

Hari-6-List dan Maps

video materi

<https://www.youtube.com/watch?v=Jy4eIW9Y64M>

1. List

list adalah sebuah object koleksi yang menyimpan daftar objek/ suatu kelompok nilai atau data yang disusun berdasarkan urutan index. Dalam Dart list dinyatakan dalam tipe List, untuk membuat list sendiri, kita dapat menggunakan kurung [], sama seperti pada array contoh :

```
void main() {  
  List<int> list = [23, 24, 25];  
  print(list[0]);  
  print(list[1]);  
  print(list[2]);  
}
```

Sedangkan penambahan elemen ke dalam objek list dilakukan dengan cara menambahkan metode add(), seperti berikut.

```
void main() {  
  List<int> list = [23, 24, 25];  
  list.add(0);  
  print(list[0]);  
  print(list[1]);  
  print(list[2]);  
  print(list[3]);  
}
```

2. Tipe Data Map

Pengertian dari tipe data map dalam dart adalah objek koleksi yang setiap elemennya berupa pasangan kunci (*key*) dan nilai (*value*). Terdapat asosiasi antara kunci dan nilai pada setiap elemen yang terdapat didalam elemen map.

Dalam satu objek map, kunci harus bersifat unik,namun tidak dengan nilai. Hal ini berarti bahwa satu nilai yang sama bisa saja muncul pada beberapa elemen map

Dalam dart, map dinyatakan dengan tipe map. Objek dari kelas Map dibuat dengan menggunakan tanda { }, dengan bentuk umum sebagai berikut.

contoh :

```
void main() {  
  
  Map<String, String> kota = {'jkt' : 'Jakarta', 'bdg' : 'Bandung', 'sby' : 'Surabaya'};  
  
  print(kota['jkt']);  
  print(kota['bdg']);  
  print(kota['sby']);  
}
```

3. Menampilkan berbagai tipe data

```
print("Total bahasa: ${languages.length}");}
```

4. List Multidimensi

Multidimensional list atau list multidimensi adalah list yang berisi list di dalamnya. Jumlah dimensi bergantung seberapa dalam list tersebut memiliki list di dalamnya. Cara akses nilai dari list multidimensi sama seperti list satu dimensi seperti biasa namun jumlah indeksnya terdapat sebanyak dimensi nya.

```
var arrayMulti = [
  [1, 2, 3],
  [4, 5, 6],
  [7, 8, 9]
];
// Maka sebagai gambaran, indeks dari array tersebut adalah
/*
  [
    [(0,0), (0,1), (0,2)],
    [(1,0), (1,1), (1,2)],
    [(2,0), (2,1), (2,2)]
  ]
*/
void main(){
print(arrayMulti[0][0]); // 1
print(arrayMulti[1][0]); // 4
print(arrayMulti[2][1]); // 8
}
```

5. Method yang ada pada List dan maps

Ketika membuat aplikasi, kita pasti akan berurusan dengan yang namanya List. Pada kali ini kita akan mengambil utility method list dari Dart yang sering digunakan dan diterapkan.

forEach()

function untuk menampilkan tiap-tiap elements.

```
var perusahaan = ['bukalapak', 'tokopedia', 'blibli'];
perusahaan.forEach((data)=> print(data)); bukalapak tokopedia blibli
```

map()

membuat list baru setelah setiap element diubah.

```
var perusahaan_id = perusahaan.map((data)=> '${data} indonesia').toList();
```

contains()

Mengecekan element dalam list, yang mana return bool.

```
print(perusahaan.contains('bukalapak'));
```

sort()

untuk melakukan sorting pengurutan dalam list.

```
var randomdata = [1,3,5,20,4,2];
randomdata.sort((a, b)=> a.compareTo(b));
print(randomdata);
```

reduce(), fold()

kompres list pada element jadi single value.

```
var sumData = randomdata.reduce((cur, next)=> cur + next);
print(sumData); /// 35const currentValue = 10;
var nextSum = randomdata.fold(currentValue, (cur, next)=> cur + next);
print(nextSum); // 45
```

every()

untuk melakukan check, apakah tiap element yang di iterasi memenuhi **test**

```
List<Map<String, dynamic>> listUser = [
  {'nama': 'bekasi', 'umur': 240},
  {'nama': 'boyolali', 'umur': 200},
  {'nama': 'jakarta', 'umur': 100},
  {'nama': 'surabaya', 'umur': 100},
];var example = listUser.every((data) => data['umur'] >= 100);
print(example); ///true
```

where(), firstWhere(), singleWhere()

mengembalikan nilai list setelah memenuhi kondisi.

```
var userYoung = listUser.where((data)=> data['umur'] > 100);
print(userYoung);

var userFirstYoung = listUser.firstWhere((data)=> data['umur'] < 200);
print(userFirstYoung); /// {'nama': 'jakarta', 'umur': 100},

var userSingle = listUser.singleWhere((data)=> data['umur'] <= 100);
print(userSingle); /// error karena ada dua kondisi yang benar
```

firstWhere itu ngambil list pertama dari banyak element **true**, dan **singleWhere** itu adalah kondisi true hanya boleh satu.

take(), skip()

ingin mengambil beberapa element dari banyaknya data pada list, gunakan method **take()** dan **skip()**

expand()

ingin melakukan flatMap? sangat bisa dengan menggunakan `expand()`

```
var pairs = [[1, 2], ['a', 'b'], [3, 4]];
var flatmaps = pairs.expand((pair)=> pair);
print(flatmaps);
```

List Comprehensions

ini bukan method tapi ini adalah update terbaru dari `dart 2.3.2` yang mana Dart bisa membuat list comprehension seperti pada bahasa pemogramman python.

```
var comph = [1,2,3,4];
var newCom = [for(var a in comph) 'new ${a}'];
```

Add dan AddAll

ini merupakan method yang di pakai untuk menambahkan index pada list kita contohnya :

```
void main(){
```

```
    List<int> myList = [];

    List<int> list = [1,2,3];

    myList.add(1);

    myList.addAll(list);

    myList.forEach((bilangan)=>{

        print(bilangan)

    });
```

```
}
```

Rating - Feedback

Berikan Rating pada posting ini:



Berikan kritik dan saran..

Submit