

**LAPORAN TUGAS GUIDED
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL 3
PENGENALAN DART**



Disusun Oleh :
Aji Prasetyo Nugroho / 2211104049
SE-06-02

Asisten Praktikum :
Muhammad Faza Zulian Gesit Al Barru
Aisyah Hasna Aulia

Dosen Pengampu :
Yudha Islami Sulistya

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

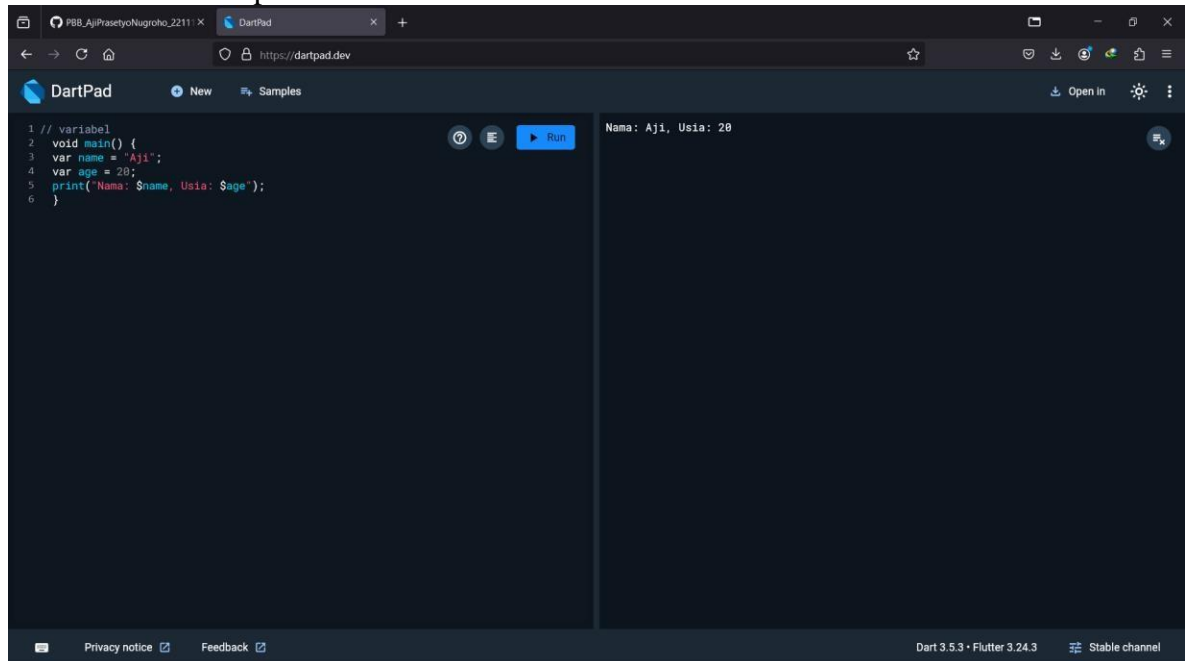
2024

GUIDED

A. Variable

Variabel dalam Dart digunakan untuk menyimpan nilai, baik berupa angka, teks, maupun objek lainnya, yang dapat digunakan dan diubah dalam program.

Source code & Output :

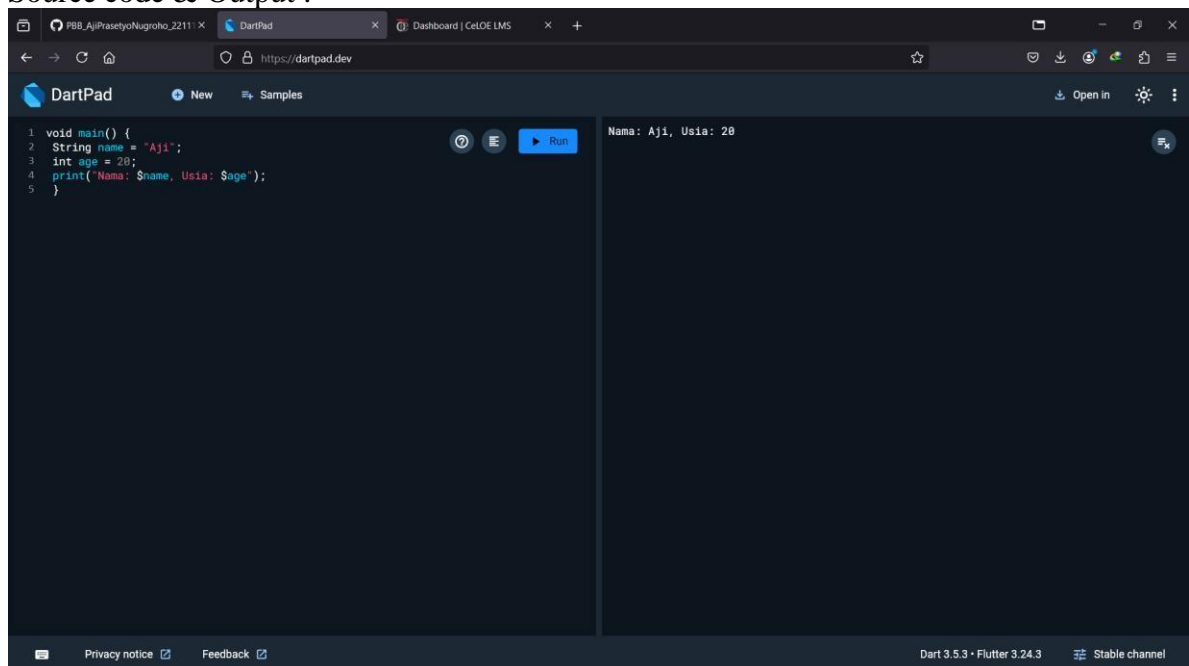


The screenshot shows the DartPad web interface. The left pane contains the following Dart code:

```
1 // variabel
2 void main() {
3   var name = "Aji";
4   var age = 28;
5   print('Nama: $name, Usia: $age');
6 }
```

The right pane displays the output: "Nama: Aji, Usia: 28". The status bar at the bottom indicates "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

Source code & Output :

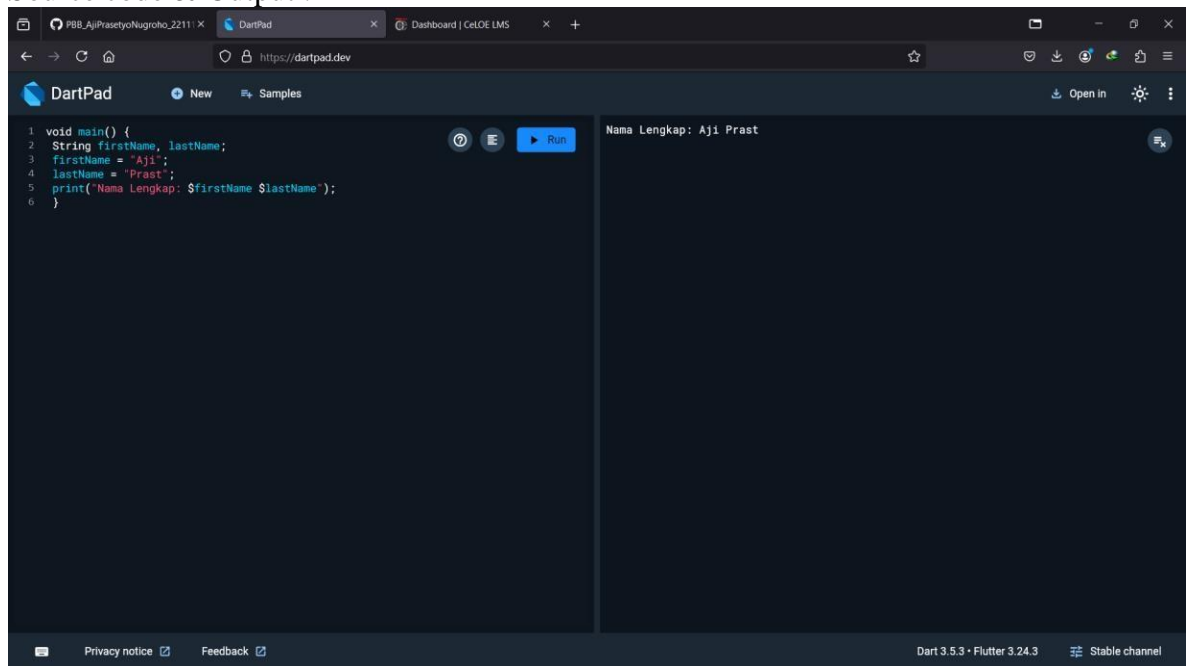


The screenshot shows the DartPad web interface with a different code example. The left pane contains the following Dart code:

```
1 void main() {
2   String name = "Aji";
3   int age = 28;
4   print('Nama: $name, Usia: $age');
5 }
```

The right pane displays the output: "Nama: Aji, Usia: 28". The status bar at the bottom indicates "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

Source code & Output :



The screenshot shows the DartPad web interface. The code editor on the left contains the following Dart code:

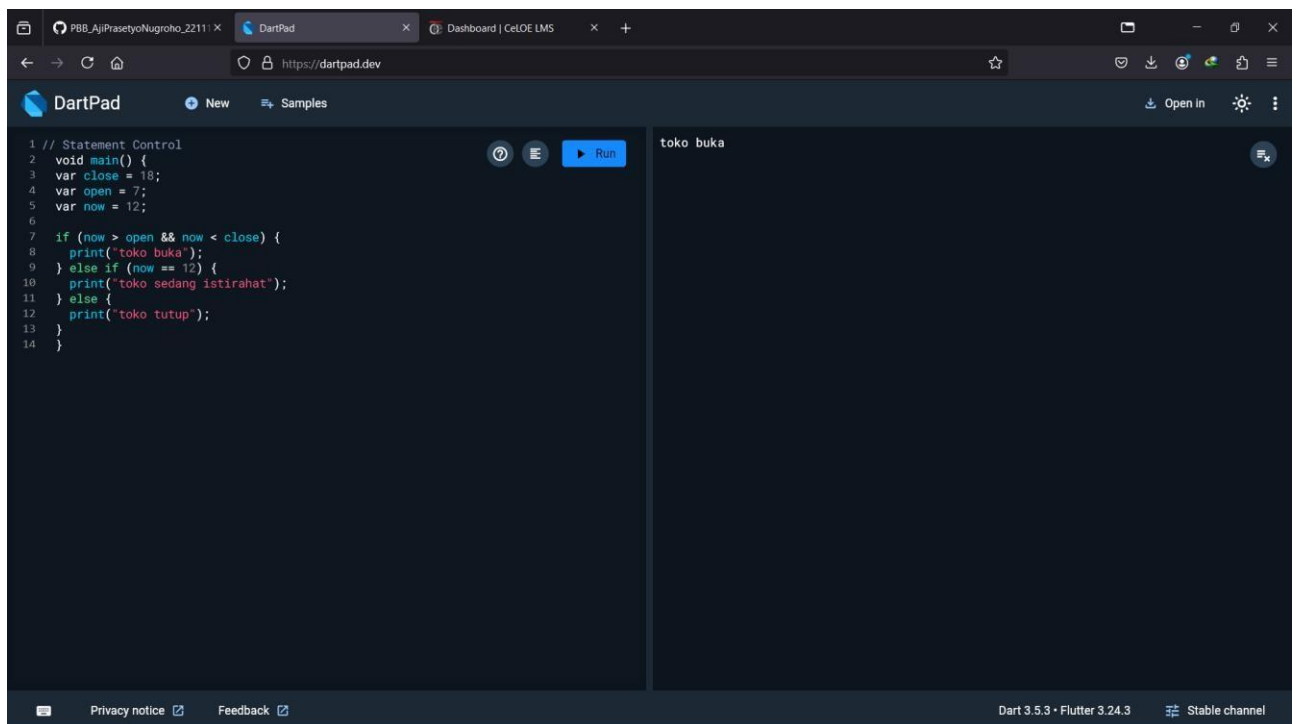
```
1 void main() {  
2   String firstName, lastName;  
3   firstName = "Aji";  
4   lastName = "Prast";  
5   print('Nama Lengkap: $firstName $lastName');  
6 }
```

The output pane on the right displays the result of the program execution: "Nama Lengkap: Aji Prast". The status bar at the bottom indicates "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

B. Statement Control

Statement control atau kontrol alur adalah struktur yang digunakan untuk mengendalikan alur eksekusi program berdasarkan kondisi tertentu.

Source code & Output:

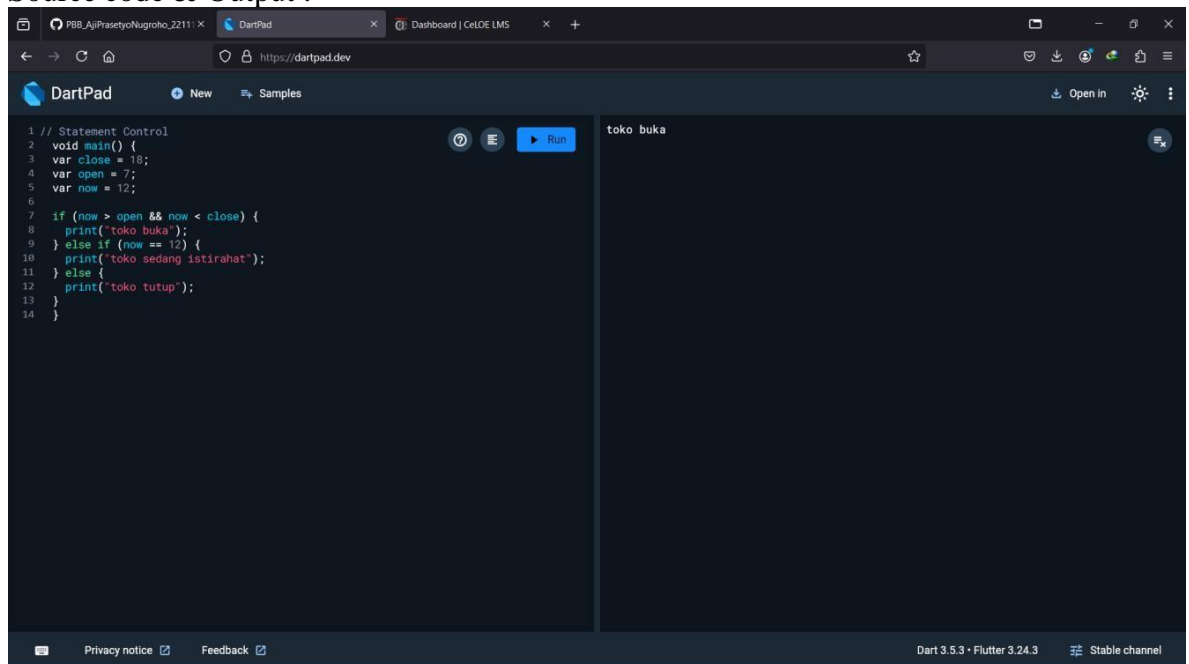


The screenshot shows the DartPad web interface with a Dart program demonstrating statement control. The code editor on the left contains the following Dart code:

```
1 // Statement Control  
2 void main() {  
3   var close = 18;  
4   var open = 7;  
5   var now = 12;  
6  
7   if (now > open && now < close) {  
8     print('toko buka');  
9   } else if (now == 12) {  
10    print('toko sedang istirahat');  
11  } else {  
12    print('toko tutup');  
13  }  
14 }
```

The output pane on the right displays the result of the program execution: "toko buka". The status bar at the bottom indicates "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

Source code & Output :



The screenshot shows the DartPad web interface. The code editor on the left contains the following Dart code:

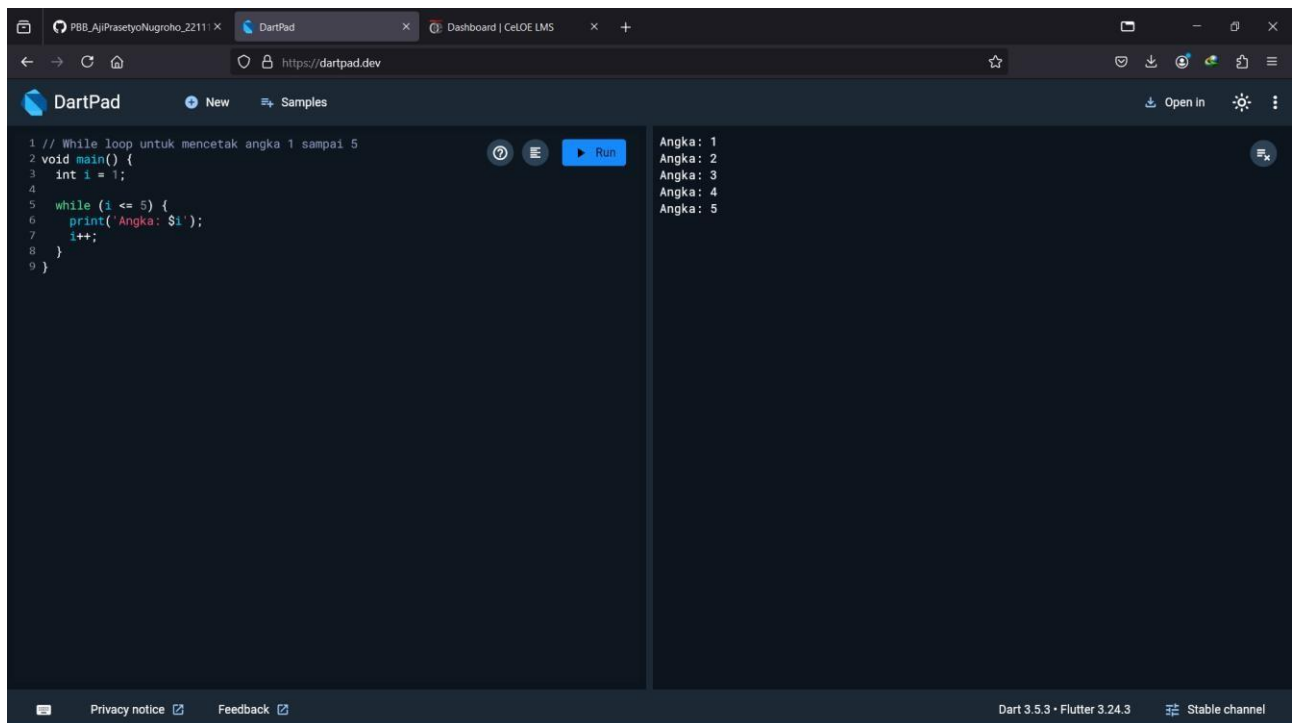
```
1 // Statement Control
2 void main() {
3   var close = 18;
4   var open = 7;
5   var now = 12;
6
7   if (now > open && now < close) {
8     print("toko buka");
9   } else if (now == 12) {
10    print("toko sedang istirahat");
11  } else {
12    print("toko tutup");
13  }
14 }
```

The output panel on the right displays the result of the code execution: "toko buka". The status bar at the bottom indicates "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

C. Looping

Looping digunakan untuk mengeksekusi blok kode berulang kali sampai kondisi tertentu terpenuhi atau berhenti.

Source code & Output :



The screenshot shows the DartPad web interface. The code editor on the left contains the following Dart code:

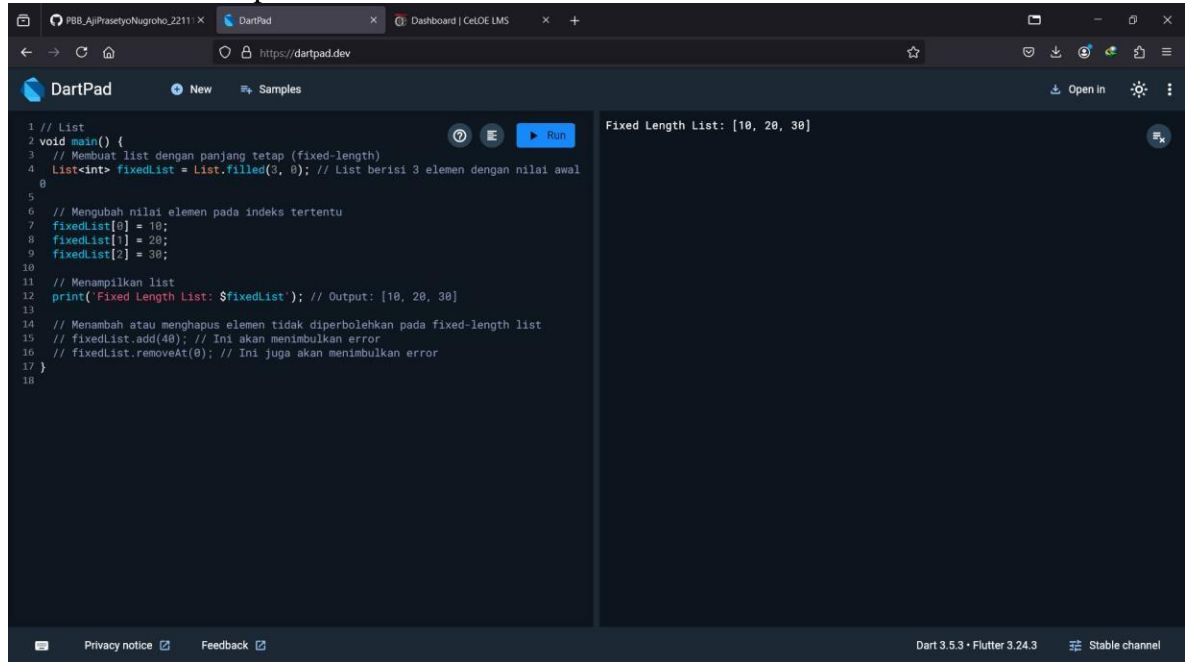
```
1 // While loop untuk mencetak angka 1 sampai 5
2 void main() {
3   int i = 1;
4
5   while (i <= 5) {
6     print('Angka: $i');
7     i++;
8   }
9 }
```

The output panel on the right displays the result of the code execution: "Angka: 1", "Angka: 2", "Angka: 3", "Angka: 4", and "Angka: 5". The status bar at the bottom indicates "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

D. List

List adalah koleksi terurut yang dapat menyimpan elemen-elemen dengan tipe yang sama atau berbeda. List di Dart bisa bersifat dinamis (dapat berubah ukuran) atau fixed-length (panjang tetap).

Source code & Output :



The screenshot shows the DartPad web interface. The code editor on the left contains the following Dart code:

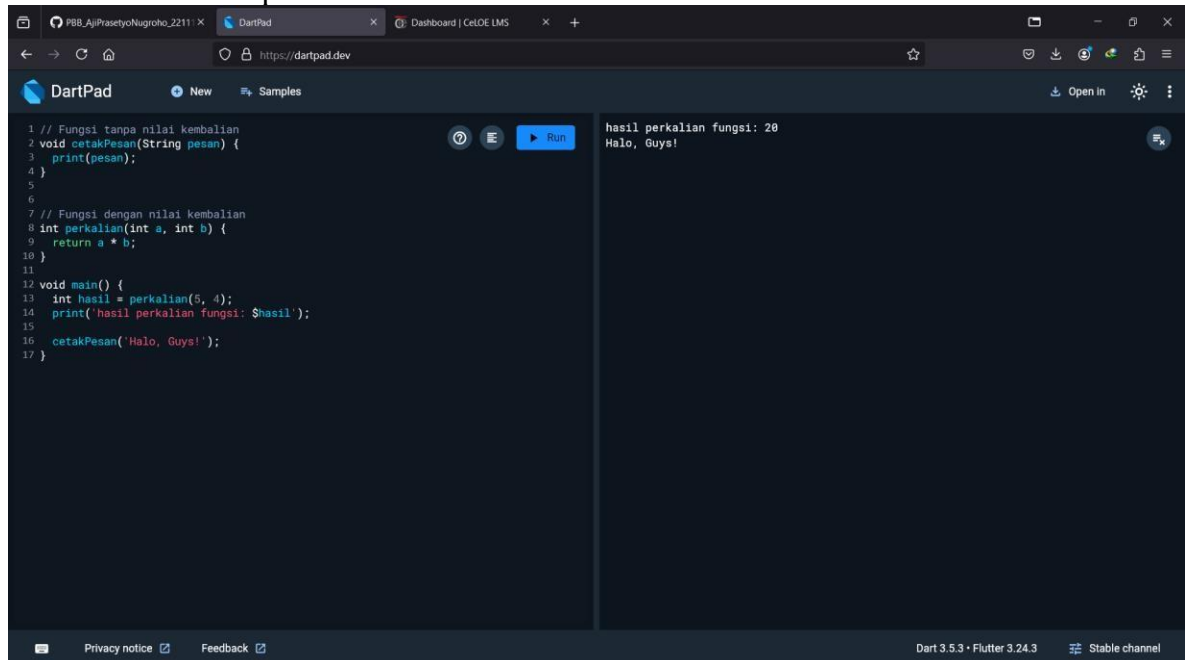
```
1 // List
2 void main() {
3   // Membuat list dengan panjang tetap (fixed-length)
4   List<int> fixedList = List.filled(3, 0); // List berisi 3 elemen dengan nilai awal 0
5
6   // Mengubah nilai elemen pada indeks tertentu
7   fixedList[0] = 10;
8   fixedList[1] = 20;
9   fixedList[2] = 30;
10
11  // Menampilkan list
12  print('Fixed Length List: $fixedList'); // Output: [10, 20, 30]
13
14  // Menambah atau menghapus elemen tidak diperbolehkan pada fixed-length list
15  // fixedList.add(40); // Ini akan menimbulkan error
16  // fixedList.removeAt(0); // Ini juga akan menimbulkan error
17 }
18
```

The output panel on the right displays the result of the code execution: "Fixed Length List: [10, 20, 30]". The bottom status bar indicates "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

E. Fungsi

Fungsi adalah blok kode yang dapat dieksekusi untuk melakukan tugas tertentu. Fungsi membantu untuk memecah kode menjadi bagian-bagian yang lebih kecil dan lebih mudah dikelola.

Source code & Output :



The screenshot shows the DartPad web interface. The code editor on the left contains the following Dart code:

```
1 // Fungsi tanpa nilai kembalian
2 void cetakPesan(String pesan) {
3   print(pesan);
4 }
5
6
7 // Fungsi dengan nilai kembalian
8 int perkalian(int a, int b) {
9   return a * b;
10 }
11
12 void main() {
13   int hasil = perkalian(5, 4);
14   print('hasil perkalian fungsi: $hasil');
15   cetakPesan('Halo, Guys!');
16 }
17
```

The output panel on the right displays the results of the code execution: "hasil perkalian fungsi: 20" and "Halo, Guys!". The bottom status bar indicates "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

UNGUIDED

1. Tugas Percabangan (Branching)

Soal:

Buatlah sebuah fungsi dalam Dart yang menerima sebuah nilai dari user, lalu melakukan percabangan untuk memberikan output berdasarkan kondisi berikut:

Deskripsi :

- Jika nilai **lebih besar dari 70**, program harus mereturn **"Nilai A"**.
- Jika nilai **lebih besar dari 40** tetapi **kurang atau sama dengan 70**, program harus mereturn **"Nilai B"**.
- Jika nilai **lebih besar dari 0** tetapi **kurang atau sama dengan 40**, program harus mereturn **"Nilai C"**.
- Jika nilai tidak memenuhi semua kondisi di atas, program harus mereturn teks kosong.

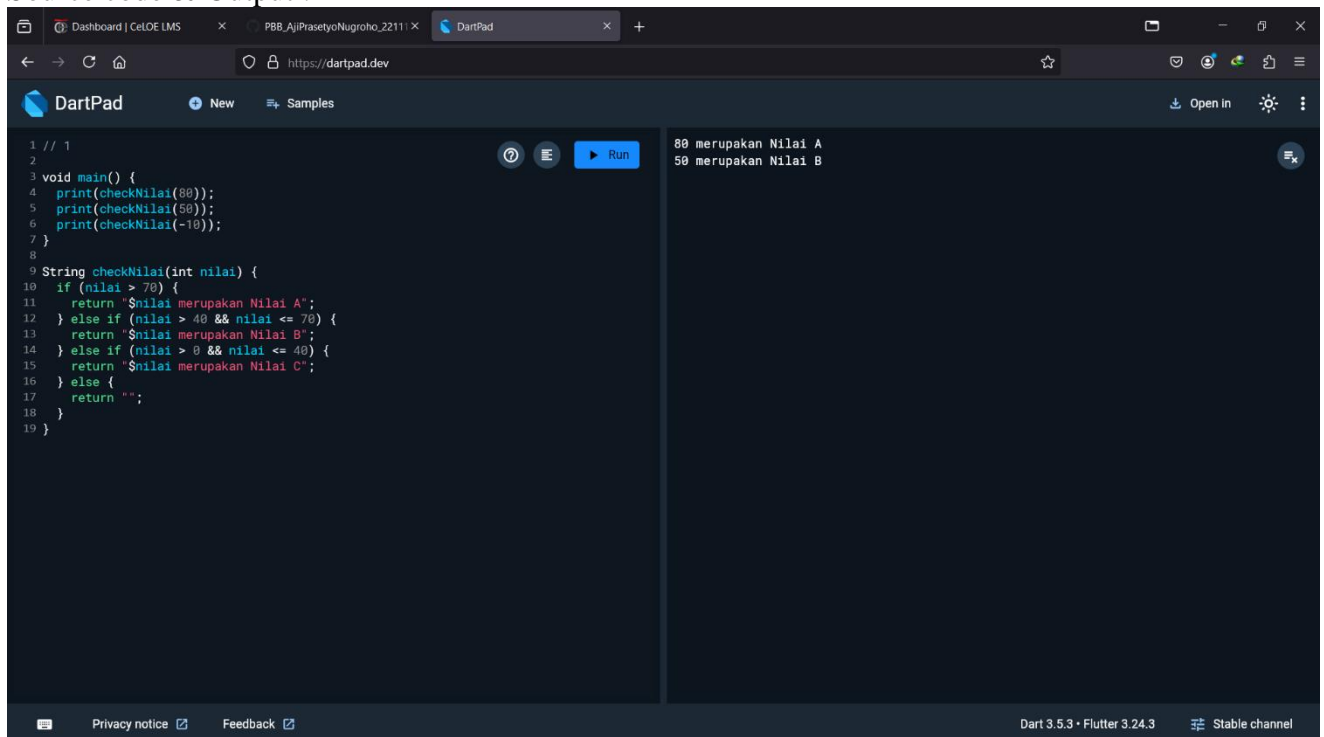
Sampel Input: 80

Sampel Output: 80 merupakan Nilai A

Sampel Input: 5

Sampel Output: 50 merupakan Nilai B

Source code & Output :



The screenshot shows the DartPad web application. The left pane contains the following Dart code:

```
1 // 1
2
3 void main() {
4   print(checkNilai(80));
5   print(checkNilai(50));
6   print(checkNilai(-10));
7 }
8
9 String checkNilai(int nilai) {
10  if (nilai > 70) {
11    return "Nilai merupakan Nilai A";
12  } else if (nilai > 40 && nilai <= 70) {
13    return "Nilai merupakan Nilai B";
14  } else if (nilai > 0 && nilai <= 40) {
15    return "Nilai merupakan Nilai C";
16  } else {
17    return "";
18  }
19 }
```

The right pane shows the output of the code:

```
80 merupakan Nilai A
50 merupakan Nilai B
```

The bottom of the interface shows the footer with "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

Kode tersebut merupakan sebuah program Dart yang mendefinisikan fungsi `checkNilai` untuk mengevaluasi suatu nilai integer dan memberikan output berdasarkan kondisi tertentu. Fungsi `checkNilai` menerima parameter nilai dan mengevaluasinya menggunakan pernyataan `if-else`. Jika nilai lebih besar dari 70, fungsi akan mengembalikan teks dengan format "<nilai> merupakan Nilai A". Jika nilai lebih besar dari 40 tetapi kurang atau sama dengan 70, maka mengembalikan "<nilai> merupakan Nilai B". Jika nilai lebih besar dari 0 tetapi kurang atau sama dengan 40, maka mengembalikan "<nilai> merupakan Nilai C". Jika tidak memenuhi semua kondisi tersebut (nilai 0 atau negatif), fungsi akan mengembalikan string kosong. Pada fungsi `main()`, program memanggil `checkNilai` dengan nilai 80, 50, dan -10, kemudian mencetak hasilnya ke konsol. Output yang dihasilkan adalah "80 merupakan Nilai A", "50 merupakan Nilai B", dan string kosong untuk -10 karena tidak memenuhi kondisi yang ditentukan.

2. Tugas Looping (Perulangan)

Soal:

Buatlah sebuah program dalam Dart yang menampilkan piramida bintang dengan menggunakan `for loop`. Panjang piramida ditentukan oleh input dari user.

Contoh Output:



Source code :

```
// 2
import 'dart:io';

Run | Debug
void main() {
  stdout.write("Masukkan tinggi piramida: ");
  int? tinggi = int.parse(stdin.readLineSync());

  cetakPiramida(tinggi);
}

void cetakPiramida(int tinggi) {
  for (int i = 1; i <= tinggi; i++) {
    for (int j = 1; j <= tinggi - i; j++) {
      stdout.write(" ");
    }

    for (int k = 1; k <= 2 * i - 1; k++) {
      stdout.write("*");
    }

    print("");
  }
}
```

Output :

```
PS D:\Praktikum PPB> dart run PBB_AjiPrasetyoNugroho_2211104049_SE_06_02\03_Pengenalan_Dart\Unguided/main.dart
Masukkan tinggi piramida: 7
  *
 ***
*****
*****
*****
*****
*****
*****
PS D:\Praktikum PPB>
```

Kode tersebut adalah program Dart yang meminta input dari user untuk menentukan tinggi sebuah piramida bintang dan kemudian mencetak piramida tersebut ke konsol. Program dimulai dengan meminta user memasukkan tinggi piramida menggunakan `stdin.readLineSync()` yang kemudian diubah menjadi tipe integer dan disimpan dalam variabel `tinggi`. Fungsi `cetakPiramida` kemudian dipanggil dengan parameter `tinggi` yang diberikan. Pada fungsi `cetakPiramida`, terdapat dua `for` loop bersarang: loop pertama mencetak spasi untuk membentuk segitiga, sedangkan loop kedua mencetak karakter bintang (*) yang jumlahnya bertambah di setiap baris. Setiap kali selesai mencetak bintang di satu baris, `print("")` digunakan untuk pindah ke baris berikutnya. Output yang dihasilkan adalah sebuah piramida bintang dengan tinggi yang sesuai input dari user, dengan karakter bintang yang membentuk pola segitiga simetris.

3. Tugas Function

Soal:

Buatlah program Dart yang meminta input berupa sebuah bilangan bulat dari user, kemudian program akan mengecek apakah bilangan tersebut merupakan bilangan prima atau bukan.

Sampel Input: 23

Sampel Output: bilangan prima

Sampel Input: 12

Sampel Output: bukan bilangan prima

Source code :

```
44 // 3
45 import 'dart:io';
46
47 Run | Debug
48 void main() {
49     stdout.write("Masukkan sebuah bilangan bulat: ");
50     int? angka = int.parse(stdin.readLineSync());
51
52     if (isPrima(angka)) {
53         print("bilangan prima");
54     } else {
55         print("bukan bilangan prima");
56     }
57
58 bool isPrima(int n) {
59     if (n <= 1) {
60         return false;
61     }
62     for (int i = 2; i <= n ~/ 2; i++) {
63         if (n % i == 0) {
64             return false;
65         }
66     }
67     return true;
68 }
```

Output :

```
PS D:\Praktikum PPB> dart run PBB_AjiPrasetyoNugroho_2211104049_SE_06_02\03_Pengenalan_Dart\Unguided/main.dart
Masukkan sebuah bilangan bulat: 23
bilangan prima
PS D:\Praktikum PPB> dart run PBB_AjiPrasetyoNugroho_2211104049_SE_06_02\03_Pengenalan_Dart\Unguided/main.dart
Masukkan sebuah bilangan bulat: 12
bukan bilangan prima
PS D:\Praktikum PPB> 
```

Kode tersebut adalah program Dart yang menerima input berupa bilangan bulat dari user dan mengecek apakah bilangan tersebut merupakan bilangan prima atau bukan. Program dimulai dengan meminta user memasukkan sebuah bilangan bulat menggunakan ``stdin.readLineSync()``, kemudian mengubah input tersebut menjadi tipe integer dan menyimpannya dalam variabel ``angka``. Setelah itu, program memanggil fungsi ``isPrima`` yang berfungsi untuk menentukan apakah ``angka`` adalah bilangan prima. Fungsi ``isPrima`` akan mengembalikan ``false`` jika bilangan kurang dari atau sama dengan 1, karena bilangan prima harus lebih dari 1. Jika ``n`` lebih dari 1, fungsi akan memeriksa apakah ``n`` habis dibagi oleh bilangan lain dalam rentang dari 2 hingga setengah dari ``n`` (``n ~/ 2``). Jika ``n`` habis dibagi oleh bilangan lain, maka ``n`` bukan bilangan prima dan fungsi mengembalikan ``false``. Jika ``n`` tidak habis dibagi oleh bilangan manapun di rentang tersebut, fungsi mengembalikan ``true``, menandakan bahwa ``angka`` adalah bilangan prima. Program utama kemudian mencetak "bilangan prima" jika fungsi mengembalikan ``true``, atau "bukan bilangan prima" jika ``false``.