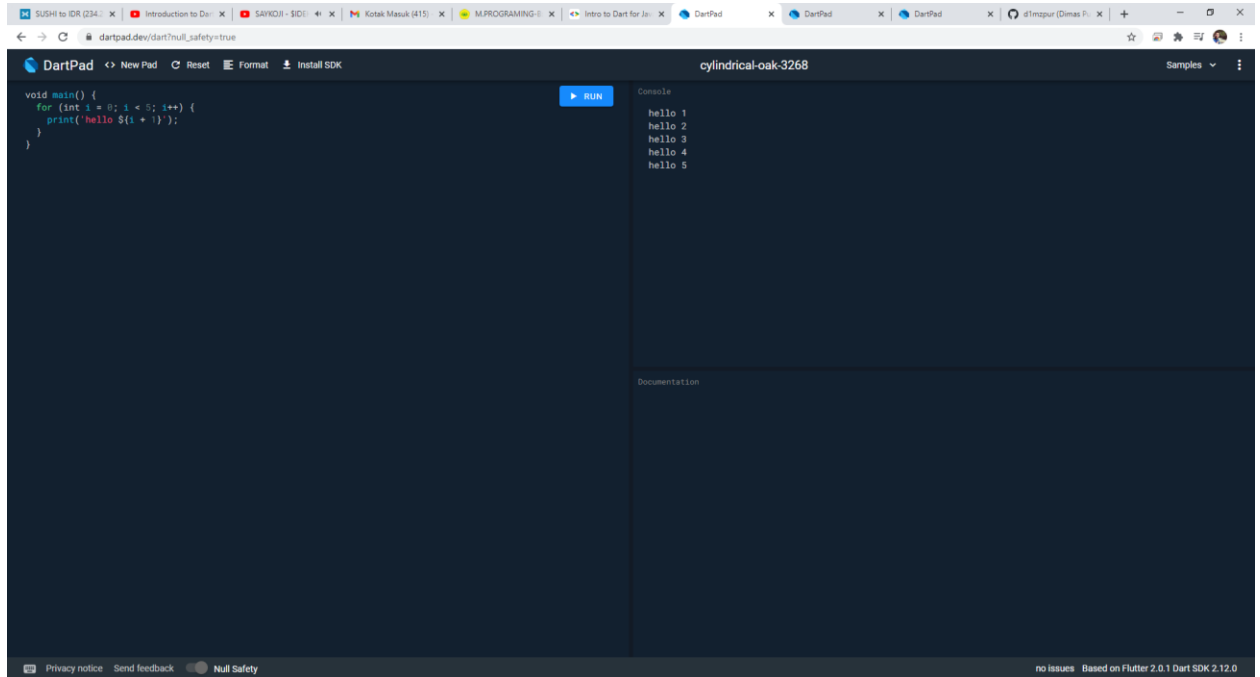


Nama : Dimas Purnomo
NPM : 065118211

1. StartPoint – main.dart



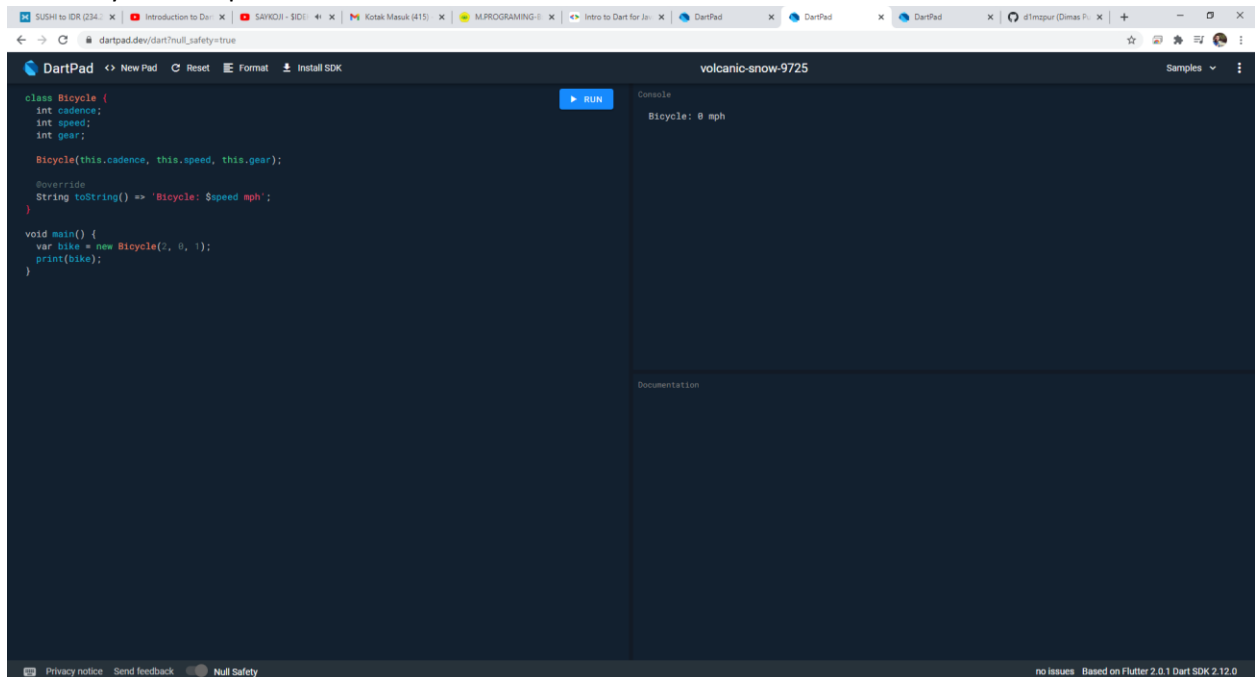
The screenshot shows the DartPad interface with a file named `main.dart`. The code is as follows:

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

The console output displays the following lines:

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

2. Basic Bicycle example – Constractor.dart



The screenshot shows the DartPad interface with a file named `Constractor.dart`. The code is as follows:

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

The console output displays the following line:

```
Bicycle: 0 mph
```

3. Final Bicycle – ReadOnly.dart

The screenshot shows the DartPad interface with the following Dart code:

```
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);
}
```

The console output shows: `Bicycle: 0 mph`. The status bar at the bottom indicates "no issues Based on Flutter 2.0.1 Dart SDK 2.12.0".

4. Rectangle – Rectangle.dart

The screenshot shows the DartPad interface with the following Dart code:

```
import 'dart:math';

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

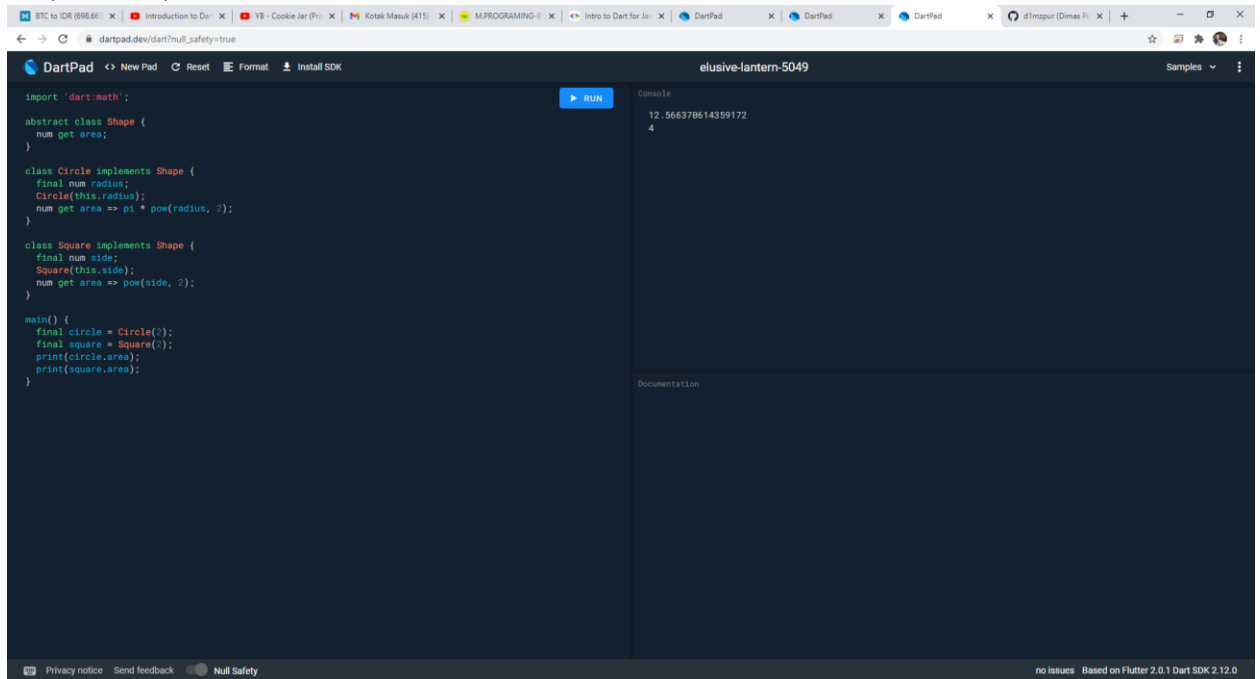
  Rectangle({this.origin = const Point(0, 0), this.width = 0, this.height = 0});

  @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());
}
```

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 2.0.1 Dart SDK 2.12.0".

5. Shapes – Shapes.dart



The screenshot shows the DartPad interface with the following code in the editor:

```
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);
}

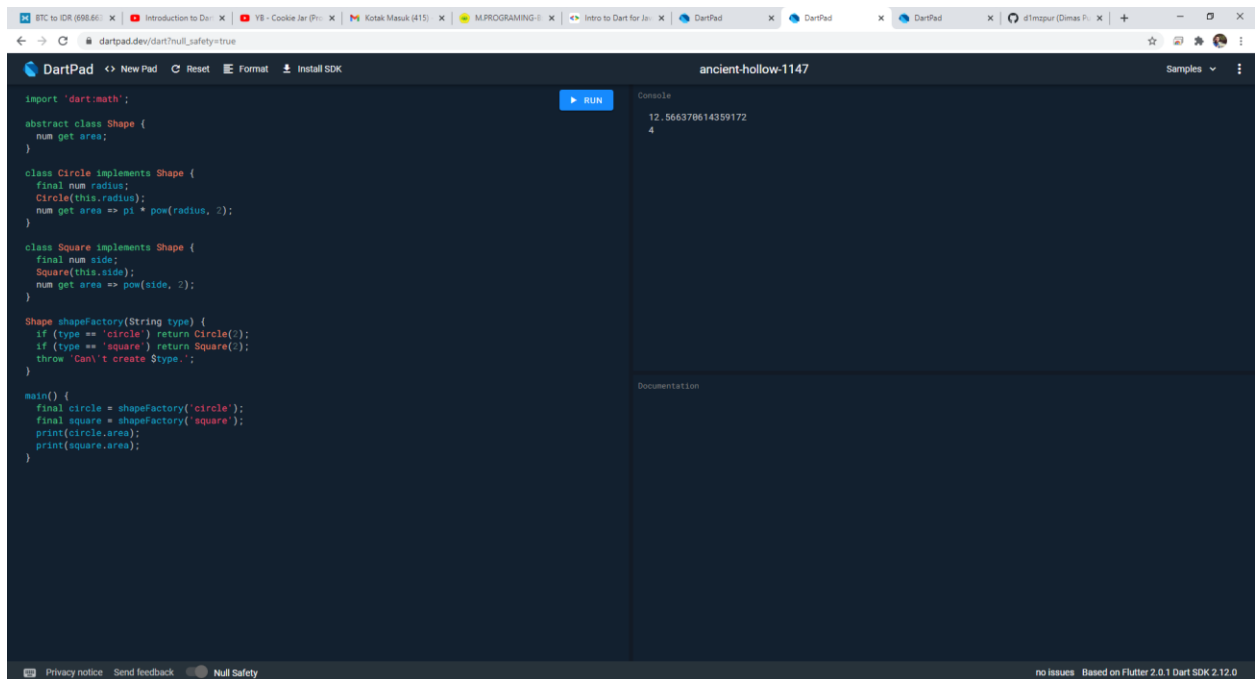
main() {
  final circle = Circle(2);
  final square = Square(2);
  print(circle.area);
  print(square.area);
}
```

The console output on the right shows the results of the program execution:

```
12.566370614359172
4
```

The status bar at the bottom indicates "no issues Based on Flutter 2.0.1 Dart SDK 2.12.0".

6. Top Level Factory Function – TopLevel.dart



The screenshot shows the DartPad interface with the following code in the editor:

```
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);
}

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

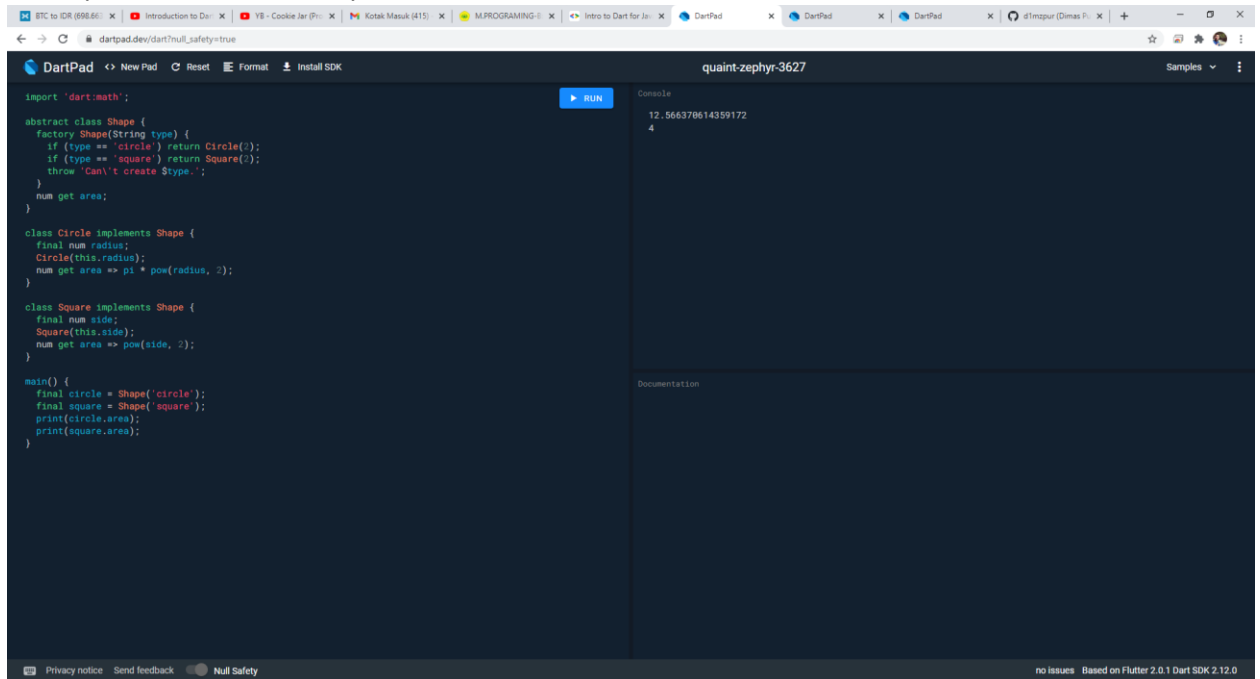
main() {
  final circle = shapeFactory('circle');
  final square = shapeFactory('square');
  print(circle.area);
  print(square.area);
}
```

The console output on the right shows the results of the program execution:

```
12.566370614359172
4
```

The status bar at the bottom indicates "no issues Based on Flutter 2.0.1 Dart SDK 2.12.0".

7. Factory Konstruktor – FactoryCon.dart



The screenshot shows the DartPad interface with the following Dart code:

```
import 'dart:math';

abstract class Shape {
  factory Shape(String type) {
    if (type == 'circle') return Circle();
    if (type == 'square') return Square();
    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);
}

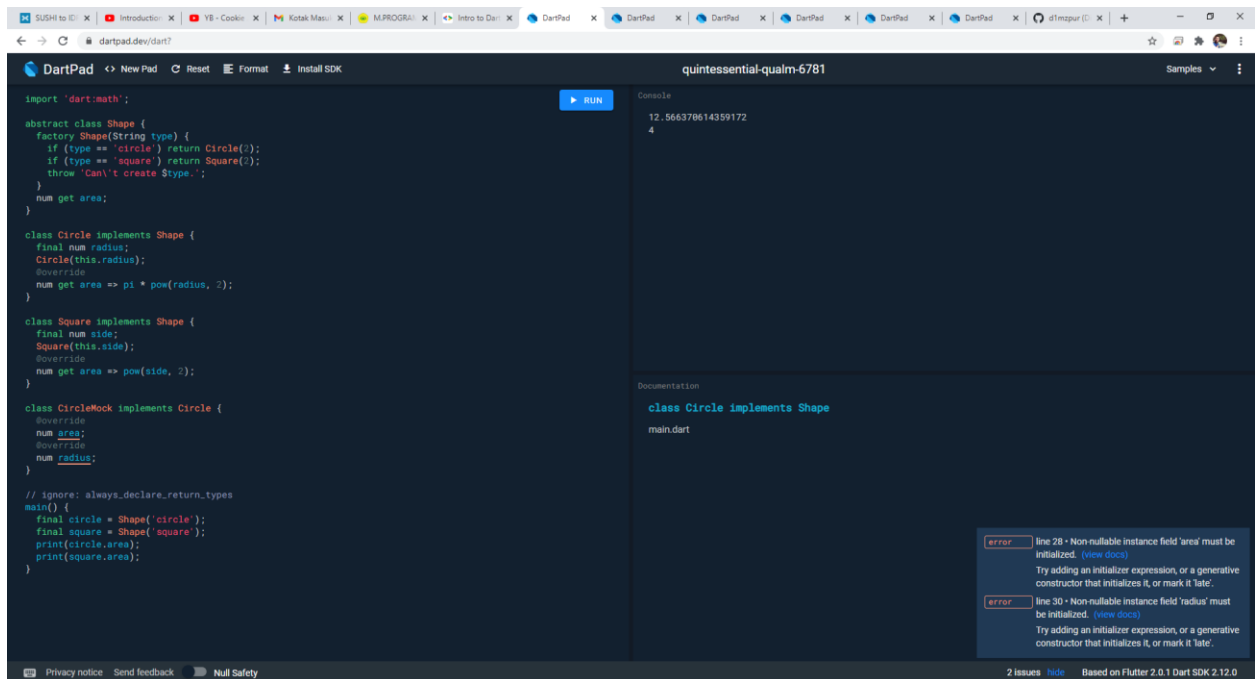
main() {
  final circle = Shape('circle');
  final square = Shape('square');
  print(circle.area);
  print(square.area);
}
```

The console output shows the execution results:

```
12.566370614359172
4
```

The status bar at the bottom indicates "no issues Based on Flutter 2.0.1 Dart SDK 2.12.0".

8. Circle Mockflow – CircleMork.dart



The screenshot shows the DartPad interface with the following Dart code:

```
import 'dart:math';

abstract class Shape {
  factory Shape(String type) {
    if (type == 'circle') return Circle();
    if (type == 'square') return Square();
    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 {
  @override
  num area;
  @override
  num radius;
}

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

The console output shows the execution results:

```
12.566370614359172
4
```

The status bar at the bottom indicates "2 issues Based on Flutter 2.0.1 Dart SDK 2.12.0".

The error messages in the bottom right corner are:

- error line 28 • Non-nullable instance field 'area' must be initialized. (view docs)
Try adding an initializer expression, or a generative constructor that initializes it, or mark it 'late'.
- error line 30 • Non-nullable instance field 'radius' must be initialized. (view docs)
Try adding an initializer expression, or a generative constructor that initializes it, or mark it 'late'.

9. Scream Implement – ScreamImp.dart

The screenshot shows the DartPad interface with the following code in the editor:

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

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

The console output on the right shows the results of the function calls:

```
Aah!
Aaah!
Aaaah!
Aaaaaah!
Aaaaaaaaaah!
Aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaah!
```

The status bar at the bottom indicates "no issues Based on Flutter 2.0.1 Dart SDK 2.12.0".

10. Scream Funcional – ScreamFung.dart

The screenshot shows the DartPad interface with the following code in the editor:

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

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

The console output on the right shows the results of the function calls:

```
Aah!
Aaah!
Aaaah!
Aaaaaah!
Aaaaaaaaaah!
Aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaah!
```

The status bar at the bottom indicates "no issues Based on Flutter 2.0.1 Dart SDK 2.12.0".

11. Scream IterableAPI

