

Bab 4

Praktikum

Pengenalan

Pemrograman Dart



Flutter

Dosen : Bambang Sugiarto, ST, MT

Program Studi S1 Teknik Informatika

Fakultas Teknik

Universitas Sangga Buana YPKP Bandung

*Dirangkum dari berbagai sumber referensi
(hanya untuk penggunaan internal/tidak untuk dipublikasikan)*

Tujuan Praktikum

- Mahasiswa mengetahui pemrograman Dart pada Flutter.
- Mahasiswa mampu membuat pemrograman Dart pada Flutter.
- Mahasiswa mampu membuat implementasi pemrograman Dart dengan membuat aplikasi mobile sederhana pada Flutter.



Pendahuluan


- **Dart** adalah bahasa pemrograman *general purpose* yang bersifat open source.
- Pemrograman Dart ini awalnya dikembangkan oleh Google.
- Dart adalah bahasa berorientasi objek dengan sintaks style bahasa C.
- Dart mendukung konsep pemrograman seperti antarmuka, kelas, dll.
- Tidak seperti bahasa pemrograman lain, Dart tidak mendukung array.
- Koleksi Dart dapat digunakan untuk mereplikasi struktur data seperti array, generik, dan pengetikan opsional.



Memulai Pemrograman Dart

- Buatlah file baru bernama *program_pertama.dart*, di dalam program flutter yang sudah diinstall sebelumnya. Silahkan sesuaikan dengan IDE yang digunakan (Android Studio atau Visual Studio Code)
- Isi file *program_pertama.dart*, tersebut dengan kode berikut :

```
void main()  
{  
    print("Dart language is easy to learn");  
}
```

- 
- Silahkan lihat hasilnya.
 - Jika berhasil maka kita sudah bisa membuat program pertama Dart kita. Selanjutnya kita dapat mengikuti pada sub bab berikutnya.

Variabel dan Tipe Data

- **Variabel** adalah sebuah simbol yang digunakan untuk menyimpan nilai.
- Sedangkan **tipe data** adalah jenis nilai yang akan kita simpan dan berasosiasi dengan variabel atau fungsi.
- Dart menggunakan kata kunci **var** untuk mendeklarasikan variabel.
- Sintaks var didefinisikan di bawah ini :

```
var name = 'Dart';
```

- Kata kunci **final** dan **const** dapat digunakan untuk mendeklarasikan konstanta



- Sehingga dapat didefinisikan seperti berikut :

```
void main() {  
    final a = 12;  
    const pi = 3.14;  
    print(a);  
    print(pi);  
}
```

- Bahasa Dart mendukung tipe data berikut:
 - **Number**, digunakan untuk mewakili literal numerik - Integer dan Double.
 - **String**, Mewakili urutan karakter. Nilai string ditentukan dalam tanda kutip tunggal atau ganda.
 - **Boolean**, art menggunakan kata kunci bool untuk merepresentasikan nilai Boolean yaitu true dan false.
 - **List and Maps**, digunakan untuk merepresentasikan kumpulan objek.



- ✓ Contoh list sederhana dapat didefinisikan sebagai berikut :

```
void main() {  
    var list = [1,2,3,4,5];  
    print(list);  
}
```

- ✓ List yang ditunjukkan di atas menghasilkan list [1,2,3,4,5].
- ✓ Map dapat didefinisikan seperti yang ditunjukkan :

```
void main() {  
    var mapping = {'id': 1, 'name': 'Dart'};  
    print(mapping);  
}
```

- **Dynamic**, jika jenis variabel tidak ditentukan, maka jenis defaultnya adalah dinamis.
- ✓ Contoh berikut menggambarkan variabel tipe dinamis :



```
void main() {  
    dynamic name = "Dart";  
    print(name);  
}
```



Pembuatan Keputusan dan Loop

- Blok pengambilan keputusan mengevaluasi suatu kondisi sebelum instruksi dieksekusi.
- Dart mendukung pernyataan `if`, `if..else` dan `switch`.
- Loop digunakan untuk mengulang satu blok kode hingga kondisi tertentu terpenuhi.
- Dart mendukung `loop for`, `for..in`, `while` dan `do.. while`.
- Contoh sederhana tentang penggunaan pernyataan kontrol dan loop dapat dilihat pada program berikut ini :

```
void main() {  
    for( var i = 1 ; i <= 10; i++ ) {  
        if(i%2==0)  
        {  
            print(i);  
        }  
    }  
}
```



Fungsi (Function)

- Fungsi adalah sekelompok pernyataan yang bersama-sama melakukan tugas tertentu.
- Contoh fungsi sederhana pada Dart :

```
void main() {  
    add(3,4);  
}
```

```
void add(int a,int b) {  
    int c;  
    c=a+b;  
    print(c);  
}
```



Pemrograman Berorientasi Objek

- Dart adalah bahasa berorientasi objek.
- Ini mendukung fitur pemrograman berorientasi objek seperti kelas, antarmuka, dll.
- Kelas adalah blue print untuk membuat objek.
- Definisi kelas meliputi yang berikut ini :
 - Fields
 - Getters dan setter
 - Constructor
 - Function
- Berikut contoh kelas sederhana menggunakan definisi di atas :



```
class Employee {
    String name;
    //getter method
    String get emp_name {
        return name;
    }

    //setter method
    void set emp_name(String name) {
        this.name = name;
    }

    //function definition
    void result()
    {
        print(name);
    }
}

void main() {
    //object creation
    Employee emp = new Employee();
    emp.name="employee1";
    emp.result(); //function call
}
```



References

- Flutter Tutorial – Install Flutter pada Windows, Mac OS dan Linux, TeknoTUT, <https://www.teknotut.com/flutter-tutorial-install-flutter-pada-windows-mac-os-dan-linux/>.
- Flutter - Introduction to Dart Programming, Tutorialspoint, https://www.tutorialspoint.com/flutter/flutter_introduction_to_dart_programming.htm.
- Mengenal Struktur Flutter dan Widget, Plimbi, <https://www.plimbi.com/article/175647/mengenal-struktur-fluter-dan-widget>.

