



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Ketentuan;

Simpanlah file lembar jawaban ini dengan format; **Kelas_Nama Lengkap**

Contoh; **TI 3C_Rifki Fakhruhin**


Upload tugas jobsheet ini dengan batas maksimum sesuai jadwal perkuliahan masing masing kelas

Upload file tugas jobsheet di website **lms.polinema**

Nama	:	Rifki Fakhruhin
Nim	:	2241720218
Kelas	:	3C

Tulislah Jawaban Pada Kolom Yang tersedia di bawah ini;

LEMBAR JAWABAN JOBSHEET-12

Soal No	Jawaban
	Praktikum 1 Langkah 1: Buat Project Baru 
	Langkah 2: Buka file main.dart <pre>import 'package:flutter/material.dart'; void main() { runApp(const MyApp()); } class MyApp extends StatelessWidget { const MyApp({super.key}); @override Widget build(BuildContext context) { return MaterialApp(title: 'Stream', theme: ThemeData(</pre>



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

```
        primarySwatch: Colors.deepPurple,  
    ),  
    home: const StreamHomePage(),  
  );  
}  
}  
  
class StreamHomePage extends StatefulWidget {  
  const StreamHomePage({super.key});  
  
  @override  
  State<StreamHomePage> createState() => _StreamHomePageState();  
}  
  
class _StreamHomePageState extends State<StreamHomePage> {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Stream Home Page'),  
      ),  
    );  
  }  
}
```

Soal 1

- Tambahkan nama panggilan Anda pada title app sebagai identitas hasil pekerjaan Anda.

```
class _StreamHomePageState extends State<StreamHomePage> {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Stream Rifki'),  
      ), // AppBar  
    ); // Scaffold  
  }  
}
```

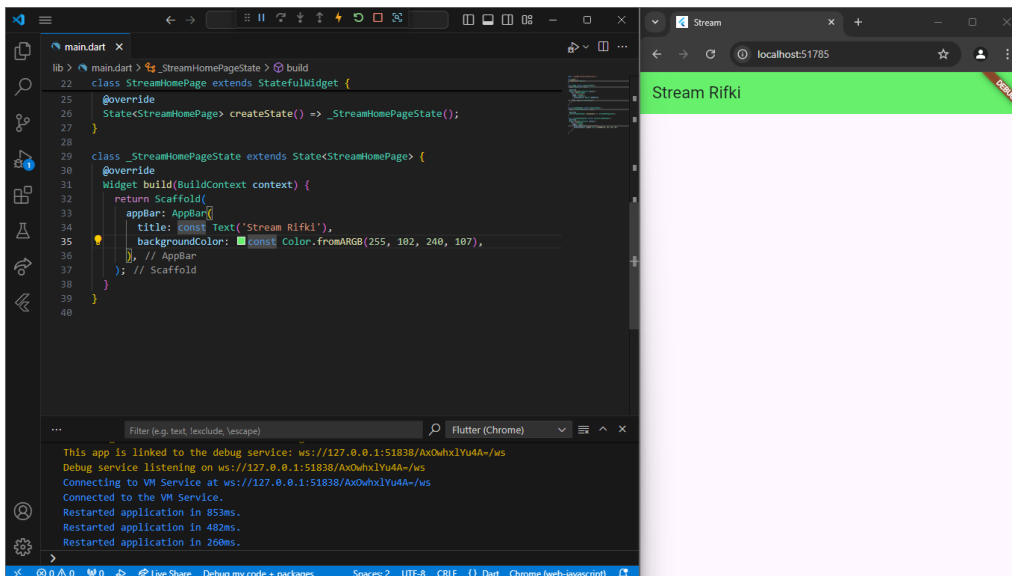
- Gantilah warna tema aplikasi sesuai kesukaan Anda.



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

```
class _StreamHomePageState extends State<StreamHomePage> {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Stream Rifki'),  
        backgroundColor: const Color.fromARGB(255, 102, 240, 107),  
      ), // AppBar  
    ); // Scaffold  
  }  
}
```

- Lakukan commit hasil jawaban Soal 1 dengan pesan "W13: Jawaban Soal 1"



Langkah 3: Buat file baru stream.dart

```
main.dart x stream.dart 1 x  
lib > stream.dart > ColorStream  
1 import 'package:flutter/material.dart';  
2  
3 class ColorStream {  
4  
5 }
```

Langkah 4: Tambah variabel colors



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

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

class ColorStream {
  final List<Color> colors = [
    Colors.blueGrey,
    Colors.amber,
    Colors.deepPurple,
    Colors.lightBlue,
    Colors.teal,
  ];
}
```

Soal 2

- Tambahkan 5 warna lainnya sesuai keinginan Anda pada variabel colors tersebut.

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

class ColorStream {
  final List<Color> colors = [
    Colors.blueGrey,
    Colors.amber,
    Colors.deepPurple,
    Colors.lightBlue,
    Colors.teal,
    //soal 2
    Colors.greenAccent,
    Colors.grey,
    Colors.orangeAccent,
    Colors.pinkAccent,
    Colors.redAccent,
  ];
}
```

- Lakukan commit hasil jawaban Soal 2 dengan pesan "W13: Jawaban Soal 2"

Langkah 5: Tambah method getColors()



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

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

class ColorStream {
  final List<Color> colors = [
    Colors.blueGrey,
    Colors.amber,
    Colors.deepPurple,
    Colors.lightBlue,
    Colors.teal,

    //soal 2
    Colors.greenAccent,
    Colors.grey,
    Colors.orangeAccent,
    Colors.pinkAccent,
    Colors.redAccent,
  ];
  Stream<Color> getColors() async* {
  }
}
```

Langkah 6: Tambah perintah yield*

```
Stream<Color> getColors() async* {
  yield* Stream.periodic(const Duration(seconds: 1), (int t) {
    int index = t % colors.length;
    return colors[index];
  }); // Stream.periodic
}
```

Soal 3

- **Jelaskan fungsi keyword yield* pada kode tersebut!**
 - Fungsi keyword yield* pada kode tersebut digunakan untuk "mengalirkan" nilai dari sebuah stream atau iterable ke dalam generator yang sedang berjalan.
- **Apa maksud isi perintah kode tersebut?**
 - Kode ini menghasilkan stream yang mengeluarkan warna secara periodik setiap detik, dengan indeks warna yang berulang berdasarkan waktu detik (t).
- **Lakukan commit hasil jawaban Soal 3 dengan pesan "W13: Jawaban Soal 3"**

Langkah 7: Buka main.dart



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

```
main.dart 1 x stream.dart  
lib > main.dart > main  
1 import 'package:flutter/material.dart';  
2 import 'stream.dart';  
3
```

Langkah 8: Tambah variabel

```
class _StreamHomePageState extends State<StreamHomePage> {  
  Color bgColor = Colors.blueGrey;  
  late ColorStream colorStream;
```

Langkah 9: Tambah method changeColor()

```
void changeColor() async {  
  await for (var eventColor in colorStream.getColors()){  
    setState(() {  
      bgColor = eventColor;  
    });  
  }  
}
```

Langkah 10: Lakukan override initState()

```
@override  
void initState() {  
  super.initState();  
  colorStream = ColorStream();  
  changeColor();  
}
```

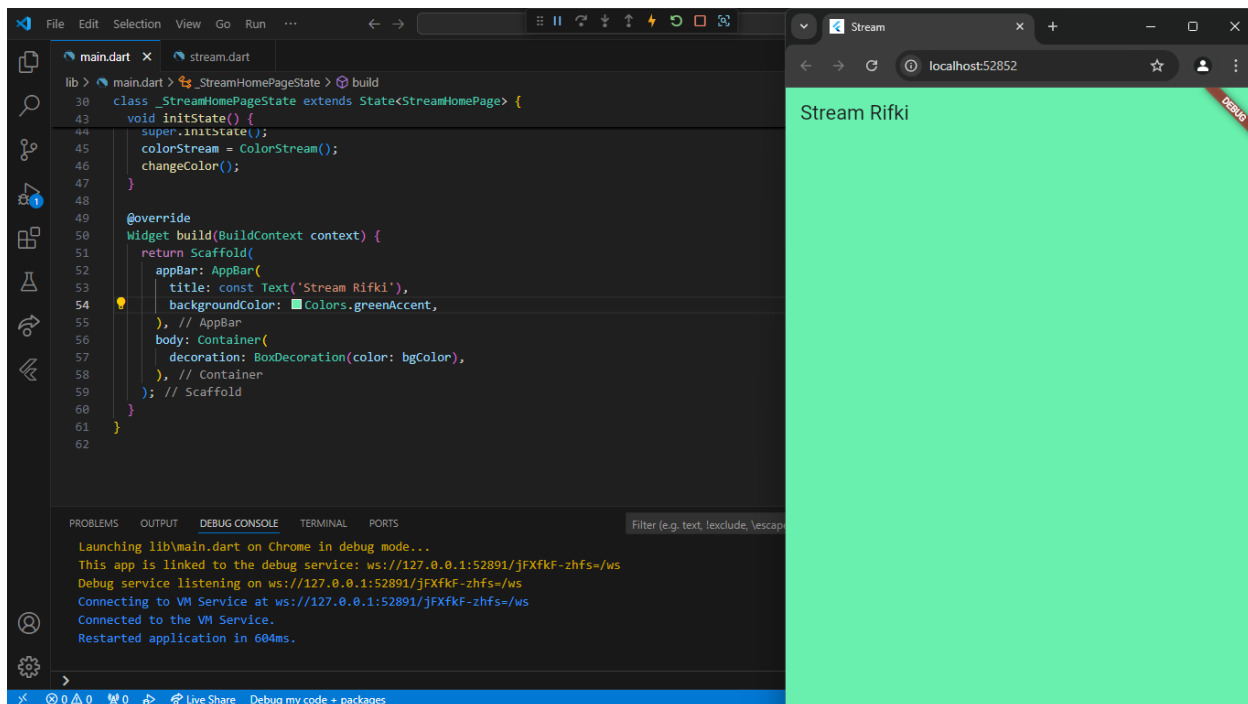
Langkah 11: Ubah isi Scaffold()



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Stream Rifki'),
    ), // AppBar
    body: Container(
      decoration: BoxDecoration(color: bgColor),
    ), // Container
  ); // Scaffold
}
```

Langkah 12: Run



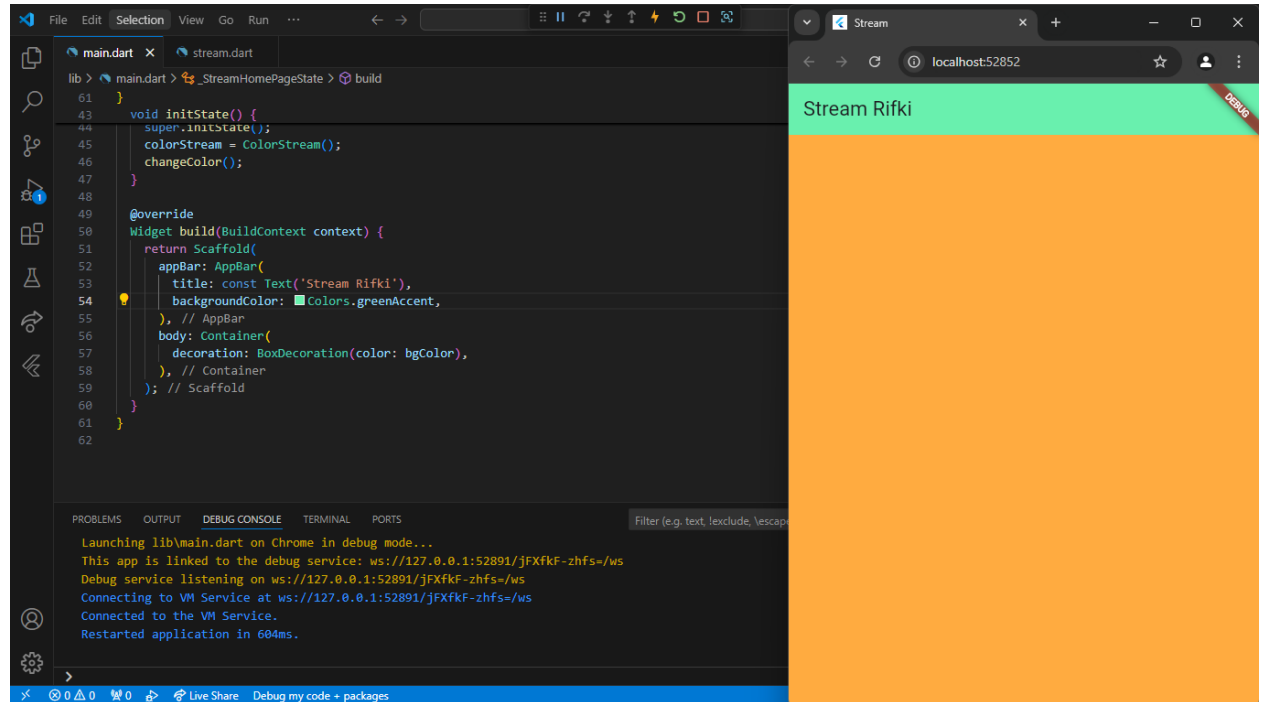
Soal 4

- Capture hasil praktikum Anda berupa GIF dan lampirkan di README.



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

- Background diatas bisa berganti warna dengan warna dari class stream.dart



- Lakukan commit hasil jawaban Soal 4 dengan pesan "W13: Jawaban Soal 4"

Langkah 13: Ganti isi method changeColor()

```
void changeColor() async {
  // await for (var eventColor in colorStream.getColors()) {
  //   setState(() {
  //     bgColor = eventColor;
  //   });
  // }
  colorStream.getColors().listen((eventColor){
    setState(() {
      bgColor = eventColor;
    });
  });
}
```




Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Soal 5

- **Jelaskan perbedaan menggunakan listen dan await for (langkah 9) !**
 - Kode 1 (await for) akan menunggu stream selesai sebelum melanjutkan eksekusi kode berikutnya. Hal ini lebih cocok untuk situasi di mana Anda ingin memproses setiap nilai stream secara berurutan dan menunggu semua data diterima.
 - Kode 2 (listen()) memungkinkan stream untuk tetap berjalan dan mendengarkan nilai baru yang dipancarkan tanpa menunggu stream selesai. Ini lebih cocok untuk aliran data yang berlangsung terus-menerus.
- **Lakukan commit hasil jawaban Soal 5 dengan pesan "W13: Jawaban Soal 5"**

Praktikum 2

Langkah 1: Buka file stream.dart

```
main.dart x stream.dart x
lib > stream.dart > ...
1 import 'package:flutter/material.dart';
2 import 'dart:async';
```

Langkah 2: Tambah class NumberStream

```
main.dart stream.dart x
lib > stream.dart > NumberStream
4 class ColorStream {
19   Stream<Color> getColors() async* {
21     int index = 0 % colors.length;
22     return colors[index];
23   }); // Stream.periodic
24 }
25
26 }
27 class NumberStream{
28
29 }
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Langkah 3: Tambah StreamController

```
class NumberStream{  
  final StreamController<int> controller = StreamController<int>();  
}
```

Langkah 4: Tambah method addNumberToSink

```
class NumberStream{  
  final StreamController<int> controller = StreamController<int>();  
  
  void addNumberToSink(int newNumber){  
    controller.sink.add(newNumber);  
  }  
}
```

Langkah 5: Tambah method close()

```
class NumberStream{  
  final StreamController<int> controller = StreamController<int>();  
  
  void addNumberToSink(int newNumber){  
    controller.sink.add(newNumber);  
  }  
  close() {  
    controller.close();  
  }  
}
```

Langkah 6: Buka main.dart

```
main.dart 2 x stream.dart  
lib > main.dart > ...  
1 import 'package:flutter/material.dart';  
2 import 'stream.dart';  
3 import 'dart:async';  
4 import 'dart:math';  
5
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Langkah 7: Tambah variabel

```
class _StreamHomePageState extends State<StreamHomePage> {  
  Color bgColor = Colors.blueGrey;  
  late ColorStream colorStream;  
  
  //praktikum 2 langkah  
  int lastNumber = 0;  
  late StreamController NumberStreamController;  
  late NumberStream numberStream;
```

Langkah 8: Edit initState()

```
@override  
void initState() {  
  numberStream = NumberStream();  
  NumberStreamController = numberStream.controller;  
  Stream stream = NumberStreamController.stream;  
  stream.listen((event){  
    setState(() {  
      lastNumber = event;  
    });  
  });  
  
  super.initState();  
  colorStream = ColorStream();  
  changeColor();  
}
```

Langkah 9: Edit dispose()

```
@override  
void dispose(){  
  NumberStreamController.close();  
  super.dispose();  
}
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Langkah 10: Tambah method addRandomNumber() void addRandomNumber() {

```
void addRandomNumber() {  
    Random random = Random();  
    int myNum = random.nextInt(10);  
    numberStream.addNumberToSink(myNum);  
}
```

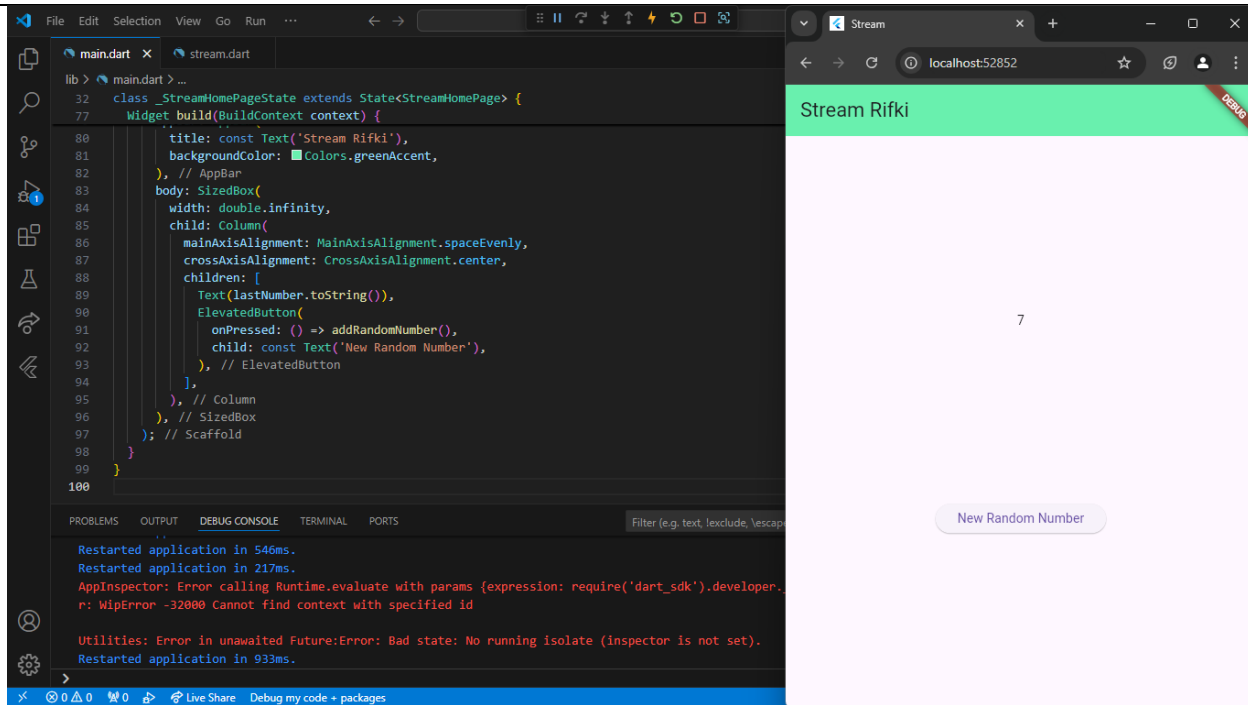
Langkah 11: Edit method build()

```
body: SizedBox(  
    width: double.infinity,  
    child: Column(  
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
        crossAxisAlignment: CrossAxisAlignment.center,  
        children: [  
            Text(lastNumber.toString()),  
            ElevatedButton(  
                onPressed: () => addRandomNumber(),  
                child: const Text('New Random Number'),  
            ), // ElevatedButton  
        ],  
    ), // Column  
), // SizedBox  
); // Scaffold  
}
```

Langkah 12: Run



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

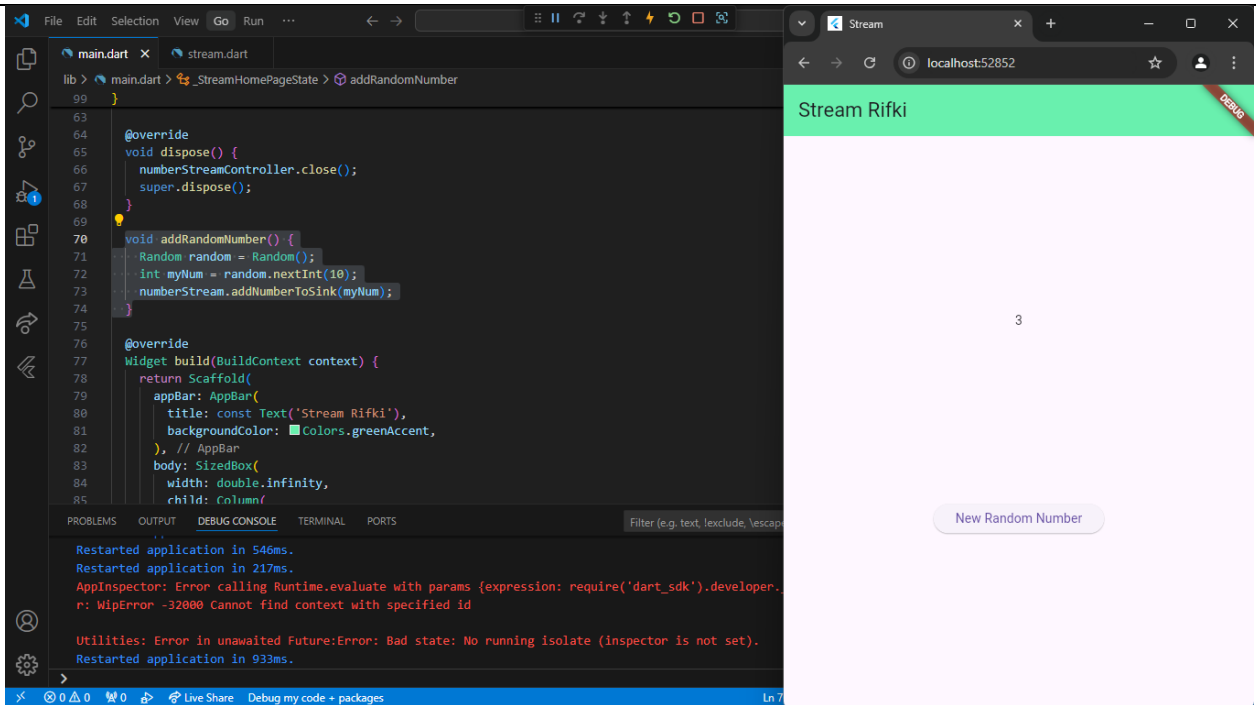


Soal 6

- **Jelaskan maksud kode langkah 8 dan 10 tersebut!**
 - Langkah 8 : berfungsi untuk mendengarkan dan memproses nilai stream
 - Langkah 10: menghasilkan dan mengirimkan nilai acak ke dalam stream.
- **Capture hasil praktikum Anda berupa GIF dan lampirkan di README.**



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile



- Lalu lakukan commit dengan pesan "W13: Jawaban Soal 6".

Langkah 13: Buka stream.dart

```
addError(){  
  controller.sink.addError('error');  
}
```

Langkah 14: Buka main.dart



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

```
@override
void initState() {
  super.initState();
  numberStream = NumberStream();
  numberStreamController = numberStream.controller;
  numberStreamController.stream.listen(
    (event) {
      setState(() {
        lastNumber = event;
      });
    },
    onError: (error) {
      setState(() {
        lastNumber = -1;
      });
    },
  );
};
```

Langkah 15: Edit method addRandomNumber()

```
void addRandomNumber() {
  // Random random = Random();
  // int myNum = random.nextInt(10);
  // numberStream.addNumberToSink(myNum);
  Random random = Random();
  numberStream.addError();
}
```

- **Jelaskan maksud kode langkah 13 sampai 15 tersebut!**
 - Langkah 13 : menambah error ke dalam stream secara eksplisit.
 - Langkah 14: mendengarkan stream dan mengatur nilai berdasarkan data atau error yang diterima.
 - Langkah 15: menghasilkan angka acak dan kemudian memicu error ke dalam stream menggunakan addError().



- Kembalikan kode seperti semula pada Langkah 15, comment `addError()` agar Anda dapat melanjutkan ke praktikum 3 berikutnya.

```
void addRandomNumber() {  
    Random random = Random();  
    int myNum = random.nextInt(10);  
    numberStream.addNumberToSink(myNum);  
    // numberStream.addError();  
}
```

- Lalu lakukan commit dengan pesan "W13: Jawaban Soal 7".

Praktikum 3

Langkah 1: Buka main.dart

```
// Praktikum 2 langkah 7  
int lastNumber = 0;  
late StreamController<int> numberStreamController;  
late NumberStream numberStream;  
//praktikum 3 langkah 1  
late StreamTransformer transformer;
```

Langkah 2: Tambahkan kode ini di initState

```
// Inisialisasi StreamTransformer  
transformer = StreamTransformer<int, int>.fromHandlers(  
    handleData: (value, sink) {  
        sink.add(value * 10);  
    },  
    handleError: (error, trace, sink) {  
        sink.add(-1);  
    },  
    handleDone: (sink) {  
        sink.close();  
    },  
);
```

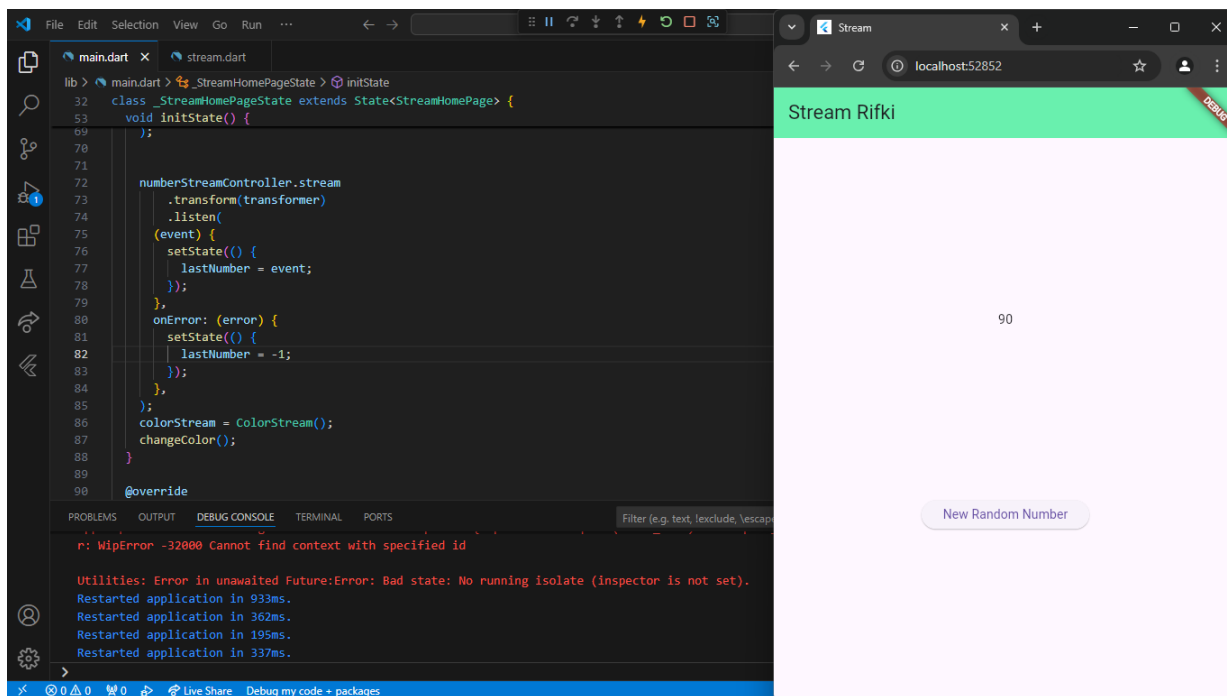



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Langkah 3: Tetap di initState

```
numberStreamController.stream
  .transform(transformer)
  .listen(
    (event) {
      setState(() {
        lastNumber = event;
      });
    },
    onError: (error) {
      setState(() {
        lastNumber = -1;
      });
    },
  );
colorStream = ColorStream();
changeColor();
}
```

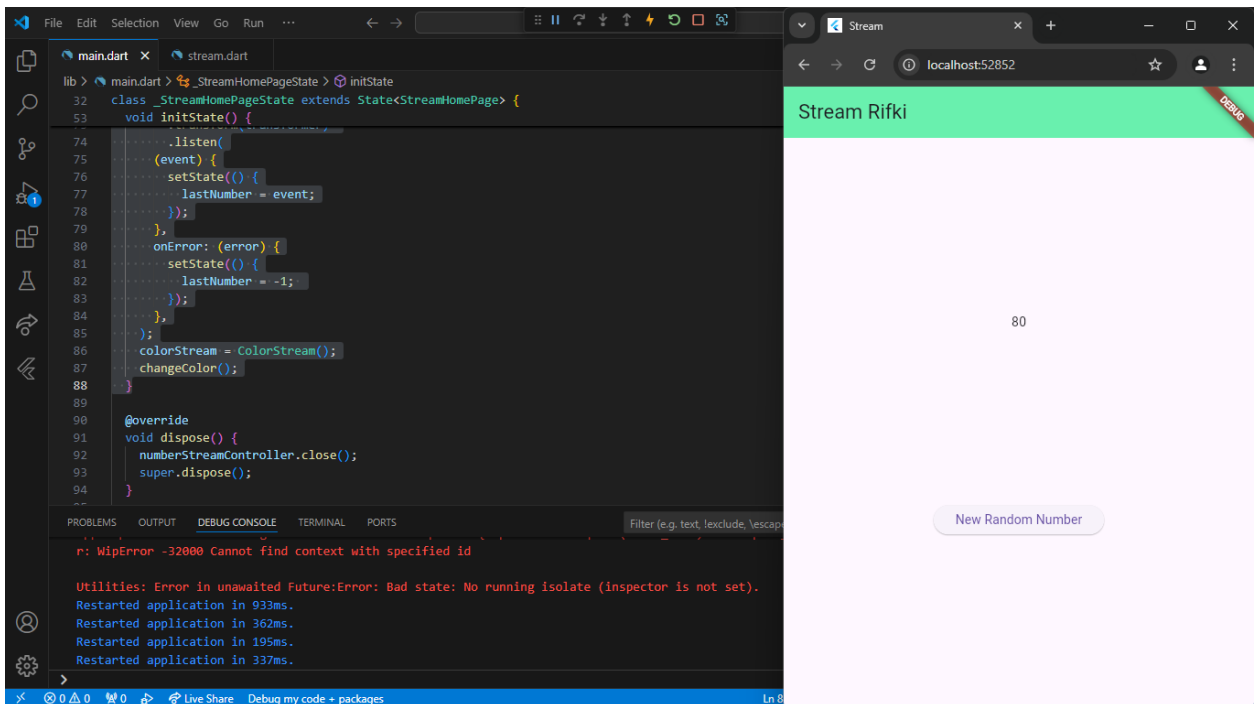
Langkah 4: Run





Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

- **Jelaskan maksud kode langkah 1-3 tersebut!**
 - Langkah 1 : mendeklarasikan variabel transformer yang akan digunakan untuk mentransformasi data dalam stream.
 - Langkah 2 : menginisialisasi transformer untuk mengubah nilai stream (mengalikan nilai dengan 10) dan menangani error serta menyelesaikan stream.
 - Langkah 3 : mengaplikasikan transformer pada stream yang diterima dari `numberStreamController`, mendengarkan perubahan data, mengupdate UI dengan data baru atau error, dan juga menginisialisasi stream warna untuk memperbarui latar belakang.
- **Capture hasil praktikum Anda berupa GIF dan lampirkan di README.**



- **Lalu lakukan commit dengan pesan "W13: Jawaban Soal 8".**



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Praktikum 4

Langkah 1: Tambah variabel

```
//praktikum 4 langkah 1  
late StreamSubscription subscription;
```

Langkah 2: Edit initState()

```
Stream<int> stream = numberStreamController.stream;  
subscription = stream.listen((event) {  
  setState(() {  
    lastNumber = event;  
  });  
});
```

Langkah 3: Tetap di initState()

```
Stream<int> stream = numberStreamController.stream;  
subscription = stream.listen((event) {  
  setState(() {  
    lastNumber = event;  
  });  
});  
  
subscription.onError((error){  
  setState(() {  
    lastNumber = -1;  
  });  
});
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Langkah 4: Tambah properti onDone()

```
subscription.onError((error){  
  setState(() {  
    lastNumber = -1;  
  });  
});  
subscription.onDone(){  
  print('onDone was called');  
});
```

Langkah 5: Tambah method baru

```
void stopStream(){  
  numberStreamController.close();  
}
```

Langkah 6: Pindah ke method dispose()

```
@override  
void dispose() {  
  numberStreamController.close();  
  super.dispose();  
  subscription.cancel();  
}
```

Langkah 7: Pindah ke method build()

```
ElevatedButton(  
  onPressed: () => stopStream(),  
  child: const Text('Stop Subscription'),  
) // ElevatedButton
```

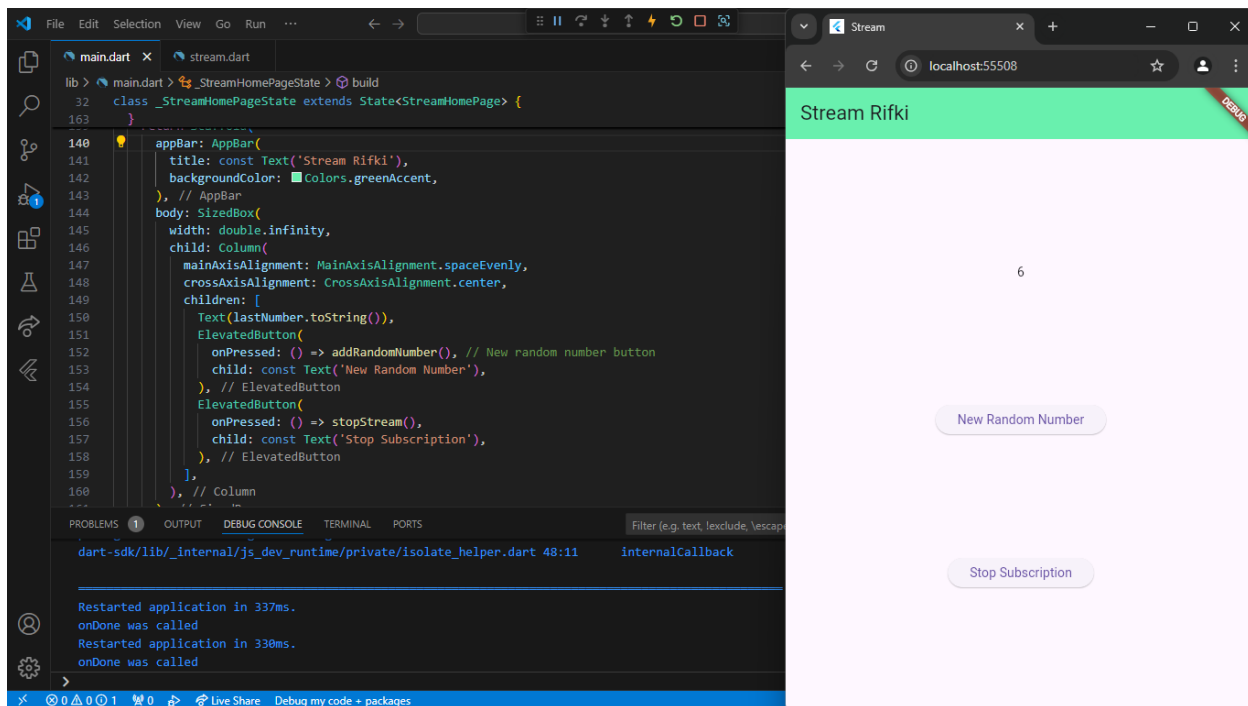


Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Langkah 8: Edit method addRandomNumber()

```
void addRandomNumber() {  
  Random random = Random();  
  int myNum = random.nextInt(10);  
  if (!numberStreamController.isClosed){  
    numberStream.addNumberToSink(myNum);  
  }else{  
    setState(() {  
      lastNumber = -1;  
    });  
  }  
}
```

Langkah 9: Run



Langkah 10: Tekan button 'Stop Subscription'

```
Restarted application in 337ms.  
onDone was called
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Soal 9

- **Jelaskan maksud kode langkah 2, 6 dan 8 tersebut!**
 - Langkah 2 : Mendaftar ke stream dan menangani event yang diterima;
 - Langkah 6: Membatalkan (menghentikan) subscription atau pendaftaran ke stream.
 - Langkah 8 : Menambahkan angka acak ke dalam stream dan memperbarui UI berdasarkan kondisi apakah stream terbuka atau tertutup.
- **Capture hasil praktikum Anda berupa GIF dan lampirkan di README.**
- **Lalu lakukan commit dengan pesan "W13: Jawaban Soal 9".**

Praktikum 5

Langkah 1: Buka file main.dart

```
//praktikum 5 langkah 1
late StreamSubscription subscription2;
String values = '';
```

Langkah 2: Edit initState()

```
// Subscribe to the stream
subscription = broadcastStream.listen((event) {
  setState(() {
    values += '$event - ';
  });
});

subscription2 = broadcastStream.listen((event){
  setState(() {
    values += '$event - ';
  });
});
```

Langkah 3: Run

Soal 10



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

- **Jelaskan mengapa error itu bisa terjadi ?**
 - Hal itu terjadi karena mencoba untuk mendengarkan (listen) sebuah stream yang sudah didengarkan sebelumnya.

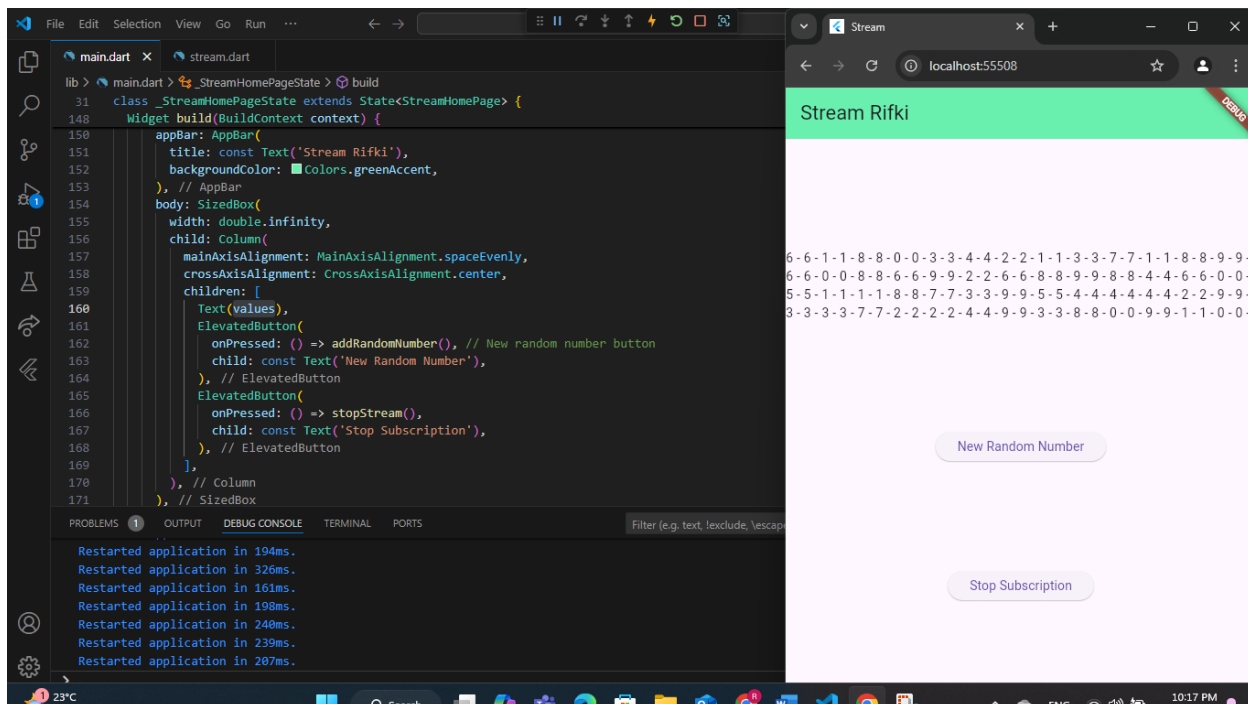
Langkah 4: Set broadcast stream

```
late StreamController<int> numberStreamController;
```

Langkah 5: Edit method build()

```
child: Column(  
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
  crossAxisAlignment: CrossAxisAlignment.center,  
  children: [  
    Text(values),
```

Langkah 6: Run





Soal 11

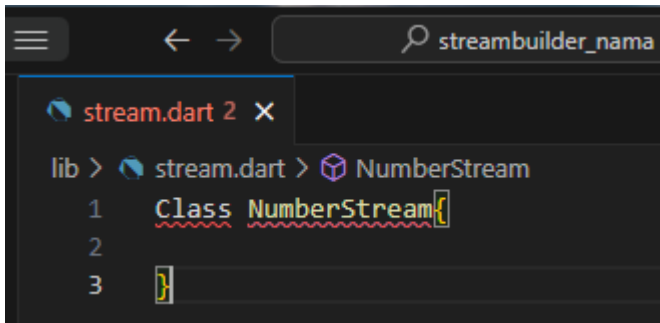
- **Jelaskan mengapa hal itu bisa terjadi ?**
 - Hal itu terjadi karena menggunakan Multiple StreamSubscription stream tersebut menjadi sebuah broadcast stream, yang artinya Anda dapat memiliki lebih dari satu pendengar (subscriber) yang mendengarkan stream yang sama secara bersamaan.
- **Capture hasil praktikum Anda berupa GIF dan lampirkan di README.**
- **Lalu lakukan commit dengan pesan "W13: Jawaban Soal 10,11".**

Praktikum 06

Langkah 1: Buat Project Baru



Langkah 2: Buat file baru stream.dart





Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Langkah 3: Tetap di file stream.dart

```
stream.dart X
lib > stream.dart > NumberStream > getNumbers
1  import 'dart:math';
2
3  class NumberStream {
4      Stream<int> getNumbers() async* {
5          yield* Stream.periodic(const Duration(seconds: 1), (int t) {
6              Random random = Random();
7              return random.nextInt(10);
8          }); // Stream.periodic
9      }
10 }
```

Langkah 4: Edit main.dart

```
import 'package:flutter/material.dart';
import 'stream.dart'; // Ensure the correct file path for this import
import 'dart:async';

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Stream',
      theme: ThemeData(
        primarySwatch: Colors.deepPurple,
      ),
      home: const StreamHomePage(),
    );
  }
}
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

```
class StreamHomePage extends StatefulWidget {  
  const StreamHomePage({super.key});  
  
  @override  
  State<StreamHomePage> createState() => _StreamHomePageState();  
}  
  
class _StreamHomePageState extends State<StreamHomePage> {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('stream'),  
      ),  
      body: Container(  
      ),  
    );  
  }  
}
```

Langkah 5: Tambah variabel

```
class _StreamHomePageState extends State<StreamHomePage> {  
  late Stream<int> NumberStream;  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('stream'),  
      ), // AppBar  
      body: Container(  
      ), // Container  
    ); // Scaffold  
  }  
}
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Langkah 6: Edit initState()

```
@override
void initState() {
  numberstream = NumberStream().getNumbers();
  super.initState();
}
```

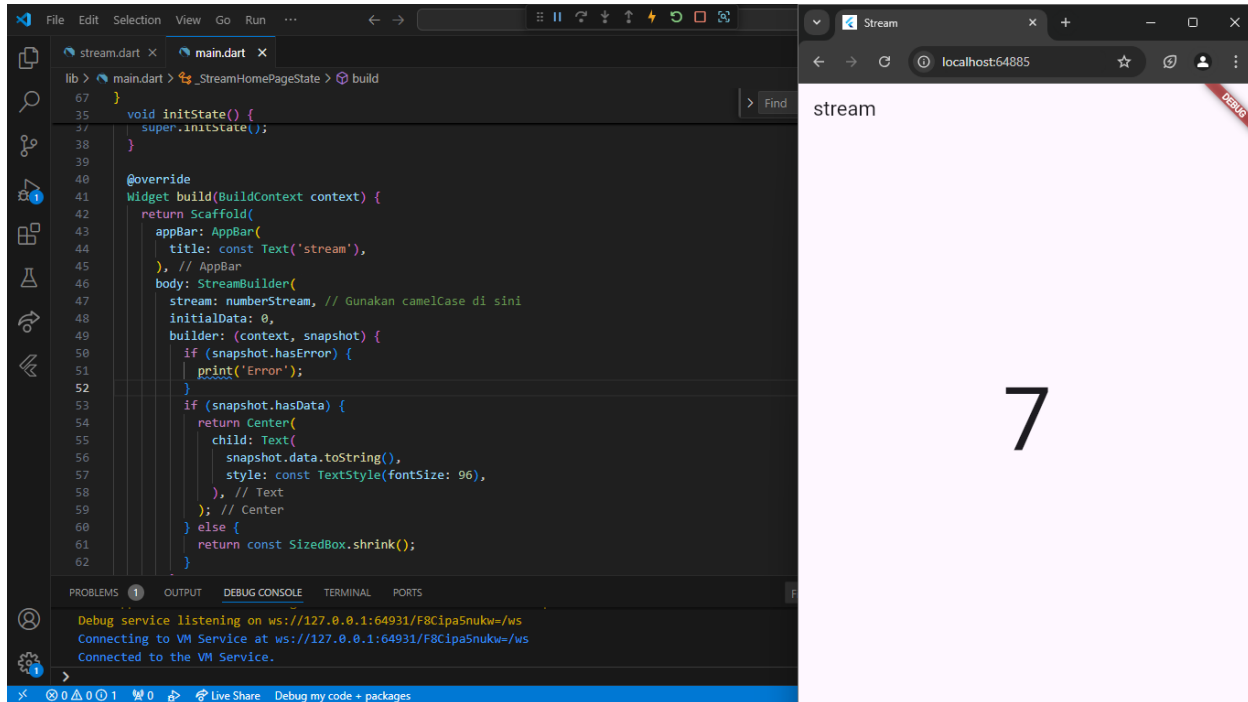
Langkah 7: Edit method build()

```
body: StreamBuilder(
  stream: numberStream, // Gunakan camelCase di sini
  initialData: 0,
  builder: (context, snapshot) {
    if (snapshot.hasError) {
      print('Error');
    }
    if (snapshot.hasData) {
      return Center(
        child: Text(
          snapshot.data.toString(),
          style: const TextStyle(fontSize: 96),
        ),
      );
    } else {
      return const SizedBox.shrink();
    }
  },
),
);
}
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Langkah 8: Run



Soal 12

- **Jelaskan maksud kode pada langkah 3 dan 7 !**
 - Langkah 3 : membuat fungsi getNumbers(), dan menampilkan nilai baru setiap 1 detik, objek Random menghasilkan nilai baru setiap 1 detik, dan di return menjadi elemen stream.
 - Langkah 7: kode tersebut berfungsi untuk menampilkan fungsi dari getNumbers, dan jika terjadi eror maka akan muncul pesan eror.
- **Capture hasil praktikum Anda berupa GIF dan lampirkan di README.**
- **Lalu lakukan commit dengan pesan "W13: Jawaban Soal 12".**

Praktikum 7

Langkah 1: Buat Project baru



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

```
lib > random_bloc.dart > ...  
1 import 'dart:async';  
2 import 'dart:math';
```

Langkah 2: Isi kode random_bloc.dart

```
lib > random_bloc.dart > ...  
1 import 'dart:async';  
2 import 'dart:math';
```

Langkah 3: Buat class RandomNumberBloc()

```
class Randomnumberbloc {}
```

Langkah 4: Buat variabel StreamController

```
import 'dart:async';  
import 'dart:math';  
  
class Randomnumberbloc {}  
  final _generateRandomController = StreamController<void>();  
  final _randomNumberController = StreamController<int>();  
  
  Sink<void> get generateRandom => _generateRandomController.sink;  
  Stream<int> get randomNumber => _randomNumberController.stream;  
  secondsStreamController.sink;
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Langkah 5: Buat constructor

```
RandomNumberBloc(){  
  _generateRandomController.stream.listen((_){}  
  final random = Random().nextInt(10);  
});  
}
```

Langkah 6: Buat method dispose()

```
void dispose(){  
  _generateRandomController.close();  
  _randomNumberController.close();  
}
```

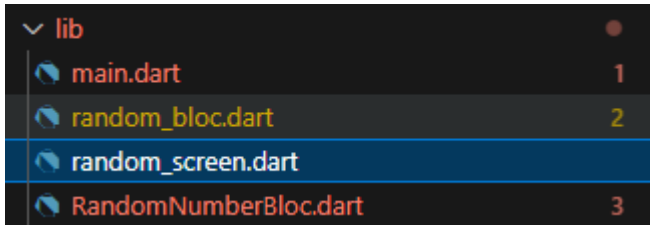
Langkah 7: Edit main.dart

```
main.dart 1 x RandomNumberBloc.dart 3  
lib > main.dart > MyApp > build  
1 import 'package:flutter/material.dart';  
2  
Run | Debug | Profile  
3 void main() {  
4   runApp(const MyApp());  
5 }  
6  
7 class MyApp extends StatelessWidget {  
8   const MyApp({super.key});  
9   @override  
10  Widget build(BuildContext context) {  
11    return MaterialApp(  
12      title: 'Flutter Demo',  
13      theme: ThemeData(  
14        primarySwatch: Colors.blue,  
15      ), // ThemeData  
16      home: const RandomScreen(),  
17    ); // MaterialApp  
18  }  
19 }
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

Langkah 8: Buat file baru random_screen.dart



Langkah 9: Lakukan impor material dan random_bloc.dart

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

Langkah 10: Buat StatefulWidget RandomScreen

```
import 'package:flutter/material.dart';  
import 'package:bloc_random_rifki/RandomNumberBloc.dart';  
  
class RandomScreen extends StatefulWidget {  
  @override  
  _RandomScreenState createState() => _RandomScreenState();  
}  
  
class _RandomScreenState extends State<RandomScreen> {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text("Random Screen"),  
      ),  
      body: Center(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: [  
            // Tambahkan widget di sini  
          ],  
        ),  
      ),  
    );  
  }  
}
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

```
}
```

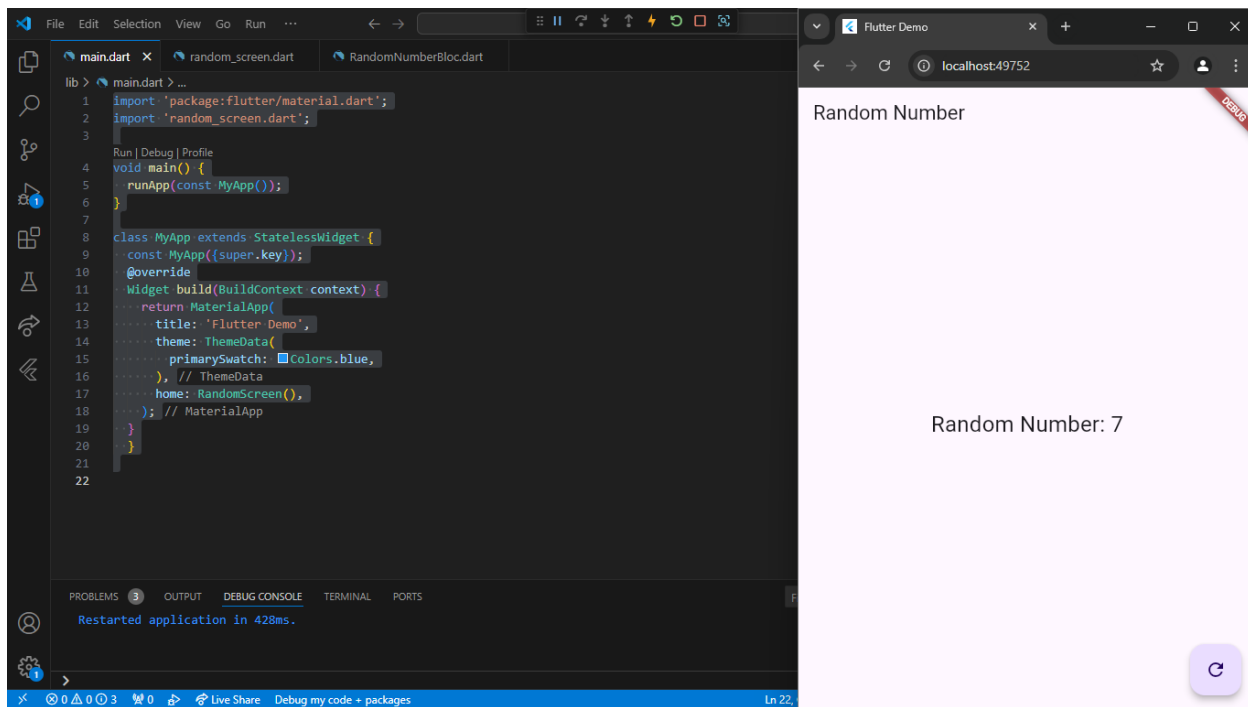
Langkah 11: Buat variabel

```
final bloc = Randomnumberbloc();
```

Langkah 12: Buat method dispose()

```
@override  
void dispose() {  
  _bloc.dispose();  
  super.dispose();  
}
```

Langkah 13: Edit method build()



Soal 13

- **Jelaskan maksud praktikum ini ! Dimanakah letak konsep pola BLoC-nya ?**
 - Pola Blo Cnya, untuk logika bisnis sendiri terdapat di kelas Randomnumberbloc. Semua proses pembuatan bilangan acak dilakukan di sini. dan **UI (User Interface)** berada di widget



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet-12: Streams
Mata Kuliah Pemograman Mobile

RandomScreen, yang hanya bertugas menampilkan data dari Stream tanpa mengetahui bagaimana data tersebut dihasilkan.

- **Capture hasil praktikum Anda berupa GIF dan lampirkan di README.**
- **Lalu lakukan commit dengan pesan "W13: Jawaban Soal 13".**