

Advance Function, Async Await, Collection Dart

Outline

Fungsi (Lanjutan)

Async-Await

Tipe Data Future

Collection

A large, solid yellow circle is partially visible on the left side of the slide.

Fungsi (Lanjutan)

A large, light yellow circle is partially visible on the right side of the slide.

Anonymous Function

- Tidak memiliki nama
- Fungsi sebagai data

Ditulis dengan:



```
() {  
    // perintah yang dijalankan saat fungsi dipanggil  
}
```

Anonymous Function (contd)

```

var hello = () {
    print('hello');
};

var jumlah = (int a, int b) {
    return a + b;
};

void main() {
    hello();
    print(jumlah(1, 2));
}

```

Diagram annotations:

- An arrow points from the text "variabel" to the variable `hello` in the first line of code.
- An arrow points from the text "anonymous function" to the function definition `() { print('hello'); }` in the first line of code.
- An arrow points from the text "memanggil anonymous function" to the function call `hello();` in the `main` function.

Arrow Function

- Dapat memiliki nama atau tidak
- Berisi 1 data (dari proses maupun data statis)
- Nilai return fungsi ini diambil dari data tersebut

Ditulis dengan:



```
() => proses_yang_dijalankan_saats_fungsi_dipanggil();
```

Arrow Function (contd)

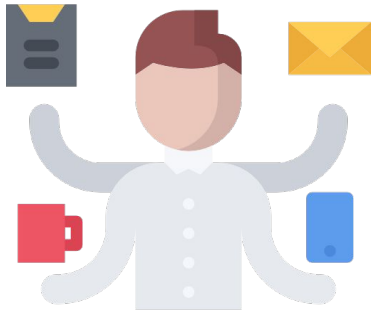
- Fungsi hello melakukan print di dalamnya
- Fungsi jumlah memberi nilai saat dijalankan sehingga dapat dilakukan print

```
var hello = () => print('hello');  
var jumlah = (int a, int b) => a + b;  
  
void main() {  
    hello();  
    print(jumlah(1, 2));  
}
```

The background features two large, semi-transparent circular shapes. On the left, a bright yellow circle is partially visible. On the right, a larger, light cream-colored circle is partially visible. The text "Async-Await" is centered between these two shapes.

Async-Await

Deskripsi



- Menjalankan beberapa proses tanpa perlu menunggu
- Proses ditulis dalam bentuk fungsi
- Await akan menunggu hingga proses async selesai



Simulasi

P1

menampilkan teks ke layar
dalam waktu 1 detik

1 detik



P2

menampilkan teks ke layar
tanpa waktu tunggu



Simulasi (contd)

```
Future<void> P1 () async {
    await Future.delayed(Duration(seconds: 1), () {
        print('hello dari P1');
    });
}
```

```
void P2 () {
    print('hello dari P2');
}
```

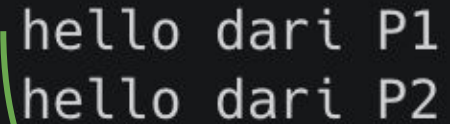
```
void main() async {
```

fungsi main menjadi async
karena di dalamnya ada await

```
    await P1();
    P2();
```

P2 dijalankan belakangan
setelah P1 selesai

```
}
```



```
hello dari P1
hello dari P2
```

P1 selesai lebih dahulu

The background features two large, overlapping circles. The circle on the left is a solid bright yellow and is partially cut off by the left edge of the frame. The circle on the right is a light cream or off-white color and is also partially cut off by the right edge of the frame. The text "Tipe Data Future" is centered between these two circles.

Tipe Data Future


Deskripsi



Data yang dapat ditunggu



Membawa data return dari
fungsi async



Penggunaan Future

```
Future<String> P1 () {
    return Future.delayed(Duration(seconds: 1), () {
        return 'hello dari P1';
    });
}
```

data yang di-return

```
void main() async {
    var data = await P1();
    print(data);
}
```

P1 menghasilkan Future yang di dalamnya terdapat sebuah String

```
hello dari P1
```

Collection

Kumpulan data pada satu tempat

List

Menyimpan data (elemen) secara berbaris

...
0	1	2	3	4

Tiap data memiliki index

List (contd)

```
void main() async {
```

```
var scores = [];  
scores.add(60);  
scores.add(80);  
scores.add(90);  
scores.add(70);  
scores.add(85);  
print(scores);
```

membuat list

menambah data

```
print(scores[0]);  
print(scores[1]);  
print(scores[2]);  
print(scores[3]);  
print(scores[4]);
```

mengambil data
berdasarkan index

```
}
```

```
[60, 80, 90, 70, 85]  
60  
80  
90  
70  
85
```

List (contd)

```
void main() async {
```

```
    var scores = [60, 80, 90, 70, 85];  
    print(scores);
```

```
    for(var score in scores) {  
        print(score);  
    }
```

```
}
```

membuat list disertai data-datanya

mengambil seluruh data pada list

Map

Menyimpan data secara key-value
Key berguna selayaknya index pada list

key1	value1
key2	value2
key3	value3
key4	value4
key5	value5

Map (contd)

```
void main() async {
    var student = {};
    student['name'] = 'Alex Under';
    student['age'] = 16;
    print(student);

    print(student['name']);
    print(student['age']);
}
```

membuat map

menambah data

mengambil data berdasarkan key

```
{name: Alex Under, age: 16}
Alex Under
16
```

Map (contd)

```
void main() async {
```

```
    var student = {  
        'name': 'Alex Under',  
        'age': 16  
    };  
    print(student);
```

membuat map disertai
data-datanya

```
    for(var key in student.keys) {  
        print(student[key]);  
    }
```

mengambil seluruh
data pada list

```
}
```

A large, solid yellow circle is centered on the slide, serving as a background for the text.

Terima Kasih

Task

1. Buatlah sebuah fungsi dengan spesifikasi berikut:
 - a. menerima 2 parameter, yaitu list data dan pengali,
 - b. lakukan perulangan pada list data secara asynchronous,
 - c. tiap perulangan, kalikan elemen list data dengan pengali,
 - d. return list baru yang berisi hasil proses di atas.

2. Buatlah sebuah list dengan spesifikasi berikut:
 - a. tiap element-nya berupa list juga,
 - b. tiap element wajib terdapat 2 data (sub-element).
 Buatlah sebuah Map dengan menggunakan list tersebut!

