

MODUL 1

BAHASA PEMROGRAMAN DART (1)

Pengertian Dart

Dart adalah bahasa pemrograman berorientasi objek open source yang dikembangkan oleh Google. Dart diperkenalkan pada tahun 2011 dan sejak itu menjadi pilihan populer untuk membangun aplikasi berkinerja tinggi. Dart menggabungkan fitur terbaik dari bahasa seperti Java, JavaScript, dan C#.

Hello World

Pada bahasa pemrograman Dart, fungsi utama yang akan pertama kali dijalankan adalah fungsi `main()`. Berikut ini adalah contoh kode Dart untuk menampilkan `Hello World`.

```
1 main() {  
2   print("Hello World");  
3 }
```

Klik tombol **Run** untuk menjalankan kode diatas. Jika menggunakan VS Code, buka terminal dan ketikkan dart `hello.dart`.

Pada baris 3 program, kita menggunakan kode **print** untuk mencetak teks Hello World. `print()` digunakan untuk menampilkan output kode di terminal. Perlu diingat bahwa akhir setiap pernyataan pada Dart harus diakhiri dengan titik koma (;).

Data Types and Variables

Object

Dart adalah bahasa pemrograman berorientasi objek, artinya semua yang ada di Dart merupakan sebuah objek. Objek dalam bahasa pemrograman mempunyai karakteristik yang disebut **properti** dan perilaku yang disebut **metode**.

Variables

Struktur penulisan variabel di Dart adalah sebagai berikut

```
dataType variableName = Initial Value
```

Konvensi penamaan variabel di Dart adalah camel case. Artinya kita menggunakan huruf besar pada huruf pertama setiap kata kecuali kata pertama, dan tidak menggunakan pemisah, misalnya lowerCamelCase.

Berikut ini contoh variabel.

```

1 main() {
2   var pesan = "Halo Dunia";
3   print(pesan);
4
5   String nama = "Budi";
6   int umur = 20;
7   double tinggiBadan = 170.5;
8   bool mahasiswa = true;
9   print("Nama saya $nama");
10  print("Umur saya $umur tahun");
11  print("Tinggi saya $tinggiBadan cm");
12  print("Seorang mahasiswa: $mahasiswa");
13 }

```

Data Types

Tipe data yang didukung oleh Dart dapat dilihat pada tabel berikut ini

Data Type	Keyword	Description
Numbers	int, double, num	It represents numeric values
Strings	String	It represents a sequence of characters
Booleans	bool	It represents Boolean values true and false
Lists	List	It is an ordered group of items
Maps	Map	It represents a set of values as key-value pairs
Sets	Set	It is an unordered list of unique values of same types
Runes	runes	It represents Unicode values of String
Null	null	It represents null value

Selain menggunakan tipe data diatas, kita juga bisa menggunakan tipe data var. Dart akan menentukan tipe data yang sesuai dengan initial value dari variabel tersebut.

```

1 ▾ main() {
2     var nama = "Budi";
3     var umur = 20;
4     var tinggi = 170.5;
5
6     print(nama.runtimeType);
7     print(umur.runtimeType);
8     print(tinggi.runtimeType);
9 }

```

Jalankan kode berikut ini, kenapa terjadi error?

```

1 ▾ main() {
2     var nama = "Budi";
3     print(nama);
4     nama = 2;
5 }

```

Type Conversion

Kita dapat melakukan konversi dari satu tipe data ke tipe data lainnya.

```

1 ▾ main() {
2     String nilaiString = "1";
3     print("String: $nilaiString");
4
5     // konversi String ke int
6     int nilaiInteger = int.parse(nilaiString);
7     //konversi String ke double
8     double nilaiDouble = double.parse(nilaiString);
9     print("String to int: $nilaiInteger");
10    print("String to double: $nilaiDouble");
11
12    // konversi int ke String
13    nilaiString = nilaiInteger.toString();
14    print("int to String: $nilaiString");
15    // konversi double ke String
16    nilaiString = nilaiDouble.toString();
17    print("double to String: $nilaiString");
18
19    //konversi int ke double
20    print("int to double: ${nilaiInteger.toDouble()}");
21    //konversi double ke int
22    print("double to int ${nilaiDouble.toInt()}");
23 }
24

```

Kenapa kode dibawah ini error? perbaiki kode agar dapat mengkonversi String ke int tanpa mengubah nilai dari variabel nilaiString.

```
1 ▾ main() {  
2   String nilaiString = "1.0";  
3   int nilaiInteger = int.parse(nilaiString);  
4   print(nilaiInteger);  
5 }
```

Operators

Jenis operator pada Dart :

- Arithmetic Operators
- Increment and Decrement Operators
- Assignment Operators
- Relational and Logical Operators
- Type Test Operators

Arithmetic Operators

Operator aritmatika digunakan untuk operasi aritmatika seperti penambahan dan pengurangan. Dibawah ini adalah daftar operator aritmatika pada Dart.

Operator Symbol	Operator Name	Description
+	Addition	For adding two operands
-	Subtraction	For subtracting two operands
-expr	Unary Minus	For reversing the sign of the expression
*	Multiplication	For multiplying two operands
/	Division	For dividing two operands and give output in double
~/	Integer Division	For dividing two operands and give output in integer
%	Modulus	Remainder After Integer Division

```

1 ▾ main() {
2   var operand1 = 10;
3   var operand2 = 7;
4
5   print(operand1 + operand2);
6   print(operand1 - operand2);
7   print(-operand1);
8   print(operand1 * operand2);
9   print(operand1 / operand2);
10  print(operand1 ~/ operand2);
11  print(operand1 % operand2);
12 }

```

Increment and Decrement Operators

Digunakan untuk menambah atau mengurangi variabel.

Operator Symbol	Operator Name	Description
++var	Pre Increment	Increase Value By 1. var = var + 1 Expression value is var+1
--var	Pre Decrement	Decrease Value By 1. var = var - 1 Expression value is var-1
var++	Post Increment	Increase Value By 1. var = var + 1 Expression value is var
var--	Post Decrement	Decrease Value By 1. var = var - 1 Expression value is var

```

1 ▾ main() {
2   int num1 = 0;
3   int num2 = 0;
4
5   num2 = ++num1;
6   print("The value of num2 is $num2");
7
8   num1 = 0;
9   num2 = 0;
10
11  num2 = num1++;
12  print("The value of num2 is $num2");
13 }

```

Apa perbedaan operator ++ didepan dan dibelakang variable?

Assignment Operators

Digunakan untuk memberikan suatu nilai ke variabel.

Operator Type	Description
=	Assign a value to a variable
+=	Adds a value to a variable
-=	Reduces a value to a variable
*=	Multiply value to a variable
/=	Divided value by a variable

```
1 ▾ main() {  
2     double age = 24;  
3     age += 1;  
4     print("Setelah penambahan, variabel age menjadi: $age");  
5     age -= 1;  
6     print("Setelah pengurangan, variabel age menjadi: $age");  
7     age *= 2;  
8     print("Setelah perkalian, variabel age menjadi: $age");  
9     age /= 2;  
10    print("Setelah pembagian, variabel age menjadi: $age");  
11 }
```

Relational and Logical Operators

Digunakan untuk perbandingan.

Operator Symbol	Operator Name	Description
>	Greater than	Used to check which operand is bigger and gives result as boolean
<	Less than	Used to check which operand is smaller and gives result as boolean
>=	Greater than or equal to	Used to check which operand is bigger or equal and gives result as boolean
<=	Less than or equal to	Used to check which operand is smaller or equal and gives result as boolean
==	Equal to	Used to check operands are equal to each other and gives result as boolean
!=	Not equal to	Used to check operand are not equal to each other and gives result as boolean
&&	AND	Return true if all conditions are true
	OR	Return true if one of the conditions is true
!	NOT	Return false if the result is true and vice versa

```
1 ▾ main() {  
2     int num1 = 10;  
3     int num2 = 5;  
4  
5     print(num1 == num2);  
6     print(num1 < num2);  
7     print(num1 > num2);  
8     print(num1 <= num2);  
9     print(num1 >= num2);  
10 }
```

```

1 ▾ main() {
2   int userid = 123;
3   int userpin = 456;
4
5   print((userid == 123) && (userpin == 456));
6   print((userid == 1213) && (userpin == 456));
7   print((userid == 123) || (userpin == 456));
8   print((userid == 1213) || (userpin == 456));
9   print((userid == 123) != (userpin == 456));
10 }
11

```

Type Test Operators

Digunakan untuk memeriksa tipe data saat runtime.

Operator Symbol	Operator Name	Description
is	is	Gives boolean value true if the object has a specific type
is!	is not	Gives boolean value false if the object has a specific type

```

1 ▾ main() {
2   double type1 = 5.0;
3   int type2 = 87;
4   String type3 = "educative";
5   bool type4 = true;
6
7   print(type1 is int);
8   print(type2 is int);
9   print(type3 is String);
10  print(type4 is double);
11  print(type4 is! double);
12 }
13

```

Selesaikan kode dibawah ini untuk mengkonversi suhu dari Fahrenheit ke Celcius dan Kelvin sehingga menghasilkan output berikut ini.

50 fahrenheit = 10.0 celsius

50 fahrenheit = 283.15 kelvin

```

1 ▾ main() {
2   var fahrenheit = 50;
3
4   // tulis kode anda disini
5 }

```


Collection

Terdapat tiga struktur data di Dart yaitu List, Set, dan Map.

List

List adalah array di Dart yang digunakan untuk menyimpan koleksi objek secara berurutan.

```
1 ▾ main() {  
2   var nama = ["Budi", "Joko", "Susi"];  
3   // membuat variabel list kosong  
4   var umur = <int>[];  
5  
6   // mengambil elemen ke 1  
7   print("Nama ke 1: ${nama[0]}");  
8   // mengetahui jumlah elemen list  
9   print("Panjang list: ${nama.length}");  
10  // menambahkan satu elemen ke list  
11  umur.add(20);  
12  // menambahkan beberapa elemen ke list  
13  umur.addAll([21, 22]);  
14  print("Isi list umur: $umur");  
15  
16  // menghapus satu elemen dari list  
17  nama.removeAt(1);  
18  print(nama);  
19  // menghapus semua isi list  
20  umur.clear();  
21  print(umur);  
22 }  
23
```

Umumnya saat kita ingin melakukan operasi terhadap seluruh elemen list kita akan menggunakan perulangan. Dart memiliki fungsi `map()` yang dapat digunakan untuk mempermudah proses tersebut.

Sebagai contoh, misalkan kita punya list nilai, dan setiap elemen dalam list nilai akan kita kalikan dengan 2.

```
1 ▾ main() {  
2   var nilaiLama = [1, 2, 3, 4, 5];  
3  
4   var nilaiBaru = nilaiLama.map((nilai) => nilai * 2).toList();  
5   print(nilaiLama);  
6   print(nilaiBaru);  
7 }
```

Buat kode untuk mengkonversi semua elemen pada list celsius menjadi fahrenheit menggunakan fungsi map().

Output: [77.0, 86.0, 82.4, 68.0, 82.4]

```
1 ▾ main() {  
2     var celsius = [25, 30, 28, 20, 28];  
3  
4     // tuliskan kode anda disini  
5 }
```

Set

Set adalah kumpulan item unik yang tidak berurutan. Artinya item tidak mempunyai posisi tertentu dalam suatu set, oleh karena itu suatu set tidak boleh mempunyai duplikat dari item yang sama.

```
1 ▾ main() {  
2     // membuat set  
3     var setOfNumbers = <num>{1, 1.5, 2, 2.5};  
4     // membuat set kosong  
5     var setOfFruit = <String>{};  
6  
7     // menambahkan item ke set  
8     setOfFruit.add('apples');  
9     setOfFruit.add('bananas');  
10    setOfFruit.add('oranges');  
11  
12    print(setOfNumbers);  
13    print(setOfFruit);  
14  
15    // menambahkan beberapa item ke set  
16    setOfFruit.addAll(['mangos', 'grapes', 'lemons']);  
17    print(setOfFruit);  
18  
19    // mengetahui panjang set  
20    print(setOfFruit.length);  
21  
22    // menghapus item  
23    setOfFruit.remove('bananas');  
24    print(setOfFruit);  
25  
26    // memeriksa apakah sebuah item ada di set  
27    print(setOfFruit.contains('grapes'));  
28 }
```

Map

Map adalah kumpulan pasangan item dan kunci yang tidak berurutan. Setiap item memiliki kunci yang unik.

```
1▼ main() {
2▼   var capitals = <String, String>{
3     'United States': 'Washington D.C.',
4     'England': 'London',
5     'China': 'Beijing',
6   };
7   print(capitals);
8
9   // mengambil item
10  print(capitals['England']);
11
12  // menambahkan item
13  capitals['Indonesia'] = 'Jakarta';
14  print(capitals);
15
16  // mengetahui panjang map
17  print(capitals.length);
18
19  // memeriksa kunci
20  print(capitals.containsKey('Indonesia'));
21  // memeriksa item
22  print(capitals.containsValue('Solo'));
23
24  // menampilkan semua kunci
25  print(capitals.keys);
26  // menampilkan semua item
27  print(capitals.values);
28
29  // menghapus kunci dan item
30  capitals.remove('China');
31  print(capitals);
32 }
33
```