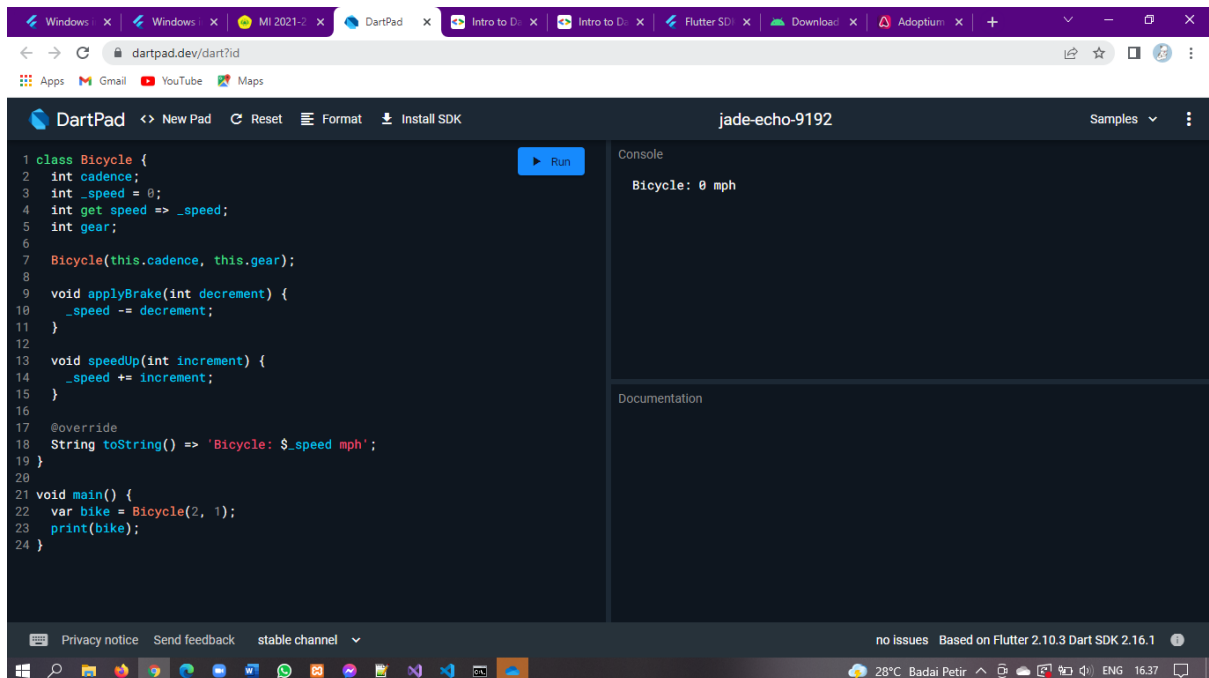


Nama : Nabila Oktaviani

NPM : 085020008

## Tugas Pemrograman Mobile 1

### 1. Bycicle

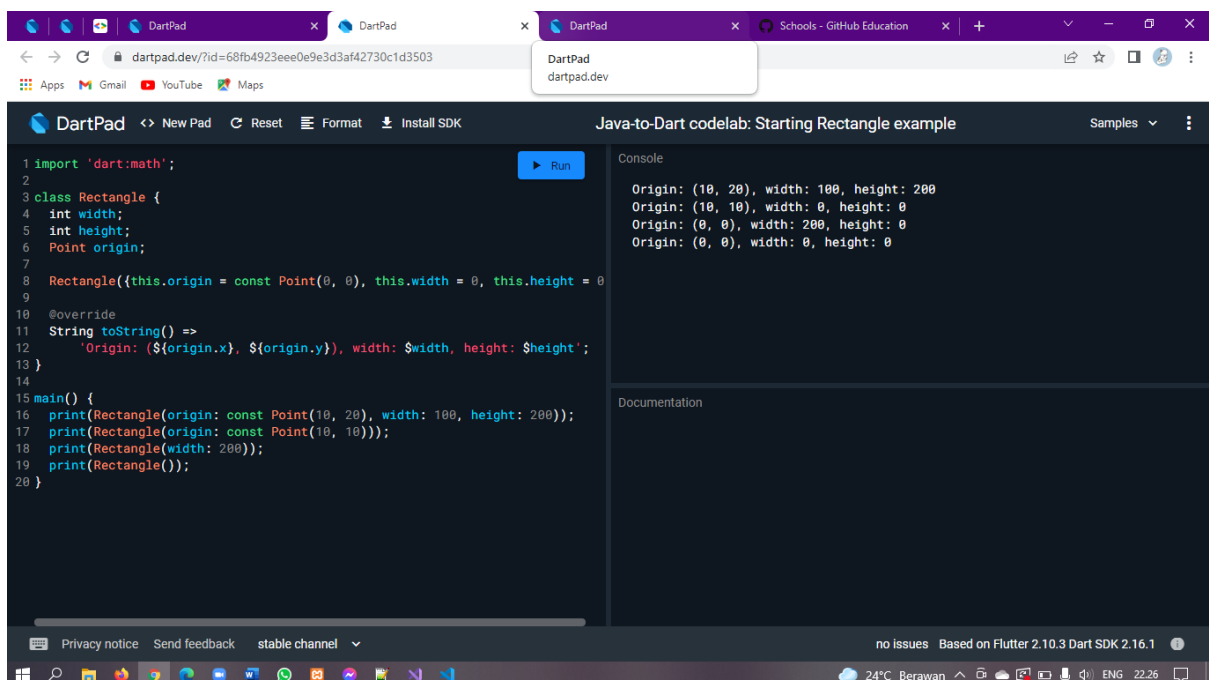


The screenshot shows the DartPad web interface. The code editor contains a Dart class named `Bicycle` with the following code:

```
1 class Bicycle {
2   int cadence;
3   int _speed = 0;
4   int get speed => _speed;
5   int gear;
6
7   Bicycle(this.cadence, this.gear);
8
9   void applyBrake(int decrement) {
10    _speed -= decrement;
11  }
12
13  void speedUp(int increment) {
14    _speed += increment;
15  }
16
17  @override
18  String toString() => 'Bicycle: $_speed mph';
19 }
20
21 void main() {
22   var bike = Bicycle(2, 1);
23   print(bike);
24 }
```

The console output shows: `Bicycle: 0 mph`. The status bar at the bottom indicates "no issues Based on Flutter 2.10.3 Dart SDK 2.16.1".

### 2. Rectangle



The screenshot shows the DartPad web interface. The code editor contains a Dart class named `Rectangle` with the following code:

```
1 import 'dart:math';
2
3 class Rectangle {
4   int width;
5   int height;
6   Point origin;
7
8   Rectangle({this.origin = const Point(0, 0), this.width = 0, this.height = 0}) {
9     width = width;
10    height = height;
11  }
12
13  @override
14  String toString() => 'Origin: (${origin.x}, ${origin.y}), width: $width, height: $height';
15 }
16
17 void main() {
18   print(Rectangle(origin: const Point(10, 20), width: 100, height: 200));
19   print(Rectangle(origin: const Point(10, 10)));
20   print(Rectangle(width: 200));
21   print(Rectangle());
22 }
```

The console output shows: `Origin: (10, 20), width: 100, height: 200`, `Origin: (10, 10), width: 0, height: 0`, `Origin: (0, 0), width: 200, height: 0`, and `Origin: (0, 0), width: 0, height: 0`. The status bar at the bottom indicates "no issues Based on Flutter 2.10.3 Dart SDK 2.16.1".

### 3. Circle Class

The screenshot shows the DartPad interface with the title "Java-to-Dart codelab: Starting Shapes example". The code editor contains the following Dart code:

```
1 import 'dart:math';
2
3 abstract class Shape {
4   num get area;
5 }
6
7 class Circle implements Shape {
8   final num radius;
9   Circle(this.radius);
10  num get area => pi * pow(radius, 2);
11 }
12
13 class Square implements Shape {
14   final num side;
15   Square(this.side);
16   num get area => pow(side, 2);
17 }
18
19 main() {
20   final circle = Circle(2);
21   final square = Square(2);
22   print(circle.area);
23   print(square.area);
24 }
```

The console output shows the results of the program execution:

```
12.566370614359172
4
```

The bottom status bar indicates "2 issues show" and "Based on Flutter 2.10.3 Dart SDK 2.16.1".

### 4. Circle Mock

The screenshot shows the DartPad interface with the title "Java-to-Dart codelab: Starting Rectangle example". The code editor contains the following Dart code:

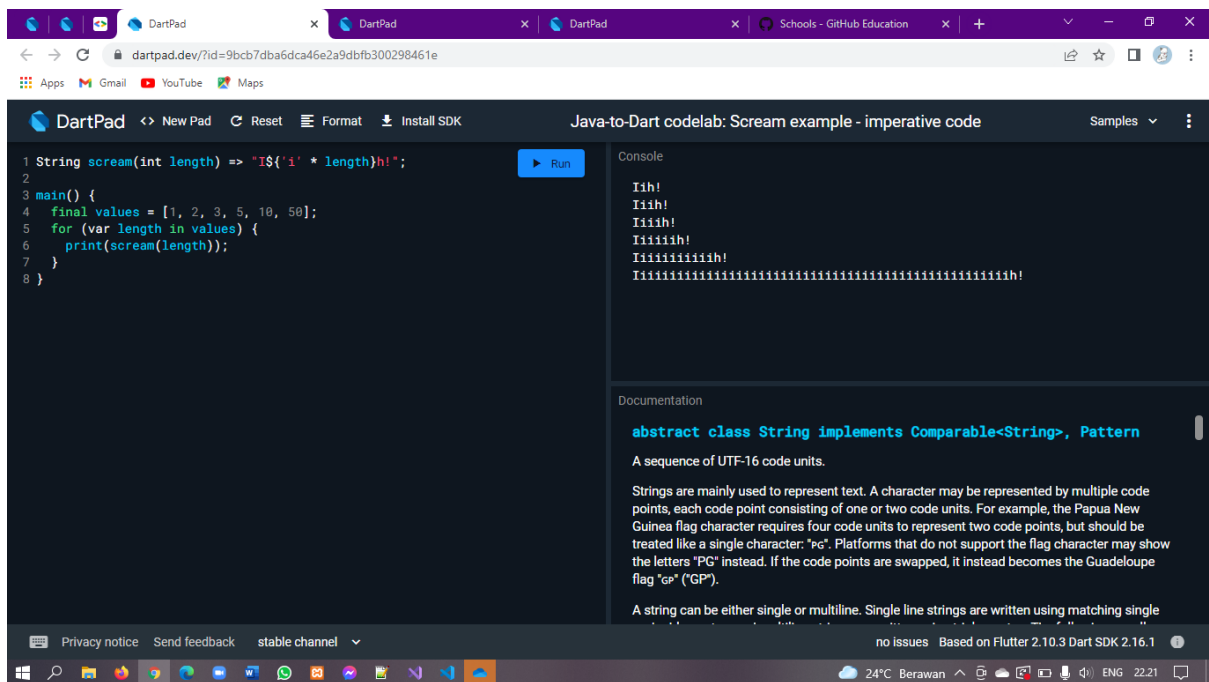
```
1 import 'dart:math';
2
3 abstract class Shape {
4   factory Shape(String type) {
5     if (type == 'circle') return Circle(2);
6     if (type == 'square') return Square(2);
7     throw 'Can\'t create $type.';
8   }
9   num get area;
10 }
11
12 class Circle implements Shape {
13   final num radius;
14   Circle(this.radius);
15   num get area => pi * pow(radius, 2);
16 }
17
18 class Square implements Shape {
19   final num side;
20   Square(this.side);
21   num get area => pow(side, 2);
22 }
23
24 class CircleMock implements Circle {
25   num area = 0;
26   num radius = 0;
27 }
28
```

The console output shows the results of the program execution:

```
12.566370614359172
4
```

The bottom status bar indicates "4 issues show" and "Based on Flutter 2.10.3 Dart SDK 2.16.1".

## 5. String Scream



The screenshot shows the DartPad web interface with the following components:

- Browser Tabs:** DartPad, DartPad, DartPad, Schools - GitHub Education.
- Address Bar:** dartpad.dev/?id=9bcb7dba6dca46e2a9dbfb300298461e
- Navigation Bar:** Apps, Gmail, YouTube, Maps.
- DartPad Header:** DartPad, <> New Pad, Reset, Format, Install SDK, Java-to-Dart codelab: Scream example - imperative code, Samples.
- Code Editor:**

```
1 String scream(int length) => "I${'i' * length}h!";
2
3 main() {
4   final values = [1, 2, 3, 5, 10, 50];
5   for (var length in values) {
6     print(scream(length));
7   }
8 }
```
- Run Button:** A blue button with a play icon and the text "Run".
- Console:**

```
Iih!
Iiih!
Iiiih!
Iiiiiih!
Iiiiiiiih!
Iiiiiiiiiih!
```
- Documentation:**

**abstract class String implements Comparable<String>, Pattern**

A sequence of UTF-16 code units.

Strings are mainly used to represent text. A character may be represented by multiple code points, each code point consisting of one or two code units. For example, the Papua New Guinea flag character requires four code units to represent two code points, but should be treated like a single character: "ꠘꠘ". Platforms that do not support the flag character may show the letters "PG" instead. If the code points are swapped, it instead becomes the Guadeloupe flag "Gꠘ" ("GP").

A string can be either single or multiline. Single line strings are written using matching single
- Footer:** Privacy notice, Send feedback, stable channel, no issues, Based on Flutter 2.10.3 Dart SDK 2.16.1, 24°C Berawan, ENG 22:21.