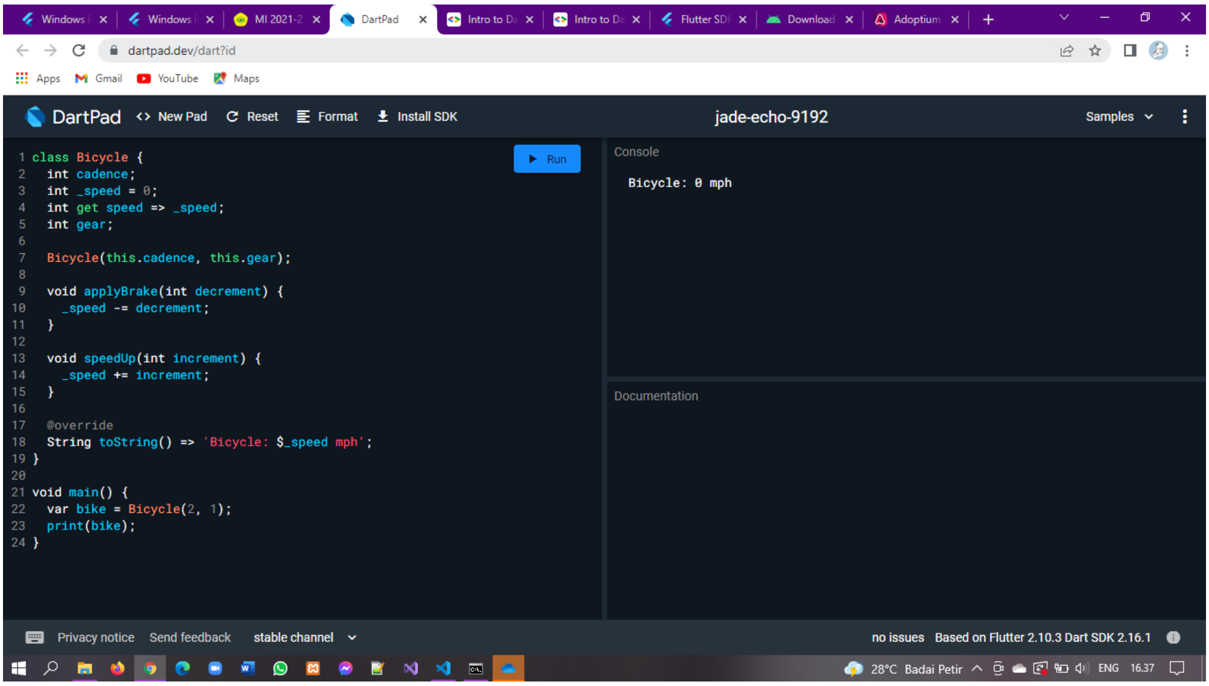
Nama : Nabila Oktaviani

NPM : 085020008

Tugas Pemrograman Mobile 1

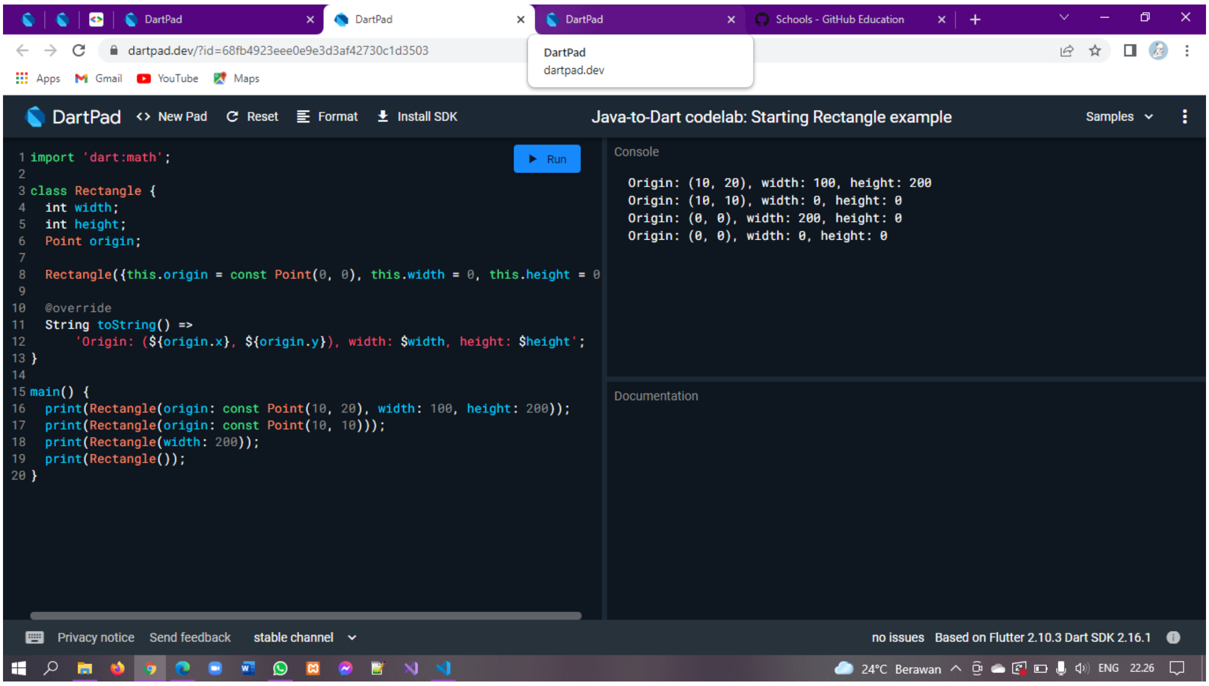
1. Bycicle



```
class Bicycle {
  int cadence;
  int _speed = 0;
  int get speed => _speed;
  int gear;

  Bicycle(this.cadence, this.gear);

  void applyBrake(int decrement) {
    _speed -= decrement;
  }

  void speedUp(int increment) {
    _speed += increment;
  }

  @override
  String toString() => 'Bicycle: $_speed mph';
}

void main() {
  var bike = Bicycle(2, 1);
  print(bike);
}
```

```
Bicycle: 0 mph
```

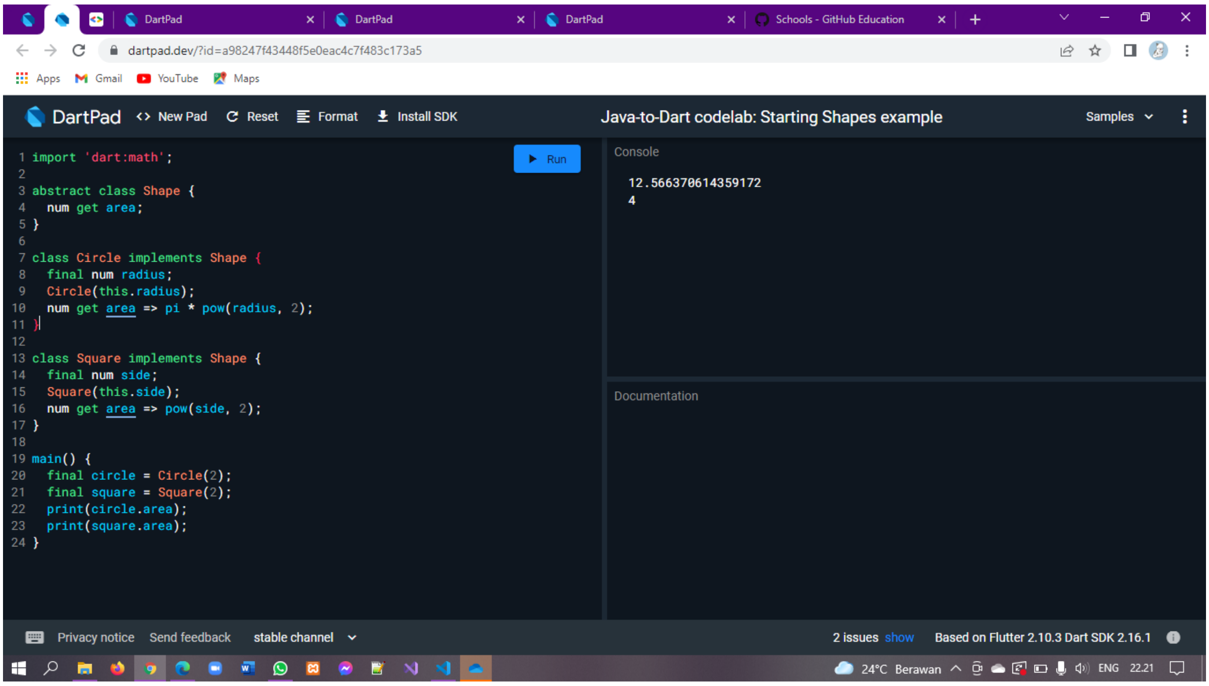
2. Rectangle



```
import 'dart:math';

class Rectangle {
  int width;
  int height;
  Point origin;

  Rectangle({this.origin = const Point(0, 0), this.width = 0, this.height = 0

  @override
  String toString() =>
      'Origin: (${origin.x}, ${origin.y}), width: $width, height: $height';
}

main() {
  print(Rectangle(origin: const Point(10, 20), width: 100, height: 200));
  print(Rectangle(origin: const Point(10, 10)));
  print(Rectangle(width: 200));
  print(Rectangle());
}
```

```
Origin: (10, 20), width: 100, height: 200
Origin: (10, 10), width: 0, height: 0
Origin: (0, 0), width: 200, height: 0
Origin: (0, 0), width: 0, height: 0
```

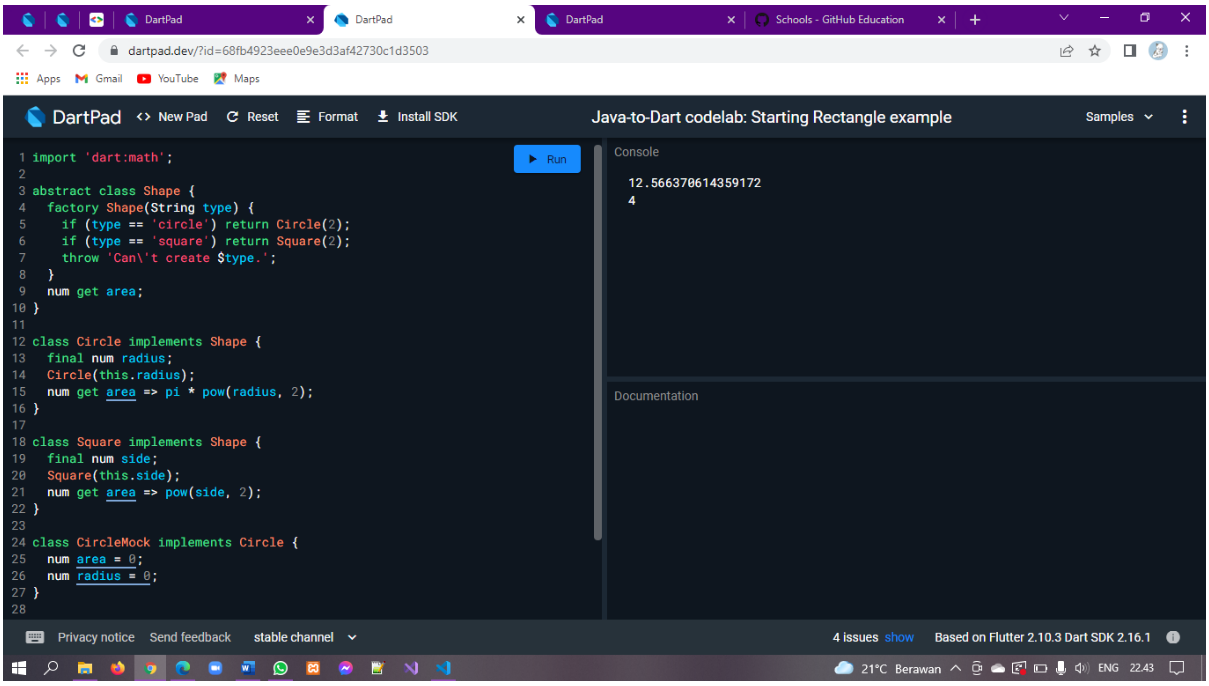
3. Circle Class



```dart
import 'dart:math';

abstract class Shape {
  num get area;
}

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

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

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

```
12.566370614359172
4
```

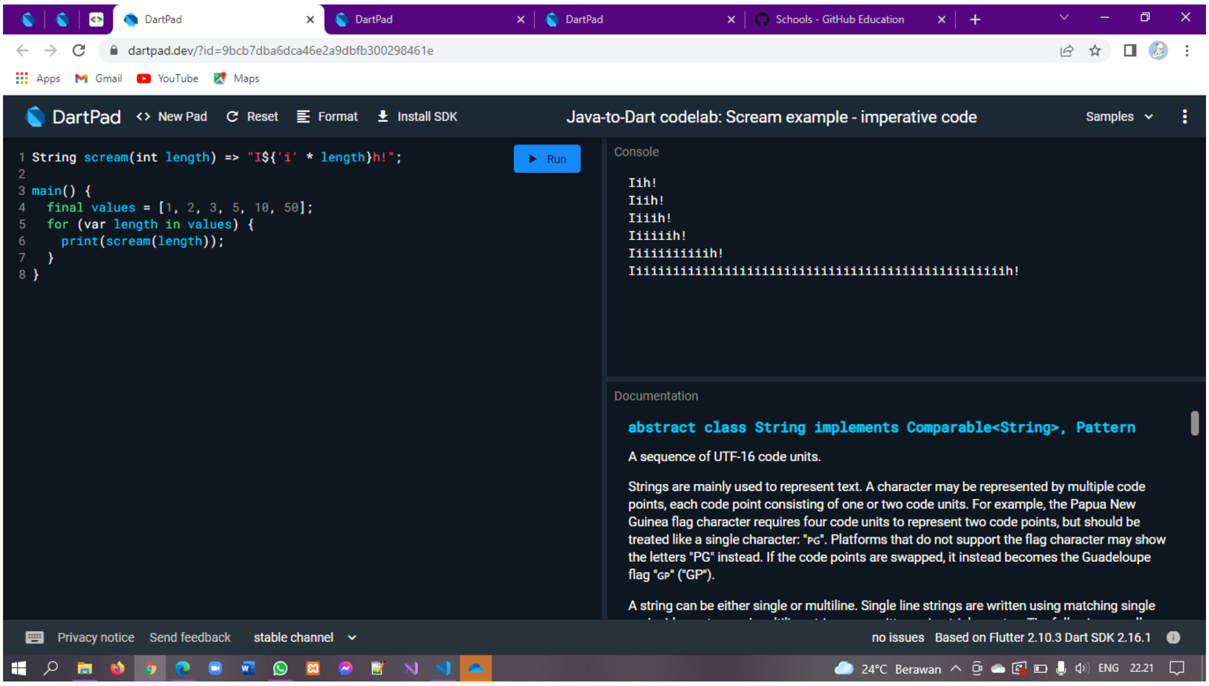
4. Circle Mock



```dart
import 'dart:math';

abstract class Shape {
  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);
  num get area => pi * pow(radius, 2);
}

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

class CircleMock implements Circle {
  num area = 0;
  num radius = 0;
}

```

```
12.566370614359172
4
```

5. String Scream



```dart
String scream(int length) => "I${'i' * length}h!";

main() {
  final values = [1, 2, 3, 5, 10, 50];
  for (var length in values) {
    print(scream(length));
  }
}
```

```
Iih!
Iiih!
Iiiih!
Iiiiiih!
Iiiiiiiiiiih!
Iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiih!
```

`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: "PG". 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 "GP" ("GP").

A string can be either single or multiline. Single line strings are written using matching single