

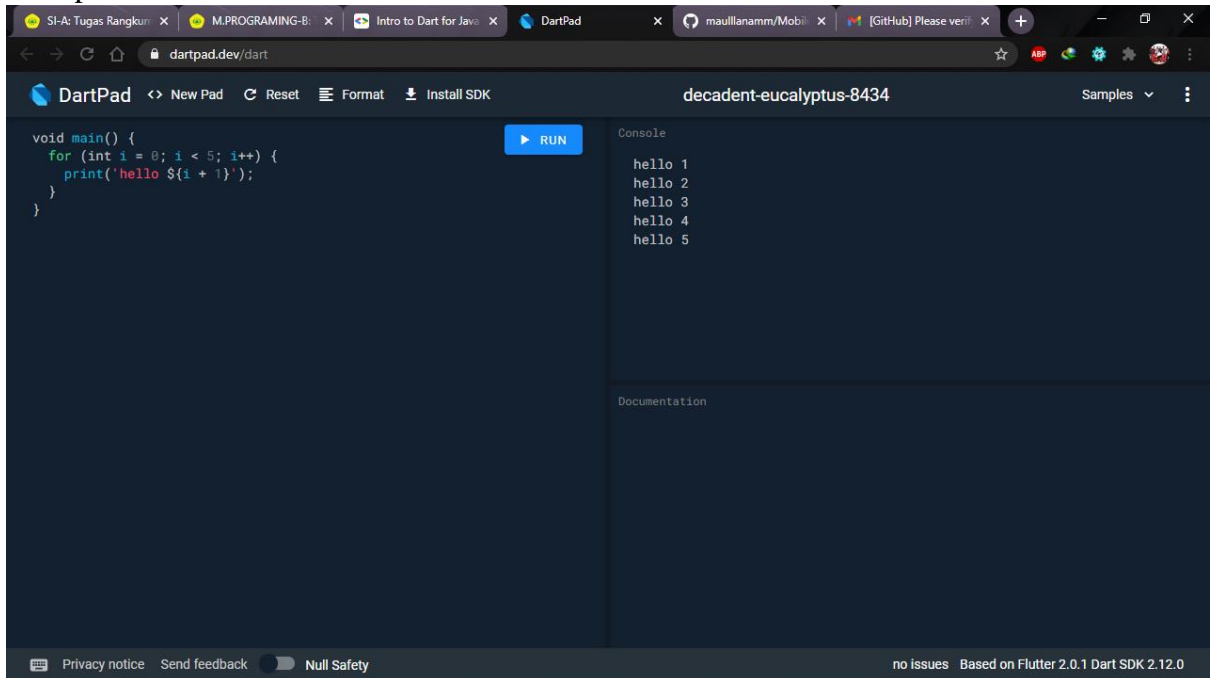
Nama : Maulana Muhammad

NPM : 065117167

---

## Tugas 1 Mobile Programing Dart

### 1. Start point



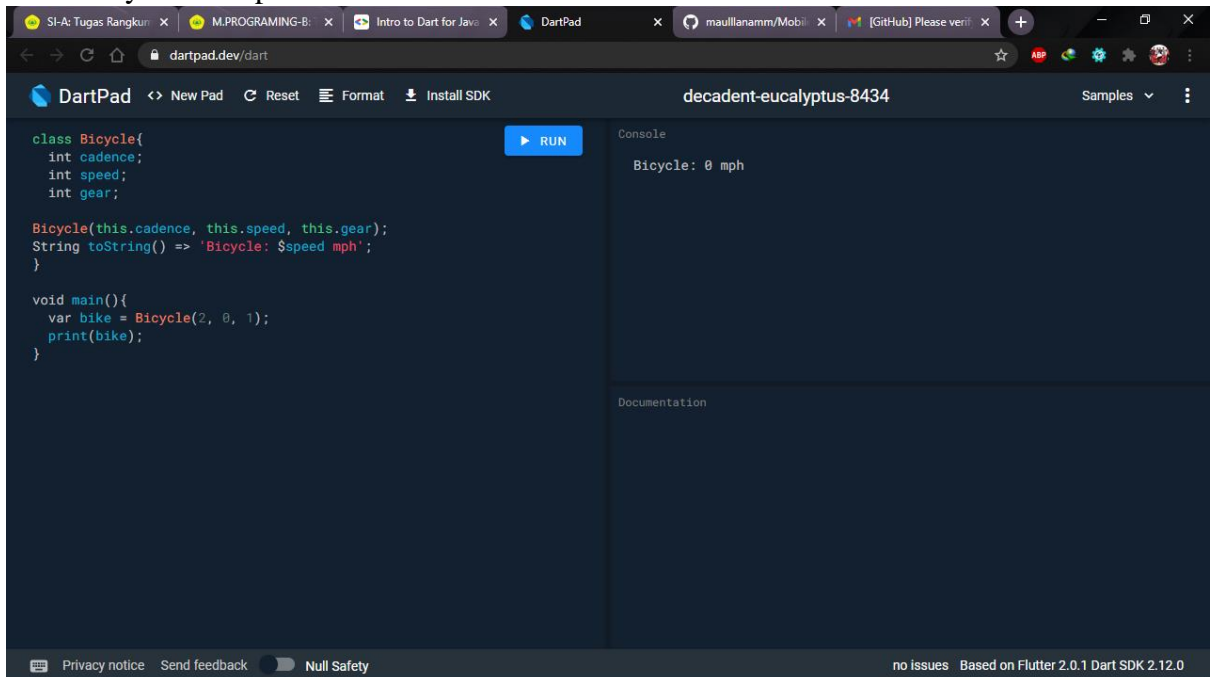
The screenshot shows the DartPad web interface. The code editor contains a Dart program that prints 'hello' five times. The console output shows the results of the program execution.

```
void main() {  
  for (int i = 0; i < 5; i++) {  
    print('hello ${i + 1}');  
  }  
}
```

Console output:

```
hello 1  
hello 2  
hello 3  
hello 4  
hello 5
```

### 2. Basic bicycle example



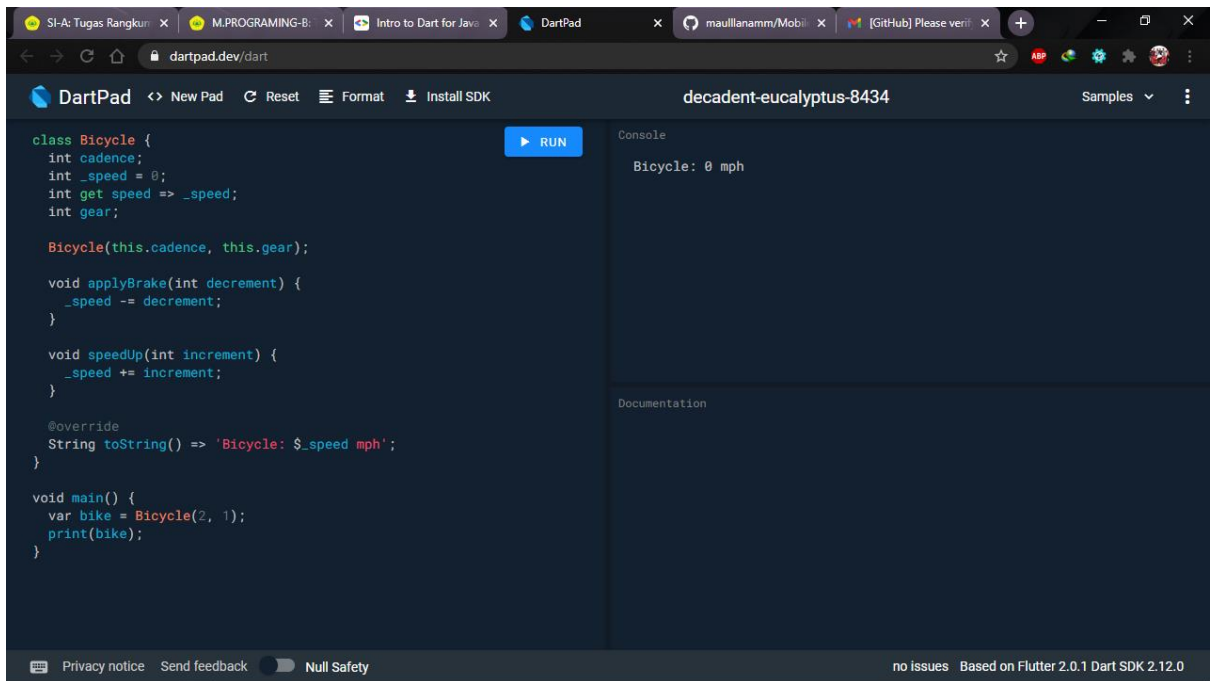
The screenshot shows the DartPad web interface. The code editor contains a Dart program that defines a `Bicycle` class and creates an instance of it. The console output shows the string representation of the `Bicycle` object.

```
class Bicycle {  
  int cadence;  
  int speed;  
  int gear;  
  
  Bicycle(this.cadence, this.speed, this.gear);  
  String toString() => 'Bicycle: $speed mph';  
}  
  
void main() {  
  var bike = Bicycle(2, 0, 1);  
  print(bike);  
}
```

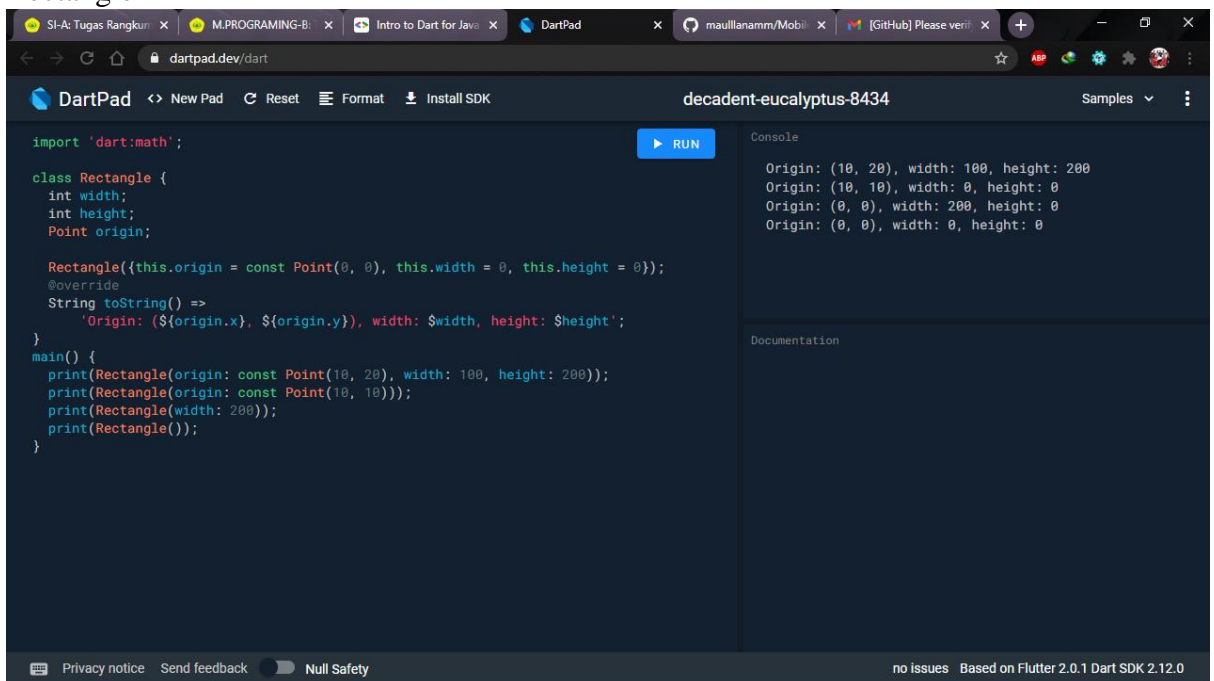
Console output:

```
Bicycle: 0 mph
```

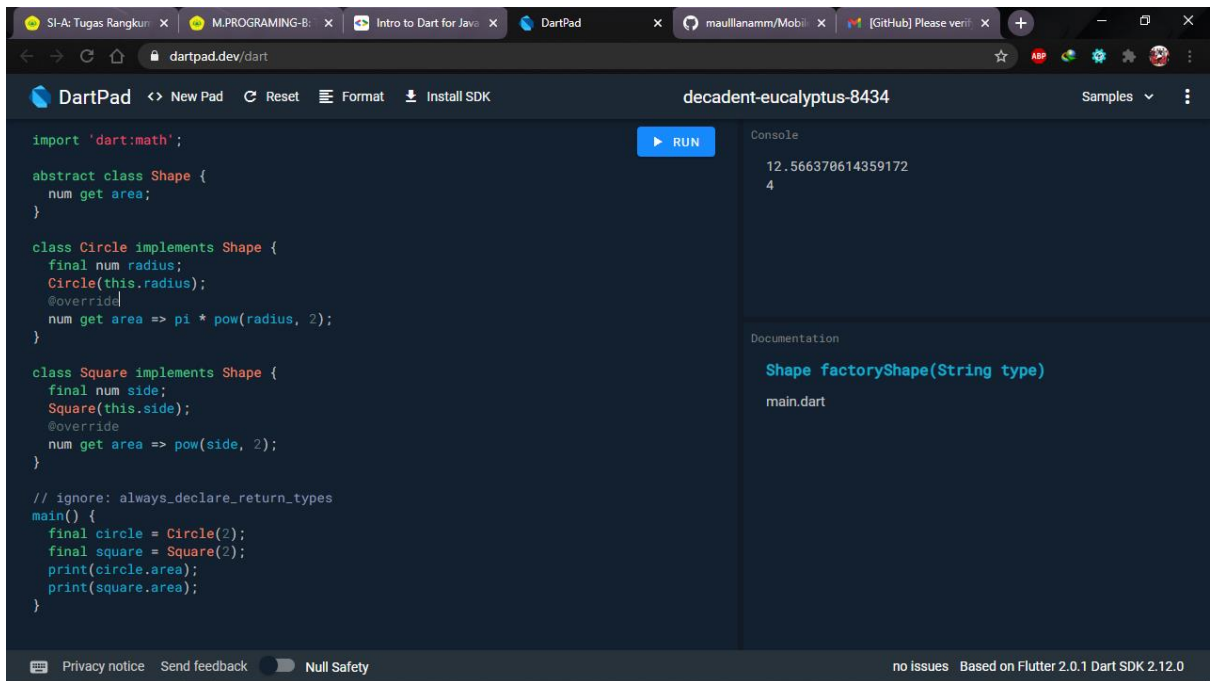
### 3. Final bicycle



#### 4. Rectangle



#### 5. Shape



```
import 'dart:math';

abstract class Shape {
  num get area;
}

class Circle implements Shape {
  final num radius;
  Circle(this.radius);
  @override
  num get area => pi * pow(radius, 2);
}

class Square implements Shape {
  final num side;
  Square(this.side);
  @override
  num get area => pow(side, 2);
}

// ignore: always_declare_return_types
main() {
  final circle = Circle(2);
  final square = Square(2);
  print(circle.area);
  print(square.area);
}
```

Console

```
12.566370614359172
4
```

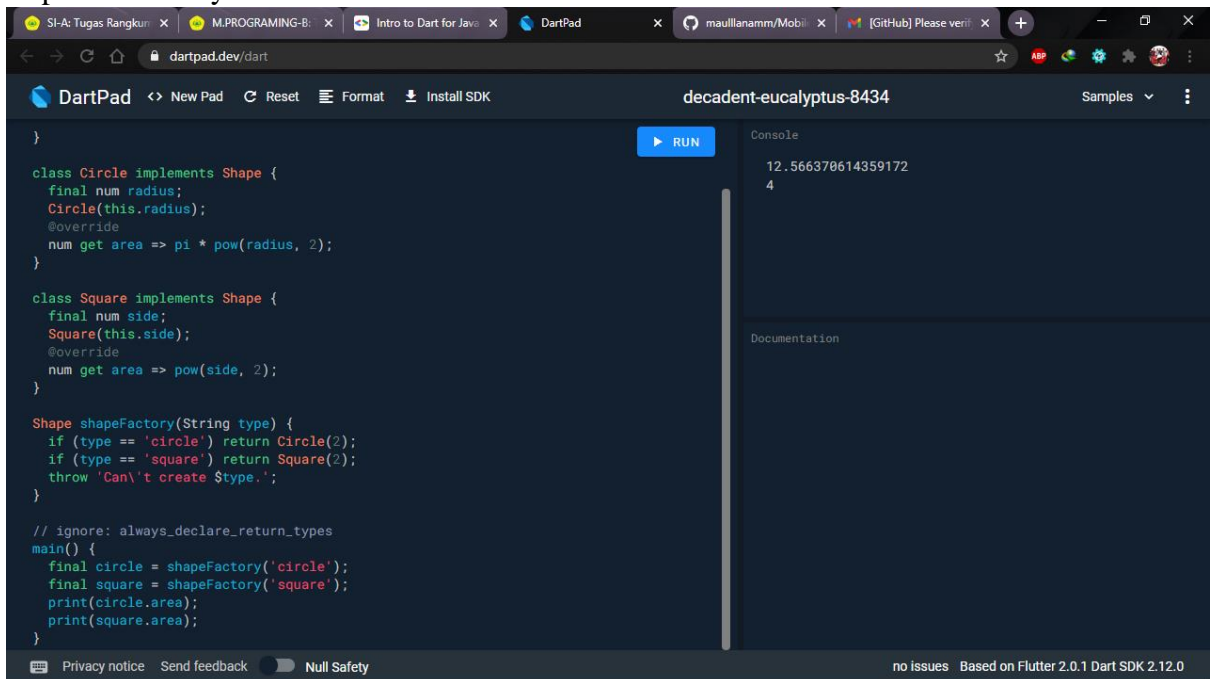
Documentation

Shape factoryShape(String type)

main.dart

Privacy notice Send feedback Null Safety no issues Based on Flutter 2.0.1 Dart SDK 2.12.0

## 6. Top level factory function



```
}

class Circle implements Shape {
  final num radius;
  Circle(this.radius);
  @override
  num get area => pi * pow(radius, 2);
}

class Square implements Shape {
  final num side;
  Square(this.side);
  @override
  num get area => pow(side, 2);
}

Shape shapeFactory(String type) {
  if (type == 'circle') return Circle(2);
  if (type == 'square') return Square(2);
  throw 'Can\'t create $type.';
}

// ignore: always_declare_return_types
main() {
  final circle = shapeFactory('circle');
  final square = shapeFactory('square');
  print(circle.area);
  print(square.area);
}
```

Console

```
12.566370614359172
4
```

Documentation

Privacy notice Send feedback Null Safety no issues Based on Flutter 2.0.1 Dart SDK 2.12.0

## 7. Factory constructure

The screenshot shows the DartPad web interface. The code defines a `Shape` factory, `Circle` and `Square` classes implementing `Shape`, and a `main` function. The console output shows the area of a circle (12.566370614359172) and a square (4).

```
factory Shape(String type) {
  if (type == 'circle') return Circle(2);
  if (type == 'square') return Square(2);
  throw 'Can\'t create $type.';
}

num get area;

class Circle implements Shape {
  final num radius;
  Circle(this.radius);
  @override
  num get area => pi * pow(radius, 2);
}

class Square implements Shape {
  final num side;
  Square(this.side);
  @override
  num get area => pow(side, 2);
}

// ignore: always_declare_return_types
main() {
  final circle = Shape('circle');
  final square = Shape('square');
  print(circle.area);
  print(square.area);
}
```

Console output:

```
12.566370614359172
4
```

## 8. Circle mockflow

The screenshot shows the DartPad web interface with the same code as before, but with an additional `CircleMock` class. The console output is the same as the previous screenshot.

```
factory Shape(String type) {
  if (type == 'circle') return Circle(2);
  if (type == 'square') return Square(2);
  throw 'Can\'t create $type.';
}

num get area;

class Circle implements Shape {
  final num radius;
  Circle(this.radius);
  @override
  num get area => pi * pow(radius, 2);
}

class Square implements Shape {
  final num side;
  Square(this.side);
  @override
  num get area => pow(side, 2);
}

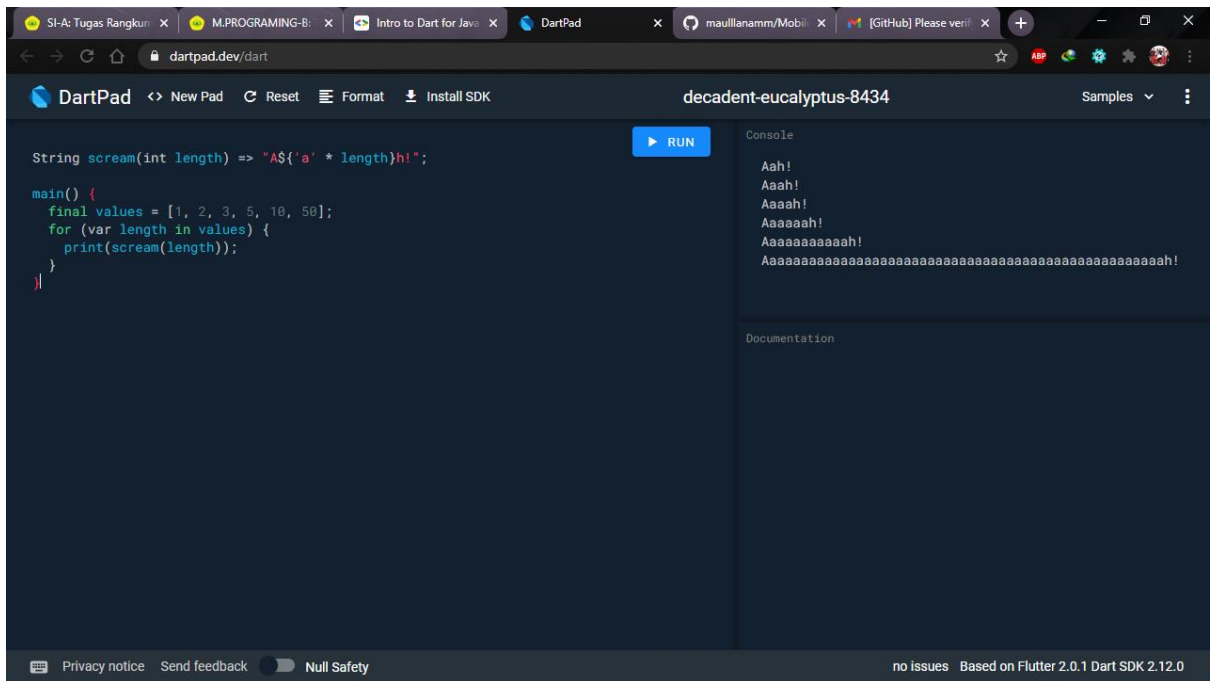
class CircleMock implements Circle {
  num area;
  num radius;
}

// ignore: always_declare_return_types
main() {
  final circle = Shape('circle');
  final square = Shape('square');
  print(circle.area);
  print(square.area);
}
```

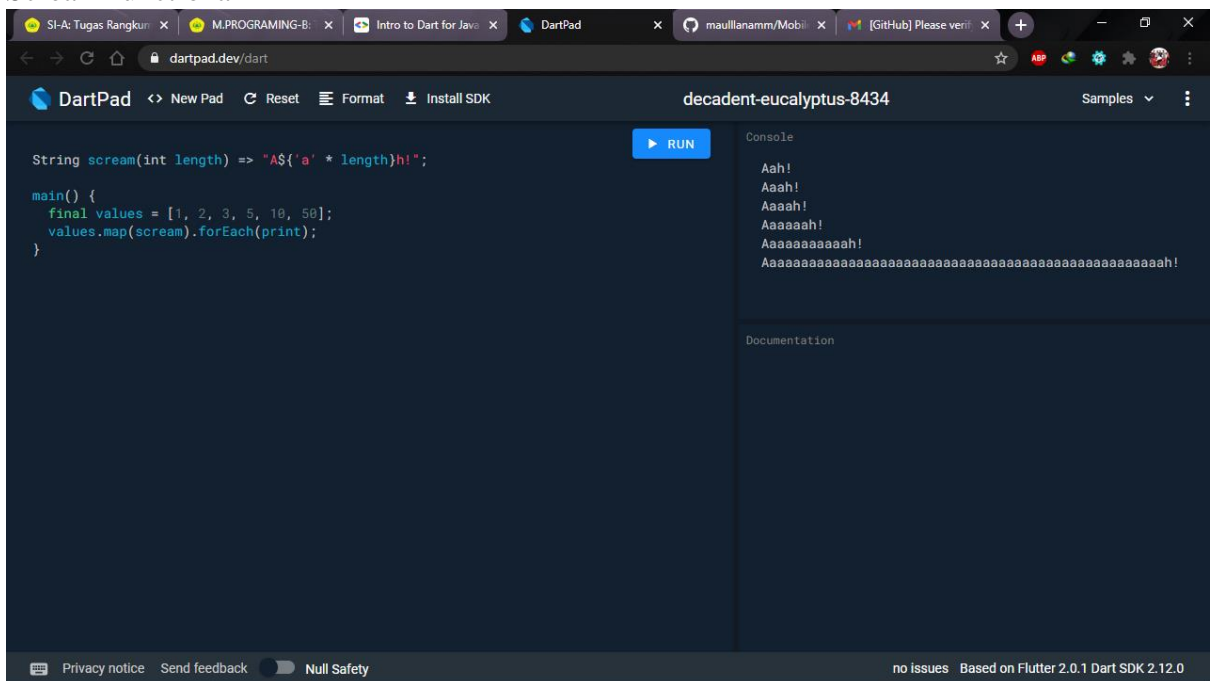
Console output:

```
12.566370614359172
4
```

## 9. Scream implement



## 10. Scream functional



## 11. Scream iterableAPI

SI-A: Tugas Rangkun x M.PROGRAMING-B: x Intro to Dart for Java x DartPad x maullanammi/Mobi x [GitHub] Please veri x

dartpad.dev/dart

DartPad

New Pad Reset Format Install SDK

decadent-eucalyptus-8434

Samples

```
String scream(int length) => "A${'a' * length}h!";

main() {
  final values = [1, 2, 3, 5, 10, 50];
  values.skip(1).take(3).map(scream).forEach(print);
}
```

RUN

Console

Aaah!  
Aaaah!  
Aaaaaah!

Documentation

Privacy notice Send feedback ☐ Null Safety

no issues Based on Flutter 2.0.1 Dart SDK 2.12.0