

TUGAS GUIDED
Praktikum Pemrograman Perangkat Bergerak

MODUL 3
03_Pengenalan_Dart



Disusun Oleh :
Arwin Nabel Arroffif
(2211104057)

Asisten Praktikum :
Faza: Jarvishsoft
Hasna: asyhasnaa

Dosen Pengampu :
Yudha Islami Sulistya

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2024

PENGENALAN DART

Tujuan Praktikum

Mahasiswa mampu memahami konsep layout pada Flutter

Mahasiswa dapat mengimplementasikan desain user interface pada Flutter

Apa itu Dart?

Dart adalah bahasa pemrograman yang open source dan general purpose. Dart dikembangkan oleh Google dan ditujukan untuk membuat aplikasi multiplatform seperti mobile, desktop, dan web. Dart merupakan bahasa modern dan berfitur lengkap. Dart juga memiliki banyak kemiripan dengan bahasa lain yang sudah banyak dikenal seperti Java, C#, Javascript, Swift, dan Kotlin.

Contoh kode:

```
void main() {  
  print('Praktikum Perangkat Bergerak');  
}
```

Dart memiliki karakteristik, yakni :

- *Statically typed,*

```
var name = 'Praktikum PPB';
```

```
String language = 'Dart';
```

-

Type inference,

```
print('Hello $name. Welcome to $language!');
```

-

String interpolation,

```
print('Hello $name. Welcome to $language!');
```

- *Multi-paradigm*

1. Mendukung paradigma Object-Oriented Programming (OOP), dapat membuat program terstruktur yang memiliki konsep object, classes, properties, constructor, dll.
2. Functional Programming (FP), dapat menerapkan fitur-fitur seperti: higherorder function, recursion, first-class citizen, dll.

A. Variable

```
Run | Debug | Profile
void main() {
  var name = 'Arwin';
  var umur = 21;
  var namalengkap = 'Arwin Nabiel Arrofif';
  var nilai = 80;
  int i = 1;
  List<int> fixedList = List.filled(3, 0);
  String firstname = 'Arwin';
  var lastname = 'Nabiel Arrofif';

  // MENGGABUNGKAN MENGGUNAKAN CAMEL CASE
  print('hai $firstname $lastname');
}
```

B. Statement Control

• IF-ELSE Statement

Untuk memproses suatu kondisi dalam program, kita dapat menggunakan ekspresi if. Jika hasil dari ekspresi tersebut bernilai **true (benar)**, maka blok kode di dalam **if** akan dijalankan. Namun, jika kondisi tersebut tidak terpenuhi atau bernilai **false (salah)**, kita dapat menggunakan ekspresi **else** untuk mengeksekusi blok kode alternatif.

```
// ELSE STATEMENT
if (umur > 21) {
    print('$firstname MASIH MUDA PRODUKTIF $umur');
} else if (umur == 15) {
    print('$firstname GERING URIP $umur');
} else {
    print('$firstname MASIH MUDA PUYENG $umur');
}
```

• Switch-Case Statement

Switch-case digunakan untuk memilih salah satu dari banyak blok kode untuk dieksekusi berdasarkan nilai suatu ekspresi. Dengan switch-case, program dapat memeriksa variabel dan menjalankan kode yang sesuai dengan nilai yang diberikan. Jika tidak ada case yang cocok, blok default dapat digunakan untuk menangani nilai yang tidak terduga.

Contoh Kode

```
// SWITCH CASE
switch (nilai) {
  case 'a':
    print('nilai gue');
    break;
  case 'b':
    print('nilai cukup');
    break;
  case 'c':
    print('nilai baik');
    break;
  default:
    print('nilai alhamdulillah');
}
```

C. Looping

Secara umum, terdapat dua cara untuk melakukan *looping* di dart, yaitu menggunakan *for loop* dan *while loop*.

- For Loop

For loop digunakan untuk mengulangi suatu blok kode dengan jumlah perulangan yang sudah ditentukan.

- While Loops

While loop digunakan ketika jumlah iterasi belum pasti, dan perulangan akan berlangsung selama kondisi yang diberikan bernilai true.

```

// While loop untuk mencetak angka 1 sampai 5
while (i <= 5) {
    print('Angka: $i');
    i++; // Increment untuk menghindari loop tak berujung
}

// Mengubah elemen dalam list
fixedList[0] = 50;
💡fixedList[1] = 60;
fixedList[2] = 70;
print('Fixed Length List: $fixedList');
}

```

D. List

List adalah tipe data yang digunakan untuk merepresentasikan koleksi nilai yang terurut. List dapat berisi elemen-elemen dengan tipe data yang sama atau berbeda. List dapat diinisialisasi dengan nilai-nilai yang diberikan, atau dapat dibuat kosong dan kemudian diisi kemudian.

- **Fixed Length List**

- Tidak dapat diubah ukurannya setelah dibuat.
- Elemen di dalamnya bisa diubah, tetapi kita tidak bisa menambah atau menghapus elemen setelah list tersebut diinisialisasi dengan ukuran tetap.

Note : Elemen pada indeks tertentu bisa diubah, tetapi kita

tidak bisa menambah atau menghapus elemen setelah ukuran list ditetapkan.

- **Growable List**

Gunakan growable list apabila memiliki banyak object yang tidak menentu atau banyaknya object yang terus bertambah.

```
// Mengubah elemen dalam list
fixedList[0] = 50;
fixedList[1] = 60;
fixedList[2] = 70;
print('Fixed Length List: $fixedList');

List<int> growableList = [];
// Menambahkan elemen baru ke dalam list
growableList.add(50);
growableList.add(60);
growableList.add(70);
print('Growable List setelah menambah elemen: $growableList');

// Menambahkan lebih banyak elemen
growableList.add(80);
💡growableList.add(90);
print(growableList); // Output: [50, 60, 70, 80, 90]

// Menghapus elemen dari list
growableList.remove(20);
print('Growable List setelah menghapus elemen: $growableList');
```

Note: **Dapat diubah ukurannya** secara dinamis, kita bisa menambah atau menghapus elemen kapan saja.

E. Fungsi

Fungsi adalah blok kode yang dapat dijalankan untuk melakukan tugas tertentu dan, jika diperlukan, dapat mengembalikan nilai. Fungsi dalam Dart juga dapat menerima parameter atau argumen sebagai input. Pada bahasa pemrograman yang mendukung Object Oriented Programming, fungsi atau prosedur memiliki peranan yang sangat penting. Untuk menghasilkan kualitas kode yang sangat baik, programmer bisa menggunakan beberapa prinsip pemrograman yang umum digunakan seperti SOLID,

KISS, YAGNI, dsb. Semua prinsip tersebut menjunjung tinggi *separation of concern* yang artinya setiap kodingan memiliki tanggung jawabnya sendiri dan mengurangi sebanyak mungkin boilerplate code.

Tambahkan return apabila anda mendefinisikan sebuah fungsi, contohnya ada pada codingan diatas pada fungsi diatas, number merupakan parameter.

Variable diluar fungsi yang dibuat agar dapat digunakan didalam fungsi.

Menambahkan Parameter

Parameter adalah variabel yang didefinisikan dalam tanda kurung saat mendeklarasikan fungsi, dan digunakan untuk menerima nilai atau data yang dilewatkan ke fungsi ketika fungsi tersebut dipanggil. Parameter memungkinkan sebuah fungsi menjadi lebih dinamis, karena fungsinya dapat menerima input yang berbeda setiap kali dipanggil.

Fungsi memiliki scope yang terbatas, tentunya fungsi butuh input dari luar agar program didalamnya bisa memproses tugasnya.

Pada fungsi diatas, number merupakan parameter. Variable diluar fungsi yang dibuat agar dapat digunakan didalam fungsi.

```
// Definisi Fungsi: Fungsi sederhana yang mendefinisikan parameter 'number'
int kaliDua(int number) {
    return number * 2;
}

Run | Debug | Profile
void main() {
    int hasil = kaliDua(5);
    print("Hasil perkalian adalah: $hasil");
}
```


