

**LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL 3
PENGENALAN DART**



Disusun Oleh :

Zivana Afra Yulianto / 2211104039

S1SE-06-02

Asisten Praktikum :

Muhammad Faza Zulian Gesit Al Barru

Aisyah Hasna Aulia

Dosen Pengampu :

Yudha Islami Sulistya, S.Kom., M.Cs

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO 2024**

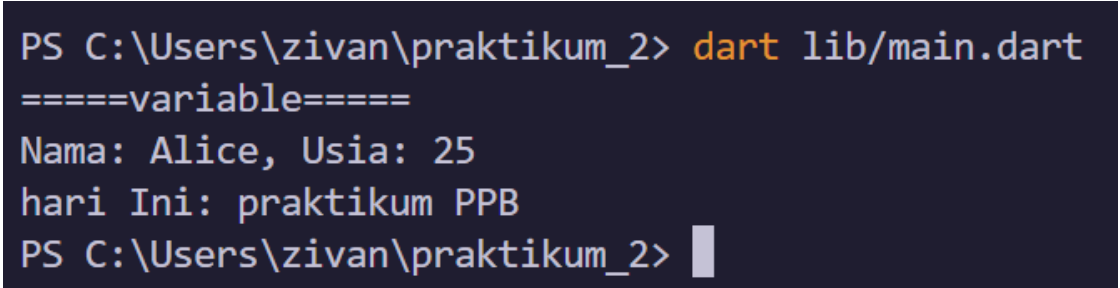
A. GUIDED

1. Variable

Code :

```
void main() {  
  
    print('==== variable====');  
  
    var name = "Alice"; // Tipe data String  
  
    var age = 25; // Tipe data Integer  
  
    print("Nama: $name, Usia: $age");  
  
  
    String firstName = 'praktikum';  
  
    var lastName = 'PPB';  
  
    print("hari Ini: $firstName $lastName");  
}
```

Running :



```
PS C:\Users\zivan\praktikum_2> dart lib/main.dart  
====variable====  
Nama: Alice, Usia: 25  
hari Ini: praktikum PPB  
PS C:\Users\zivan\praktikum_2> |
```

Deskripsi :

Variabel pada **Dart** adalah tempat penyimpanan data yang memiliki nama dan tipe tertentu, yang dapat dideklarasikan menggunakan `var`, `final`, `const`, atau tipe data eksplisit seperti `int` dan `String`. Variabel dengan `var` akan otomatis memiliki tipe sesuai nilai yang diberikan, sedangkan `final` dan `const` digunakan untuk nilai yang tidak bisa diubah setelah inisialisasi. Dart juga mendukung **null safety**, yang memastikan variabel tidak dapat bernilai `null` kecuali secara eksplisit diizinkan dengan tanda `?`.

2. If-Else

Code :

```
void main() {  
  
    print('====if else====');  
  
    var open = 8;  
  
    var close = 17;  
  
    var now = 20;  
  
    if (now >= open && now < close) {  
  
        print("toko buka");  
  
    } else if (now == 12) {  
  
        print("toko istirahat");  
  
    } else {  
  
        print("toko tutup");  
  
    }  
  
    //condition ? true : false  
  
    print('====if else 1 line====');  
  
    var toko = now > open ? "toko tutup" : "toko tutup";  
  
    print(toko);  
  
}
```

Running :

```
PS C:\Users\zivan\praktikum_2> dart lib/main.dart  
====if else====  
toko tutup  
====if else 1 line====  
toko tutup  
PS C:\Users\zivan\praktikum_2> █
```

Deskripsi :

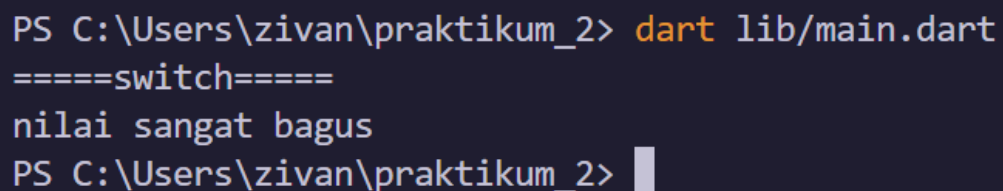
Pada **Dart**, struktur **if-else** digunakan untuk melakukan percabangan logika berdasarkan kondisi tertentu. Jika kondisi di dalam pernyataan `if` bernilai **true**, blok kode di dalamnya akan dieksekusi. Jika tidak, blok `else` akan dijalankan. Selain itu, kita dapat menambahkan beberapa kondisi dengan **else if** untuk memeriksa lebih dari satu kondisi.

3. Switch

Code :

```
void main() {  
    print('====switch====');  
    var nilai = 'a';  
    switch (nilai) {  
        case 'a':  
            print('nilai sangat bagus');  
            break;  
        case 'b':  
            print('nilai baik');  
            break;  
        case 'c':  
            print('nilai cukup');  
            break;  
        default:  
            print('nilai tidak tersedia');  
    }  
}
```

Running :



```
PS C:\Users\zivan\praktikum_2> dart lib/main.dart  
====switch====  
nilai sangat bagus  
PS C:\Users\zivan\praktikum_2>
```

Deskripsi :

Pada **Dart**, struktur **switch** digunakan untuk memilih salah satu dari beberapa blok kode yang akan dijalankan berdasarkan nilai dari sebuah ekspresi. Setiap blok diawali dengan **case** yang mencocokkan nilai tertentu, dan jika cocok, kode di dalam case tersebut dieksekusi. Untuk menangani nilai yang tidak sesuai dengan case manapun, digunakan **default**. Setiap case biasanya diakhiri dengan **break** untuk mencegah eksekusi berlanjut ke case berikutnya.

4. Looping

Code :

```
void main() {  
  
    print('====looping====');  
  
    for (int i = 0; i <= 10; i++) {  
  
        print(i);  
  
    }  
  
    int i = 1;  
  
    while (i <= 5) {  
  
        print('angka: $i');  
  
        i++;  
  
    }  
  
}
```

Running :

```
PS C:\Users\zivan\praktikum_2> dart lib/main.dart  
====looping====  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
angka: 1  
angka: 2  
angka: 3  
angka: 4  
angka: 5  
PS C:\Users\zivan\praktikum_2> |
```

Deskripsi :

Looping pada Dart adalah proses pengulangan suatu blok kode secara berulang hingga kondisi tertentu terpenuhi. Dart menyediakan beberapa jenis loop, seperti **for**, **while**, dan **do-while**. Loop **for** biasanya digunakan ketika jumlah pengulangan diketahui, sedangkan **while** dan **do-while** digunakan untuk pengulangan berdasarkan kondisi.

5. List Fixed dan Growable

Code :

```
void main() {  
  
  print('====list fixed====');  
  
  List<int> fixedList = List.filled(3, 0);  
  
  fixedList[0] = 10;  
  fixedList[1] = 20;  
  fixedList[2] = 30;  
  print('fixed list : $fixedList');  
  
  //list growable  
  print('====list growable====');  
  
  List<int> growableList = [];  
  growableList.add(10);  
  growableList.add(20);  
  growableList.add(30);  
  print(growableList);  
  growableList.add(40);  
  growableList.add(50);  
  growableList.add(60);  
  print(growableList);  
  
  growableList.remove(20);  
  print(growableList);  
  
}
```

Running :

```
PS C:\Users\zivan\praktikum_2> dart lib/main.dart  
====list fixed====  
fixed list : [10, 20, 30]  
====list growable====  
[10, 20, 30]  
[10, 20, 30, 40, 50, 60]  
[10, 30, 40, 50, 60]  
PS C:\Users\zivan\praktikum_2> █
```

Deskripsi :

Pada **Dart**, terdapat dua jenis **List**: **fixed-length list** dan **growable list**. **Fixed-length list** memiliki ukuran tetap, artinya jumlah elemen di dalamnya tidak dapat diubah setelah dibuat, meskipun nilai elemennya bisa diganti. Contoh deklarasinya adalah `List<int>.filled(3, 0)`, yang membuat list dengan panjang tetap 3. Sementara itu, **growable list** dapat berubah ukurannya secara dinamis, memungkinkan elemen-elemen baru ditambahkan atau dihapus. List growable biasanya dibuat menggunakan `[]` atau dengan `List<int>()`, dan cocok untuk situasi di mana jumlah data tidak diketahui secara pasti saat deklarasi.

6. Fungsi

Code :

```
void main() {  
    void cetakPesan(String pesan) {  
        print(pesan);  
    }  
    int perkalian(int a, int b) {  
        return a * b;  
    }  
    int hasil = perkalian(5, 4);  
    print('hasil perkalian fungsi : $hasil');  
    cetakPesan('halo guys');  
}
```

Running :

```
PS C:\Users\zivan\praktikum_2> dart lib/main.dart  
hasil perkalian fungsi : 20  
halo guys  
PS C:\Users\zivan\praktikum_2> █
```

Deskripsi :

Fungsi pada **Dart** adalah blok kode yang dapat digunakan kembali untuk menjalankan tugas tertentu. Fungsi dideklarasikan dengan kata kunci **return type**, nama fungsi, parameter opsional, dan isi fungsi. Fungsi dapat mengembalikan nilai menggunakan `return`, atau hanya menjalankan perintah tanpa mengembalikan apapun (menggunakan `void`).

B. UNGUIDED

1. Soal nomor 1

Code :

```
import 'dart:io';

//nomor 1

String cekNilai(int nilai) {

  if (nilai > 70) {

    return "Nilai A";

  } else if (nilai > 40 && nilai <= 70) {

    return "Nilai B";

  } else if (nilai > 0 && nilai <= 40) {

    return "Nilai C";

  } else {

    return "";

  }

}

void main() {

  stdout.write("Masukkan nilai: ");

  int nilai = int.parse(stdin.readLineSync());

  String hasil = cekNilai(nilai);

  if (hasil.isNotEmpty) {

    print("$nilai merupakan $hasil");

  } else {

    print("Nilai tidak valid");

  }

}
```

Running :

```
PS C:\PPB_ZivanaAfraYulianto_2211104039_SE0602> dart 03_Pengenalan_Dart/Unguided/1.dart
Masukkan nilai: 80
80 merupakan Nilai A
PS C:\PPB_ZivanaAfraYulianto_2211104039_SE0602> █
```

Deskripsi :

Program di atas menerima input nilai dari pengguna, kemudian mengevaluasi nilai tersebut menggunakan fungsi `cekNilai`. Fungsi ini memeriksa tiga kondisi: jika nilai lebih dari 70, mengembalikan "Nilai A"; jika nilai antara 41 hingga 70, mengembalikan "Nilai B"; dan jika nilai antara 1 hingga 40, mengembalikan "Nilai C". Jika nilai di luar rentang ini, fungsi mengembalikan string kosong. Di dalam fungsi `main`, input nilai dikonversi menjadi tipe integer dan hasil dari fungsi `cekNilai` diperiksa. Jika hasil tidak kosong, program mencetak nilai dan kategori nilai, sedangkan jika kosong, program mengeluarkan pesan bahwa nilai tidak valid.

2. Soal nomor 2

Code :

```
import 'dart:io';

void main() {

  stdout.write("Masukkan panjang piramida: ");

  int panjang = int.parse(stdin.readLineSync()!);

  for (int i = 1; i <= panjang; i++) {

    // Menampilkan spasi

    for (int j = 1; j <= panjang - i; j++) {

      stdout.write(" ");

    }

    // Menampilkan bintang

    for (int k = 1; k <= 2 * i - 1; k++) {

      stdout.write("*");

    }

    print(""); // Pindah ke baris berikutnya

  }

}
```

Running :

```
PS C:\PPB_ZivanaAfraYulianto_2211104039_SE0602> dart 03_Pengenalan_Dart/Unguided/2.dart
Masukkan panjang piramida: 4
*
***
*****
PS C:\PPB_ZivanaAfraYulianto_2211104039_SE0602> █
```

Deskripsi :

Program di atas meminta pengguna untuk memasukkan panjang piramida, lalu mencetak piramida bintang dengan jumlah baris sesuai input. Pada setiap baris, dua loop digunakan: loop pertama mencetak spasi untuk meratakan piramida, dan loop kedua

mencetak simbol bintang. Jumlah spasi berkurang seiring bertambahnya baris, sedangkan jumlah bintang bertambah secara berurutan sesuai dengan pola piramida. Setelah setiap baris selesai, program pindah ke baris berikutnya menggunakan `print("")`. Hasil akhirnya adalah piramida simetris yang terbentuk dari simbol bintang.

3. Soal nomor 3

Code :

```
import 'dart:io';

bool cekBilanganPrima(int n) {

  if (n <= 1) {

    return false; // Bilangan <= 1 bukan bilangan prima

  }

  for (int i = 2; i <= n ~/ 2; i++) {

    if (n % i == 0) {

      return false; // Jika ada faktor selain 1 dan n, maka bukan bilangan prima

    }

  }

  return true;

}

void main() {

  stdout.write("Masukkan bilangan bulat: ");

  int bilangan = int.parse(stdin.readLineSync()!);

  if (cekBilanganPrima(bilangan)) {

    print("$bilangan merupakan bilangan prima.");

  } else {

    print("$bilangan bukan bilangan prima.");

  }

}
```

Running :

```
PS C:\PPB_ZivanaAfraYulianto_2211104039_SE0602> dart 03_Pengenalan_Dart/Unguided/3.dart
Masukkan bilangan bulat: 7
7 merupakan bilangan prima.
PS C:\PPB_ZivanaAfraYulianto_2211104039_SE0602> █
```

Deskripsi :

Program di atas memeriksa apakah suatu bilangan bulat yang dimasukkan oleh pengguna merupakan bilangan prima atau bukan. Fungsi `cekBilanganPrima` menerima

bilangan sebagai argumen dan memeriksa apakah bilangan tersebut memiliki faktor selain 1 dan bilangan itu sendiri. Jika bilangan kurang dari atau sama dengan 1, fungsi langsung mengembalikan **false**. Loop digunakan untuk memeriksa faktor bilangan dari 2 hingga setengah dari bilangan tersebut. Jika ditemukan faktor, fungsi mengembalikan **false** (bukan prima); jika tidak, mengembalikan **true** (prima). Hasil pemeriksaan kemudian ditampilkan kepada pengguna melalui fungsi **main**.