

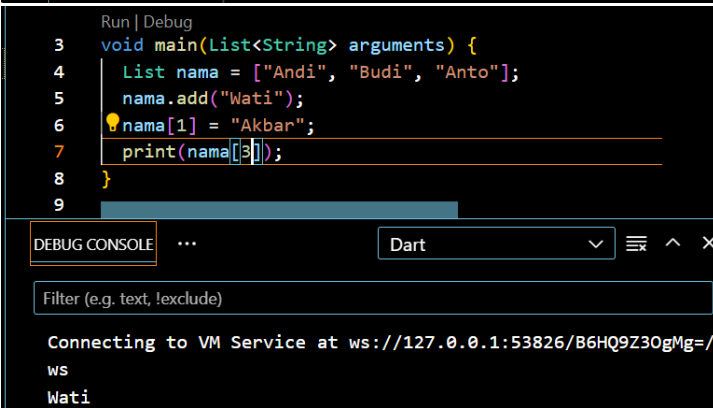
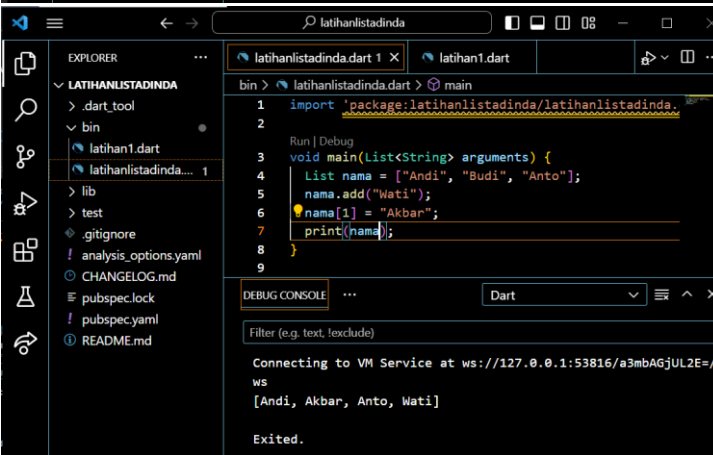
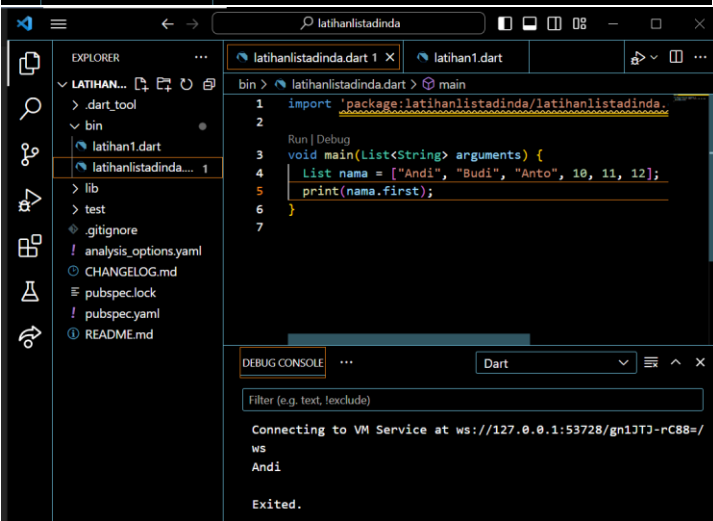
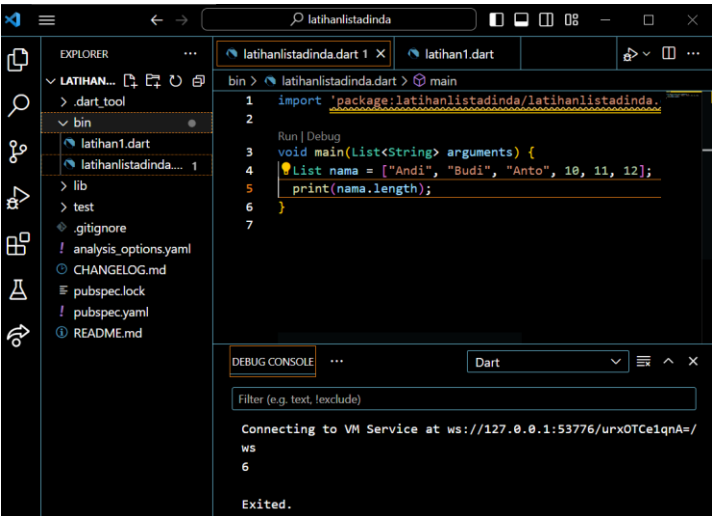
WEEK 4
TASK REPORT
-MOBILE PROGAMMING -



2023/2024

By :
Adinda Rahajeng Silvia Pranesti
2142720158 / 02
TI3I

1. UJI COBA W/ MR.ADE



latihanlistadinda.dart 1latihan1.dart X

bin > latihan1.dart > main

Run | Debug

```
1 void main(List<String> arguments) {
2   Set<String> nama = Set<String>();
3   nama.add("Polinema1");
4   nama.add("Polinema2");
5   nama.add("Polinema3");
6   for (String n in nama) {
7     print("nama $n");
8   }
9 }
10
```

DEBUG CONSOLE

Dart

Filter (e.g. text, lexclude)

Connecting to VM Service at ws://127.0.0.1:53875/knJNM2CSyqQ=/ws

nama Polinema1
nama Polinema2
nama Polinema3

3. Collection : maps : using the map constructor, adding elements n accesing spesific elements

latihanlistadinda.dart 1maps.dart Xlatihan1.dart

bin > maps.dart > main

Run | Debug

```
1 void main() {
2   var gifts = {
3     // Key: Value
4     'first': 'partridge',
5     'second': 'turtledoves',
6     'fifth': 1
7   };
8
9   var nobleGases = {
10    2: 'helium',
11    10: 'neon',
12    18: 2,
13  };
14
15  print(gifts);
16  print(nobleGases);
17 }
18
```

Filter (e.g. text, lexclude)

Dart

Connecting to VM Service at ws://127.0.0.1:53934/YhL_vPLOT_g=/ws

{first: partridge, second: turtledoves, fifth: 1}
{2: helium, 10: neon, 18: 2}

latihanlistadinda.dart 1maps.dart Xlatihan1.dart

bin > maps.dart > main

Run | Debug

```
8
9
10 var nobleGases = {
11   2: 'helium',
12   10: 'neon',
13   18: 2,
14 };
15
16 var mhs1 = Map<String, String>();
17 gifts['first'] = 'partridge';
18 gifts['second'] = 'turtledoves';
19 gifts['fifth'] = 'golden rings';
20
21 var mhs2 = Map<int, String>();
22 nobleGases[2] = 'helium';
23 nobleGases[10] = 'neon';
24 nobleGases[18] = 'argon';
25
26 print(gifts);
27 print(nobleGases);
28
```

Filter (e.g. text, lexclude)

Dart

Connecting to VM Service at ws://127.0.0.1:53957/Et_1vQ_iz3w=/ws

{first: partridge, second: turtledoves, fifth: golden rings}
{2: helium, 10: neon, 18: argon}

2. PRAKTIKUM 1

- UJICOBAl

```
void main() {
  var list = [1, 2, 3];
  assert(list.length == 3);
  assert(list[1] == 2);
  print(list.length);
  print(list[1]);

  list[1] = 1;
  assert(list[1] == 1);
  print(list[1]);
}
```

OUTPUT :

```
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> dart run bin/praktikum1.dart
3
2
1
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> 
```

Penjelasan : Hasilnya mencerminkan perubahan yang dilakukan pada list `list` sesuai dengan instruksi dalam kode.

- Updated with nama & nim

```
void main() {
  final List<String?> mahasiswa = List.filled(5, null);
  mahasiswa[1] = 'Adinda Rahajeng';
  mahasiswa[2] = '2141720158';

  print('Nama: ${mahasiswa[1]}');
  print('NIM: ${mahasiswa[2]}');
}
```

Output :

```
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> dart run bin/praktikum1updated.dart
Nama: Adinda Rahajeng
NIM: 2141720158
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> 
```

3. PRAKTIKUM 2



- With updated nama & NIM

```
void main() {
  var names1 = <String>{};
  Set<String> names2 = {};
  var names3 = {}; // Ini akan membuat Map, bukan Set.

  // Menambahkan elemen nama dan NIM Anda pada names1 dan names2.
```

```
names1.add('Adinda Rahajeng');
names1.add('2141720158');

names2.addAll(['Adinda Rahajeng', '2141720158']);

print('names1 (Set): $names1');
print('names2 (Set): $names2');
print('names3 (Map): $names3');
}
```

Output :

```
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> dart run bin/praktikum2.dart
names1 (Set): {Adinda Rahajeng, 2141720158}
names2 (Set): {Adinda Rahajeng, 2141720158}
names3 (Map): {}
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> █
```

4. PRAKTIKUM 3:

```
void main() {
  var gifts = {
    // Key: Value
    'first': 'partridge',
    'second': 'turtledoves',
    'fifth': 'golden rings',
  };

  var nobleGases = {
    2: 'helium',
    10: 'neon',
    18: 'argon',
  };

  var mhs1 = Map<String, String>();
  gifts['first'] = 'partridge';
  gifts['second'] = 'turtledoves';
  gifts['fifth'] = 'golden rings';

  var mhs2 = Map<int, String>();
  nobleGases[2] = 'helium';
  nobleGases[10] = 'neon';
  nobleGases[18] = 'argon';

  print(gifts);
  print(nobleGases);
}
```

OUTPUT :

```
Connecting to VM Service at ws://127.0.0.1:54041/7myuV2XVqQQ=/ws
{first: partridge, second: turtledoves, fifth: golden rings}
{2: helium, 10: neon, 18: argon}

Exited.
```

ASSIGNMENT PRAKTIKUM 3:

```
void main() {
  var gifts = {
```

```
// Key: Value
'first': 'partridge',
'second': 'turtledoves',
'fifth': 'golden rings',
'yourName': 'Adinda Rahajeng SP',
'yourNIM': '2141720158',
};

var nobleGases = {
  2: 'helium',
  10: 'neon',
  18: 'argon',
  32: 'Adinda Rahajeng SP',
  100: '2141720158',
};

var mhs1 = Map<String, String>();
mhs1['name'] = 'Adinda Rahajeng SP';
mhs1['nim'] = '2141720158';

var mhs2 = Map<int, String>();
mhs1['32'] = 'Adinda Rahajeng SP';
mhs1['12345'] = '2141720158';

print(gifts);
print(nobleGases);
print(mhs1);
print(mhs2);
}
```

OUTPUT :

```
Connecting to VM Service at ws://127.0.0.1:54008/q_a4rr7e80A=ws
{first: partridge, second: turtledoves, fifth: golden rings, yourName: Adinda Rahajeng SP, yourNIM: 2141720158}
{2: helium, 10: neon, 18: argon, 32: Adinda Rahajeng SP, 100: 2141720158}
{name: Adinda Rahajeng SP, nim: 2141720158, 32: Adinda Rahajeng SP, 12345: 2141720158}
{}

Exited.
```

5. PRAKTIKUM 4

- Sudah dalam perbaikan



The screenshot shows an IDE with a file explorer at the top containing files like 'latihan1.dart', 'maps.dart 2', 'PRAKTIKUM3.DART', 'prak5.dart 1', and 'latihan1.dart'. The main editor displays Dart code for a 'main' function. The code defines a list, creates a new list with an additional element, prints the original list, the new list, and their lengths. A red box highlights the code for creating 'list3' and printing its length. The bottom panel shows the 'OUTPUT' tab with the execution results, which match the output shown in the previous block. A red box highlights the output section of the IDE interface.

```
bin > prak5.dart > main
1 void main() {
2   var list = [1, 2, 3];
3   var list2 = [0, ...list];
4   print(list);
5   print(list2);
6   print(list2.length);
7
8   var list1 = [1, 2, null];
9   print(list1);
10  var list3 = [0, ...list1, 1]; // Menggunakan nullish-coalescing-operator
11  print(list3.length);
12
13
14

PROBLEMS (6) OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, exclude)
Connecting to VM Service at ws://127.0.0.1:50516/tT1Z_VXh-zA=ws
[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2, null]
4
Exited.
```

- Menambahkan NIM

The screenshot shows the VS Code editor with a file named `prak5.dart` open. The code defines a list `list1` with values `[1, 2, null]`, prints it, and then creates a new list `list3` using the `?? []` operator. A comment indicates the next step is to add a new variable `NIM`. The code then defines `NIM` as `"2141720158"`, creates a new list `list4` by mapping the characters of `NIM` to their character codes, and prints it. The output in the debug console shows the sequence of operations: connecting to the VM service, printing `[1, 2, 3]`, `[0, 1, 2, 3]`, `4`, `[1, 2, null]`, `4`, and finally `[0, 2, 1, 4, 1, 7, 2, 0, 1, 5, 8]`.

```
bin > prak5.dart > main
8 var list1 = [1, 2, null];
9 print(list1);
10 var list3 = [0, ...(list1 ?? [])]; // Menggunakan nullish coale
11 print(list3.length);
12
13 Menambahkan variabel baru yaitu NIM ADINDA RAHAJENG SP = 214172
14 var NIM = "2141720158";
15 var list4 = [0, ...NIM.runes.map((rune) => String.fromCharCode(
16
17 print(list4);
18
19
```

DEBUG CONSOLE

Connecting to VM Service at ws://127.0.0.1:50558/GkiTG0hE7MA=/ws

[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2, null]
4
[0, 2, 1, 4, 1, 7, 2, 0, 1, 5, 8]
Exited.

- Plants True

The screenshot shows the VS Code editor with a file named `prak4plant.dart` open. The code defines a `main` function where `promoActive` is set to `true`. A list `nav` is created with values `['Home', 'Furniture', 'Plants', promoActive ? 'Outlet']`, and the list is printed. The output in the debug console shows the connection to the VM service and the printed list `[Home, Furniture, Plants, Outlet]`.

```
bin > prak4plant.dart > main
Run | Debug
1 void main() {
2   var promoActive = true; // Ganti dengan true atau false sesuai
3   var nav = ['Home', 'Furniture', 'Plants', promoActive ? 'Outlet
4   print(nav);
5
6
7
```

DEBUG CONSOLE

Connecting to VM Service at ws://127.0.0.1:50589/SYgKzsz3F1M=/ws

[Home, Furniture, Plants, Outlet]

- Plants False

The screenshot shows the VS Code editor with a file named `prak4plant.dart` open. The code defines a `main` function where `promoActive` is set to `false`. A list `nav` is created with values `['Home', 'Furniture', 'Plants', promoActive ? 'Outlet']`, and the list is printed. The output in the debug console shows the connection to the VM service and the printed list `[Home, Furniture, Plants,]`.

```
bin > prak4plant.dart > main
Run | Debug
1 void main() {
2   var promoActive = false; // Ganti dengan true atau false sesuai
3   var nav = ['Home', 'Furniture', 'Plants', promoActive ? 'Outlet
4   print(nav);
5 }
6
```

DEBUG CONSOLE

Connecting to VM Service at ws://127.0.0.1:50631/_K2Ln84-m8Y=/ws

[Home, Furniture, Plants,]
Exited.

- Login manager :

```
6 var login = 'Manager'; // Ganti dengan nilai login yang sesuai
7 var nav2 = [
8   'Home',
9   'Furniture',
10  'Plants',
11  login == 'Manager' ? 'Inventory' : ''
12 ];
13 print(nav2);
14 }
```

PROBLEMS (8) OUTPUT **DEBUG CONSOLE** TERMINAL PORTS

Connecting to VM Service at ws://127.0.0.1:50681/FrmQuSJEJ80=/ws
[Home, Furniture, Plants,]
[Home, Furniture, Plants, Inventory]
Exited.

- Employee Login

```
4 print(nav);
5
6 var login = 'Manager'; // Ganti dengan nilai login yang sesuai
7 var nav2 = [
8   'Home',
9   'Furniture',
10  'Plants',
11  login == 'Employee' ? 'Inventory' : ''
12 ];
13 print(nav2);
14 }
15
```

PROBLEMS (8) OUTPUT **DEBUG CONSOLE** TERMINAL PORTS

Connecting to VM Service at ws://127.0.0.1:50740/7Co7rc1W99Y=/ws
② [Home, Furniture, Plants,]
Exited.

- Listnumber

```
bin > prak4number.dart > ...
Run | Debug
1 void main() {
2   var listOfInts = [1, 2, 3];
3   var listOfStrings = ['#0', for (var i in listOfInts) '$i'];
4   print(listOfStrings);
5 }
6
```

PROBLEMS (8) OUTPUT **DEBUG CONSOLE** **TERMINAL** PORTS

PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> dart run bin/prak4number.dart
[#0, #1, #2, #3]

Jelaskan manfaat **Collection For** : Manfaatnya yaitu untuk memungkinkan kita untuk membuat koleksi dengan lebih ekspresif dan kompak, menghindari perlu menulis loop eksternal secara manual untuk mengisi koleksi. Ini membantu dalam menghasilkan kode yang lebih mudah dibaca dan dipahami.

6. Praktikum 5

```
void main() {
  var record = {'first': 1, 'a': 2, 'b': true, 'last': 3};
  print(record);
}
```

Output :

```
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> dart run bin/praktikum5.dart
{first: 1, a: 2, b: true, last: 3}
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda>
```

- Tukar record

```
List<int> tukar(List<int> record) {
  int a = record[0];
  int b = record[1];
  return [b, a];
}

void main() {
  List<int> record = [1, 2];
  print('Sebelum tukar: $record');
```



```
record = tukar(record);
print('Setelah tukar: $record');
}
```

Output :

```
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> dart run bin/praktikum5.dart
Sebelum tukar: [1, 2]
Setelah tukar: [2, 1]
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> []
```

- Nama & NIM praktikum 5

```
void main() {
  // Inisialisasi variabel mahasiswa dengan nama dan NIM Anda
  List<dynamic> mahasiswa = ['Adinda Rahajeng', 2141720158];

  // Mencetak nilai variabel mahasiswa
  print(mahasiswa);
}
```

Output :

```
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> dart run bin/prak5nanim.dart
[Adinda Rahajeng, 2141720158]
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda>
```

```
void main() {
  var mahasiswa2 = {
    'nama': 'Adinda Rahajeng',
    'NIM': 2141720158,
    'a': 2,
    'b': true,
  };

  print(mahasiswa2['nama']); // Prints 'Adinda Rahajeng'
  print(mahasiswa2['NIM']); // Prints 2141720158
  print(mahasiswa2['a']); // Prints 2
  print(mahasiswa2['b']); // Prints true
}
```

Output :

```
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> dart run bin/prak5nanim.dart
Adinda Rahajeng
2141720158
2
true
PS D:\SEMESTER 5\Pemograman Mobile\WEEK 4\LatihanList\latihanlistadinda> []
```

7. Tugas Praktikum

- 1) Silakan selesaikan Praktikum 1 sampai 5, lalu dokumentasikan berupa screenshot hasil pekerjaan Anda beserta penjelasannya! → **done**
- 2) Jelaskan yang dimaksud Functions dalam bahasa Dart!

Answer =

- Functions (fungsi) dalam bahasa Dart adalah blok kode yang dapat digunakan untuk melakukan tugas atau operasi tertentu. Mereka digunakan untuk mengelompokkan kode ke dalam unit yang dapat digunakan kembali, sehingga Anda dapat memanggilnya dengan mudah ketika diperlukan.
 - Fungsi dapat memiliki nol atau lebih parameter sebagai input dan dapat mengembalikan nilai (atau void jika tidak mengembalikan nilai).
 - Fungsi dalam Dart dapat didefinisikan dengan kata kunci void (jika tidak mengembalikan nilai) atau dengan tipe data yang spesifik jika mengembalikan nilai.
- 3) Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!

Answer =

Ada beberapa jenis parameter dalam Dart, termasuk:

- **Required Parameters:** Ini adalah parameter yang wajib diberikan saat memanggil fungsi.
 - ➔
 - ➔ void printName(String name) {
 - ➔ print(name);
 - ➔ }
- **Optional Parameters:** Ini adalah parameter yang tidak wajib diberikan saat memanggil fungsi. Ada dua jenis opsional: menggunakan kurung kurawal {} atau tanda kurung kotak [].
 - ➔ void greet({String? name}) {
 - ➔ if (name != null) {
 - ➔ print('Hello, \$name!');
 - ➔ } else {
 - ➔ print('Hello, anonymous!');
 - ➔ }
 - ➔ }
 - ➔ }
 - ➔ void logMessage([String? message]) {
 - ➔ if (message != null) {
 - ➔ print(message);
 - ➔ } else {
 - ➔ print('No message provided.');
 - ➔ }
 - ➔ }

- 4) Jelaskan maksud Functions sebagai first-class objects beserta contoh sintaknya!

Answer = Dalam Dart, fungsi adalah objek pertama (first-class objects), yang berarti Anda dapat:

- Menyimpan fungsi dalam variabel.
- Mengirimkan fungsi sebagai argumen ke fungsi lain.
- Mengembalikan fungsi dari fungsi lain.

Contoh:

- ➔ // Menyimpan fungsi dalam variabel
- ➔ int Function(int, int) add = (a, b) => a + b;
- ➔
- ➔ // Mengirimkan fungsi sebagai argumen
- ➔ int performOperation(int Function(int, int) operation, int x, int y) {
- ➔ return operation(x, y);
- ➔ }
- ➔
- ➔ // Mengembalikan fungsi dari fungsi lain
- ➔ Function calculator() {

```
➔ return (a, b) => a + b;  
➔ }
```

5) Apa itu Anonymous Functions? Jelaskan dan berikan contohnya!

Answer = Anonymous functions (fungsi anonim) adalah fungsi yang tidak memiliki nama. Mereka digunakan ketika Anda hanya perlu mendefinisikan fungsi sekali dan tidak ingin memberikannya nama.


Contoh:


```
➔ // Anonymous function sebagai parameter  
➔ void doSomething(void Function() callback) {  
➔ callback();  
➔ }
```

```
➔ // Penggunaan anonymous function  
➔ doSomething(() {  
➔ print('Doing something...');  
➔ });
```

6) Jelaskan perbedaan Lexical scope dan Lexical closures! Berikan contohnya!

Answer :

 **Lexical Scope:** Ini mengacu pada aturan di mana variabel dapat diakses berdasarkan lingkup (scope) mereka pada saat kompilasi. Variabel dapat diakses dari dalam fungsi atau blok yang mengandung deklarasi variabel tersebut.

 **Lexical Closure:** Ini adalah kemampuan fungsi untuk mengakses variabel dari lingkup luaran (enclosing scope) bahkan setelah lingkup luaran tersebut selesai dieksekusi. Fungsi yang memiliki akses ke variabel-variabel dari lingkup luaran ini disebut closure.

Contoh:

```
➔ Function outerFunction(int x) {  
➔ int y = 10;  
➔ return () {  
➔ print(x + y); // Lexical Closure: Fungsi closure memiliki akses ke 'x' dan 'y'.  
➔ };  
➔ }
```

7) Jelaskan dengan contoh cara membuat return multiple value di Functions!

Answer : Dart tidak memiliki kemampuan langsung untuk mengembalikan multiple values dari fungsi seperti beberapa bahasa lain. Namun, Anda dapat menggunakan tipe data yang sesuai untuk mengembalikan berbagai jenis data, seperti menggunakan List, Map, atau Object untuk mengelompokkan nilai-nilai yang akan dikembalikan dari fungsi.

Contoh menggunakan List:

```
➔ List<int> getNumbers() {  
➔ return [1, 2, 3];  
➔ }
```

Contoh menggunakan Map:

```
➔ Map<String, dynamic> getPerson() {  
➔ return {  
➔ 'name': 'John',  
➔ 'age': 30,  
➔ };  
➔ }
```

➔ }

Contoh menggunakan Object:

➔ class Result {

➔ int code;

➔ String message;

➔ Result(this.code, this.message);

➔ }

➔ Result processRequest() {

➔ // ... beberapa operasi ...

➔ return Result(200, 'Sukses');

➔ }