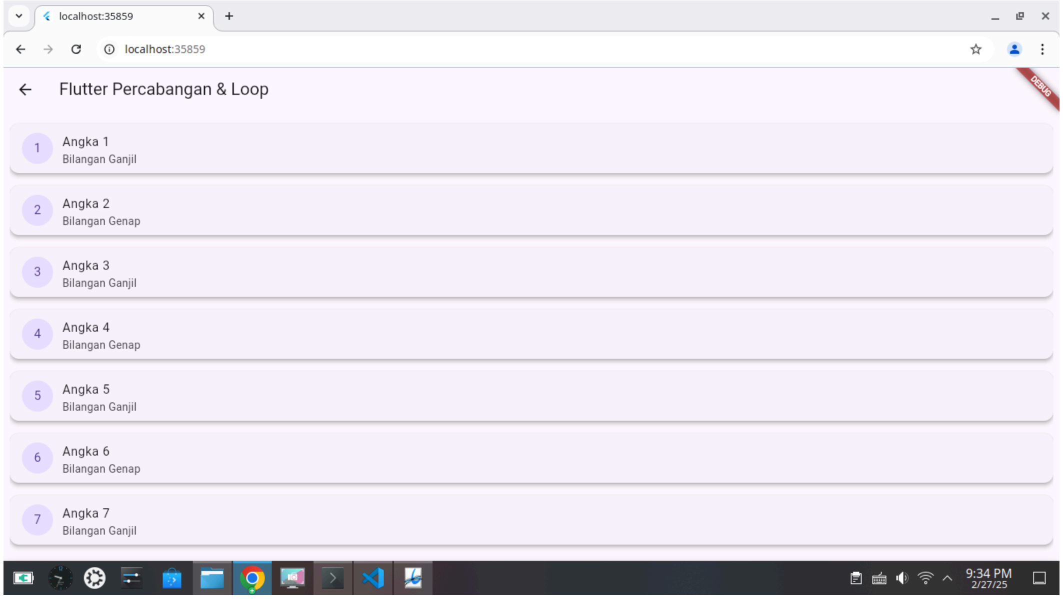


```dart
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Percabangan & Loop',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: LayoutDoWhile(),
    );
  }
}

class LayoutDoWhile extends StatelessWidget {
  const LayoutDoWhile({super.key});

  @override
  Widget build(BuildContext context) {
    List<Map<String, dynamic>> numbers = [];
    int i = 1;

    do {
      numbers.add({'number': i, 'type': (i % 2 == 0) ? 'Genap' : 'Ganjil'});
      i++;
    } while (i <= 10);

    return Scaffold(
      appBar: AppBar(
        title: Text('Flutter Percabangan & Loop'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(8.0),
        child: ListView.builder(
          itemCount: numbers.length,
          itemBuilder: (context, index) {
            return Card(
              margin: EdgeInsets.symmetric(vertical: 8.0),
              elevation: 4,
              child: ListTile(
                leading: CircleAvatar(
                  child: Text('${numbers[index]['number']}'),
                ),
                title: Text('Angka ${numbers[index]['number']}'),
                subtitle: Text('Bilangan ${numbers[index]['type']}'),
              ),
            );
          },
        ),
      ),
    );
  }
}
```

Penjelasan :

- Untuk class induk yaitu LayoutDoWhile yang akan dijalankan
- Memakai do while untuk mengisi list 1-10 dan setiap angka akan dicek ganjil atau genap
- Card Listile membuat tampilan lebih rapi karena menampilkan angka dalam lingkaran teks dan bilangan

Output :



Flutter Percabangan & Loop

1 Angka 1
Bilangan Ganjil

2 Angka 2
Bilangan Genap

3 Angka 3
Bilangan Ganjil

4 Angka 4
Bilangan Genap

5 Angka 5
Bilangan Ganjil

6 Angka 6
Bilangan Genap

7 Angka 7
Bilangan Ganjil