|  |  |
| --- | --- |
|  | Nama : Moch Fikri Setiawan |
| Kelas : SIB 3E |
| NIM : 2241760105 |
| Absen : 16 |
| Jobsheet 4 Pemrogaman Mobile |

**Praktikum 1**

* **Langkah 1**

Ketik atau salin kode program berikut ke dalam :

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]);

}

* **Langkah 2**

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan!

: Kode Dart ini membuat list [1, 2, 3], memeriksa panjang dan nilai di indeks ke-1, mencetaknya, mengubah nilai indeks ke-1 menjadi 1, memeriksa perubahan, dan mencetak nilai baru.

* **Langkah 3**

Ubah kode pada langkah 1 menjadi variabel final yang mempunyai index = 5 dengan default value = null. Isilah nama dan NIM Anda pada elemen index ke-1 dan ke-2. Lalu print dan capture hasilnya.

Apa yang terjadi ? Jika terjadi error, silakan perbaiki.

: void main() {

  final list = List.filled(5, null);

  list[1] = 'Moch Fikri Setiawan';

  list[2] = '2241760105';

  assert(list.length == 5);

  assert(list[1] == 'Moch Fikri Setiawan');

  assert(list[2] == '2241760105');

  print(list.length);

  print(list[1]);

  print(list[2]);

  print(list);

}

Error dikarenakan adanya variable string yang tidak didefinisikan

Perbaikan code :

void main() {

  final List<dynamic> list = List.filled(5, null); // Membuat list dengan 5 elemen default null dan tipe dynamic

  list[1] = 'Moch Fikri Setiawan';   // Mengisi indeks ke-1

  list[2] = '2241760105';            // Mengisi indeks ke-2

  assert(list.length == 5);          // Memastikan panjang list adalah 5

  assert(list[1] == 'Moch Fikri Setiawan'); // Memastikan elemen pada indeks 1 benar

  assert(list[2] == '2241760105');    // Memastikan elemen pada indeks 2 benar

  print(list.length);                 // Mencetak panjang list

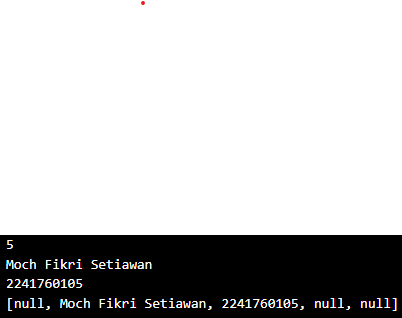
  print(list[1]);                     // Mencetak elemen indeks ke-1

  print(list[2]);                     // Mencetak elemen indeks ke-2

  print(list);                        // Mencetak seluruh isi list

}

Output :



**Praktikum 2**

* **Langkah 1**

Ketik atau salin kode program berikut ke dalam fungsi main():

void main() {

  var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};

  print(halogens);

}

* **Langkah 2**

**:** Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

**:** tidak terjadi eror

* **Langkah 3**

**:** Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

var names1 = <String>{};

Set<String> names2 = {}; // This works, too.

var names3 = {}; // Creates a map, not a set.

print(names1);

print(names2);

print(names3);

Apa yang terjadi ? Jika terjadi error, silakan perbaiki namun tetap menggunakan ketiga variabel tersebut. Tambahkan elemen nama dan NIM Anda pada kedua variabel Set tersebut dengan dua fungsi berbeda yaitu .add() dan .addAll(). Untuk variabel Map dihapus, nanti kita coba di praktikum selanjutnya.

: eror dikarenakan names3 adalah sebuah Map, bukan Set.

Perbaikan kode :

void main() {

  var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};

  print(halogens);

  var names1 = <String>{};

  Set<String> names2 = {};

  var names3 = <String>{}; // Mengubah menjadi Set<String>

  // Menambahkan nama dan NIM dengan .add() pada names1

  names1.add('Moch Fikri Setiawan');

  names1.add('2241760105');

  // Menambahkan nama dan NIM dengan .addAll() pada names2

  names2.addAll({'Moch Fikri Setiawan', '2241760105'});

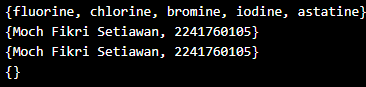
  print(names1);

  print(names2);

  print(names3);

}

Output



: Kode ini mendefinisikan dan mencetak tiga set—halogens, names1, dan names2—yang masing-masing berisi elemen yang berbeda, dengan names1 dan names2 diisi menggunakan metode .add() dan .addAll(), sementara names3 tetap kosong.

**Praktikum 3**

* **Langkah 1**

**:** ketik atau salin kode progam berikut ke dalam fungsi main()

void main(){

  var gifts = {

  // Key:    Value

  'first': 'partridge',

  'second': 'turtledoves',

  'fifth': 1

};

var nobleGases = {

  2: 'helium',

  10: 'neon',

  18: 2,

};

print(gifts);

print(nobleGases);

}

* **Langkah 2**

**:** Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

**:** tidak ada eror yang terjadi code berjalan dengan semestinya



* **Langkah3**

**:** tambahkan kode progam berikut, lalu eksekusi kode anda

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';

: apa yang terjadi ? jika terjadi error silahkan perbaiki

Tambahkan elemen nama dan nim anda pada tiap variable di atas (gifts, noblegases, mhs1 dan mhs2) dokumentasikan hasilnya dan buat laporannya!

: Kode ini menghasilkan error karena gifts memiliki nilai integer 1 yang diubah menjadi string 'golden rings', dan nobleGases memiliki nilai integer 2 yang diubah menjadi string 'argon', menyebabkan ketidakcocokan tipe data pada kedua Map.

**Perbaikan kode**

**:** void main() {

  var gifts = {

    // Key:    Value

    'first': 'partridge',

    'second': 'turtledoves',

    'fifth': 'golden rings', // Pastikan semua nilai adalah String

    'name': 'Moch Fikri Setiawan',

    'nim': '2241760105',

  };

  var nobleGases = {

    2: 'helium',

    10: 'neon',

    18: 'argon', // Pastikan semua nilai adalah String

    20: 'Moch Fikri Setiawan', // Menambahkan elemen dengan key baru

    30: '2241760105',          // Menambahkan elemen dengan key baru

  };

  print(gifts);

  print(nobleGases);

  var mhs1 = Map<String, String>();

  mhs1['first'] = 'partridge';

  mhs1['second'] = 'turtledoves';

  mhs1['fifth'] = 'golden rings';

  mhs1['name'] = 'Moch Fikri Setiawan';

  mhs1['nim'] = '2241760105';

  var mhs2 = Map<int, String>();

  mhs2[2] = 'helium';

  mhs2[10] = 'neon';

  mhs2[18] = 'argon';

  mhs2[20] = 'Moch Fikri Setiawan'; // Menambahkan elemen dengan key baru

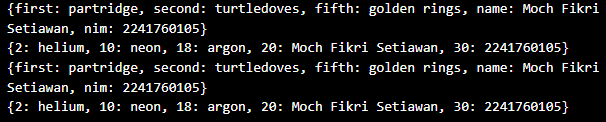
  mhs2[30] = '2241760105';         // Menambahkan elemen dengan key baru

  print(mhs1);

  print(mhs2);

}

**Output :**

****

**:** Kode ini memperbarui setiap Map dengan menambahkan elemen nama dan NIM, di mana gifts dan mhs1 menggunakan tipe data String untuk key yang baru, sementara nobleGases dan mhs2 menggunakan key integer baru untuk menambahkan elemen nama dan NIM.

**Praktikum 4**

* **Langkah 1**

**:** ketik atau salin kode progam berikut ke dalam fungsi main

void main() {

  var list = [1, 2, 3];

  var list2 = [0, ...list];

  print(list1);

  print(list2);

  print(list2.length);

}

* **Langkah 2**

**:** Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

**:** eror dikarenakan pada saat melakukan print list1 belum dideklarasikan

* **Langkah 3**

**:** Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

list1 = [1, 2, null];

print(list1);

var list3 = [0, ...?list1];

print(list3.length);

Apa yang terjadi ? Jika terjadi error, silakan perbaiki.

Tambahkan variabel list berisi NIM Anda menggunakan Spread Operators

: terjadi error dikarenakan terdapat simbol ? didalam list

Perbaikan kode :

void main() {

  var list = [1, 2, 3];

  var list2 = [0, ...list];

  print(list);

  print(list2);

  print(list2.length);

  var list1 = [1, 2, null];

  print(list1);

  var list3 = [0, ...list1];

  print(list3.length);

  // Menambahkan variabel list yang berisi NIM Anda

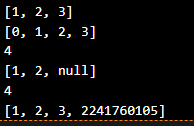
  var nimList = [2241760105];

  var combinedList = [...list, ...nimList];

  print(combinedList);

}

Output :



: Kode ini mengelola beberapa list dengan menggabungkan elemen dari list yang ada menggunakan spread operator, termasuk menambahkan nilai null dan NIM Anda ke dalam list baru.

* **Langkah 4**

**:** Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];

  print(nav);

: error dikarenakan belum mendefinisikan promoActive

Perbaikan kode :

void main() {

  var list = [1, 2, 3];

  var list2 = [0, ...list];

  print(list);

  print(list2);

  print(list2.length);

  var list1 = [1, 2, null];

  print(list1);

  var list3 = [0, ...?list1];

print(list3.length);

  // Menambahkan variabel list yang berisi NIM Anda

  var nimList = [2241760105];

  var combinedList = [...list, ...nimList];

  print(combinedList);

  // Mendefinisikan variabel promoActive

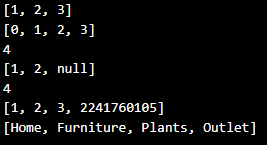
  bool promoActive = true;

  var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];

  print(nav);

}

Output :



: Kode ini mengelola beberapa list dengan berbagai metode, termasuk menggabungkan list menggunakan spread operator, mengizinkan nilai null, dan menambahkan elemen dinamis berdasarkan nilai variabel boolean promoActive

* **Langkah 5**

**:** Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

var nav2 = ['Home', 'Furniture', 'Plants', if (login case 'Manager') 'Inventory'];

  print(nav2);

**:** error dikarenakan belum melakukan deklarasi variable string login

**Perbaikan kode :**

void main() {

  var list = [1, 2, 3];

  var list2 = [0, ...list];

  print(list);

  print(list2);

  print(list2.length);

  var list1 = [1, 2, null];

  print(list1);

  var list3 = [0, ...list1]; // Menggunakan ...? untuk mengizinkan nilai null

  print(list3.length);

  // Menambahkan variabel list yang berisi NIM Anda

  var nimList = [2241760105];

  var combinedList = [...list, ...nimList];

  print(combinedList);

  // Mendefinisikan variabel promoActive

  bool promoActive = true; // Set sesuai kebutuhan, true atau false

  var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];

  print(nav);

  // Mendefinisikan variabel login dengan nilai yang sesuai

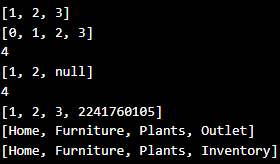
  String login = 'Manager'; // Set sesuai kebutuhan, misalnya 'Manager'

  var nav2 = ['Home', 'Furniture', 'Plants', if (login == 'Manager') 'Inventory'];

  print(nav2);

}

Output :



: Kode ini menggunakan variabel login untuk menyimpan status login dan menambahkan 'Inventory' ke dalam list nav2 jika login sama dengan 'Manager', dengan menggunakan ekspresi login == 'Manager' sebagai kondisi yang valid.

* **Langkah 6**

**:** tambahkan kode progam berikut, lalu coba eksekusi kode anda

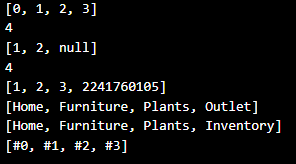
  var listOfInts = [1, 2, 3];

  var listOfStrings = ['#0', for (var i in listOfInts) '#$i'];

  assert(listOfStrings[1] == '#1');

  print(listOfStrings);

output :



Collection For dalam Dart adalah memungkinkan pembuatan dan inisialisasi list atau set secara lebih ringkas dan dinamis dengan menyertakan elemen berdasarkan iterasi dari koleksi lain, memungkinkan penerapan logika dan filter langsung dalam proses pembuatan koleksi baru

**Praktikum 5**

* **Langkah 1**

**:** ketik atau salin kode progam berikut kedalam fungsi main

void main() {

  var record = ('first', a: 2, b: true, 'last');

  print(record);

}

* **Langkah 2**

**:** Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

**:** tidak terjadi error

* **Langkah 3**

**:** Tambahkan kode program berikut di luar scope void main(), lalu coba eksekusi (Run) kode Anda.

(int, int) tukar((int, int) record) {

  var (a, b) = record;

  return (b, a);

}

: error pada kode adalah penggunaan tuple dengan nama field (a: 2, b: true) yang tidak didukung oleh Dart dan penggunaan sintaks tuple yang tidak sesuai dalam konteks fungsi tukar.

Perbaikan kode :

void main() {

  // Menggunakan tuple sederhana tanpa named fields

  var record = (1, 2);

  print('Original record: $record');

  var swapped = tukar(record); // Tidak perlu mengambil elemen berdasarkan indeks

  print('Swapped record: $swapped');

}

// Fungsi tukar untuk tuple (int, int)

(int, int) tukar((int, int) record) {

  var (a, b) = record;

  return (b, a);

}

: Variabel record menyimpan tuple dengan elemen yang sesuai, fungsi tukar menukar dua nilai dalam tuple (int, int), dan tukar((record[0], record[1])) mengambil dua elemen pertama dari record untuk ditukar

* **Langkah 4**

**:** Tambahkan kode program berikut di dalam scope void main(), lalu coba eksekusi (Run) kode Anda.

(String, int) mahasiswa;

  print(mahasiswa);

**:** Variabel mahasiswa Tidak Diinisialisasi: Variabel mahasiswa dideklarasikan dengan tipe (String, int) tetapi tidak diberi nilai awal

**Perbaikan kode :**

: void main() {

  // Menggunakan tuple sederhana tanpa named fields

  var record = (1, 2);

  print('Original record: $record');

  var swapped = tukar(record);

  print('Swapped record: $swapped');

  // Menginisialisasi variabel mahasiswa

  (String, int) mahasiswa = ('Moch Fikri Setiawan', 2241760105);

  print(mahasiswa);

}

// Fungsi tukar untuk tuple (int, int)

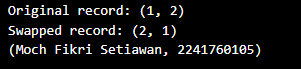
(int, int) tukar((int, int) record) {

  var (a, b) = record;

  return (b, a);

}

Output :



* **Langkah 5**

**:** Tambahkan kode program berikut di dalam scope void main(), lalu coba eksekusi (Run) kode Anda.

var mahasiswa2 = ('first', a: 2, b: true, 'last');

print(mahasiswa2.$1); // Prints 'first'

print(mahasiswa2.a); // Prints 2

print(mahasiswa2.b); // Prints true

print(mahasiswa2.$2); // Prints 'last'

: Kode mengalami error karena Dart tidak mendukung tuple dengan named fields (a dan b) dan mengakses elemen menggunakan sintaks seperti mahasiswa2.$1 dan mahasiswa2.a tidak valid.

**Perbaikan kode :**

void main() {

  // Menggunakan tuple sederhana dengan nama dan NIM Anda

  var record = (2241760105, 1); // Tukar elemen untuk cocok dengan (int, int) yang digunakan di tukar()

  print('Original record: $record');

  var swapped = tukar(record);

  print('Swapped record: $swapped');

  // Menginisialisasi variabel mahasiswa menggunakan kelas

  var mahasiswa = Mahasiswa('Moch Fikri Setiawan', 2241760105);

  print(mahasiswa);

  // Menginisialisasi variabel mahasiswa2 menggunakan kelas dengan named fields

  var mahasiswa2 = MahasiswaNamed('first', 2, true, 'last');

  print(mahasiswa2.first); // Prints 'first'

  print(mahasiswa2.middle); // Prints 2

  print(mahasiswa2.flag); // Prints true

  print(mahasiswa2.last); // Prints 'last'

}

// Fungsi tukar untuk tuple (int, int)

(int, int) tukar((int, int) record) {

  var (a, b) = record;

  return (b, a);

}

// Kelas untuk variabel mahasiswa

class Mahasiswa {

  final String name;

  final int nim;

  Mahasiswa(this.name, this.nim);

  @override

  String toString() => '($name, $nim)';

}

// Kelas untuk variabel mahasiswa2 dengan named fields

class MahasiswaNamed {

  final String first;

  final int middle;

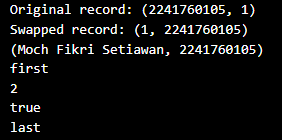
  final bool flag;

  final String last;

  MahasiswaNamed(this.first, this.middle, this.flag, this.last);

}

**Output :**

****

**Tugas praktikum :**

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

: merupakan sebuah blok kode yang memiliki tugas tertentu dan dapat dipangil berkali kali

1. Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!
   * Parameter Positional (Positional Parameters): Parameter yang harus diisi dengan urutan yang benar saat memanggil fungsi.

void main() {

  greet('Alice', 30); // Output: Hello, Alice. You are 30 years old.

}

* + Parameter Named (Named Parameters): Parameter yang didefinisikan dengan nama dan dapat diisi dengan cara yang tidak terikat urutan. Biasanya dideklarasikan dengan kurung kurawal {}.

void main() {

  greet(name: 'Alice', age: 30); // Output: Hello, Alice. You are 30 years old.

}

* + Parameter Optional Positional (Optional Positional Parameters): Parameter yang tidak wajib diisi saat memanggil fungsi dan biasanya didefinisikan dengan tanda kurung siku [ ].

void greet(String name, [int? age]) {

  if (age != null) {

    print('Hello, $name. You are $age years old.');

  } else {

    print('Hello, $name.');

  }

}

* + Parameter Default (Default Parameters): Parameter yang memiliki nilai default yang digunakan jika tidak diberikan nilai saat memanggil fungsi.

void greet(String name, [int age = 0]) {

  print('Hello, $name. You are $age years old.');

}

* + Parameter dengan Fungsi Sebagai Default (Default Function Parameters): Anda bisa memberikan nilai default berupa fungsi.

void performAction(Function action = defaultAction) {

  action();

}

void defaultAction() {

  print('Default action performed.');

}

void customAction() {

  print('Custom action performed.');

}

1. Jelaskan maksud Functions sebagai first-class objects beserta contoh sintaknya!

: Menyimpan Fungsi dalam Variabel: Anda dapat menyimpan fungsi dalam variabel dan kemudian memanggil fungsi tersebut melalui variabel.

void greet() {

  print('Hello!');

}

void main() {

  var myFunction = greet;

  myFunction(); // Output: Hello!

}

Meneruskan Fungsi sebagai Argumen: Fungsi dapat diteruskan sebagai argumen ke fungsi lain.

void performAction(Function action) {

  action();

}

void sayHello() {

  print('Hello!');

}

void main() {

  performAction(sayHello); // Output: Hello!

}

Mengembalikan Fungsi dari Fungsi Lain: Anda dapat mengembalikan fungsi dari fungsi lain.

Function createMultiplier(int multiplier) {

  return (int value) => value \* multiplier;

}

void main() {

  var doubleValue = createMultiplier(2);

  print(doubleValue(5)); // Output: 10

}

Menyimpan Fungsi dalam Struktur Data: Fungsi dapat disimpan dalam struktur data seperti list atau map.

void sayHello() {

  print('Hello!');

}

void sayGoodbye() {

  print('Goodbye!');

}

void main() {

  var functions = [sayHello, sayGoodbye];

  for (var func in functions) {

    func(); // Output: Hello! Goodbye!

  }

}

1. Apa itu Anonymous Functions? Jelaskan dan berikan contohnya!

: Anonymous Functions atau fungsi anonim adalah fungsi yang tidak memiliki nama dan biasanya didefinisikan langsung di tempat penggunaannya. Fungsi ini sering digunakan untuk operasi singkat atau untuk fungsi yang tidak akan digunakan di luar konteks tertentu.

* + Anonymous functions dalam ekspresi

void main() {

  var numbers = [1, 2, 3, 4, 5];

  var doubled = numbers.map((number) => number \* 2);

  print(doubled.toList()); // Output: [2, 4, 6, 8, 10]

}

* + Anonymous Function sebagai Argumen

void performAction(Function action) {

  action();

}

void main() {

  performAction(() {

    print('Action performed!');

  }); // Output: Action performed!

}

* + Anonymous Function yang Diberikan ke Variabel

void main() {

  var greet = () {

    print('Hello, world!');

  };

  greet(); // Output: Hello, world!

}

* + Anonymous Function dalam Collection

void main() {

  var functions = [

    () => print('Function 1'),

    () => print('Function 2'),

    () => print('Function 3')

  ];

  for (var func in functions) {

    func(); // Output: Function 1 Function 2 Function 3

  }

}

1. Jelaskan perbedaan Lexical scope dan Lexical closures! Berikan contohnya!

: Lexical Scope merujuk pada aturan akses variabel berdasarkan lokasi dalam kode, di mana variabel hanya dapat diakses dalam scope tempat mereka dideklarasikan dan scope yang terletak di dalamnya, sedangkan Lexical Closures terjadi ketika fungsi menyimpan referensi ke variabel dari scope tempat fungsi tersebut didefinisikan, memungkinkan fungsi tersebut untuk terus mengakses dan memodifikasi variabel tersebut meskipun sudah keluar dari scope asalnya.

**Lexical scope**

void main() {

  var outerVariable = 'I am outside!';

  void innerFunction() {

    var innerVariable = 'I am inside!';

    print(outerVariable); // Output: I am outside!

    print(innerVariable); // Output: I am inside!

  }

  innerFunction();

  // print(innerVariable); // Error: The variable 'innerVariable' is not in scope

}

**Lexical closures**

void main() {

  var counter = makeCounter(); // 'counter' adalah sebuah closure

  print(counter()); // Output: 1

  print(counter()); // Output: 2

  print(counter()); // Output: 3

}

1. Jelaskan dengan contoh cara membuat return multiple value di Functions!
   * menggunakan tuple

import 'package:tuple/tuple.dart';

Tuple2<int, String> getDetails() {

  return Tuple2(42, 'Hello');

}

void main() {

  var details = getDetails();

  print('Number: ${details.item1}, Message: ${details.item2}');

}

* + menggunakan list atau map

Function makeCounter() {

  var count = 0; // Variabel 'count' dideklarasikan di luar fungsi 'counter'

  return () {

    count += 1;

    return count;

  };

}

List<dynamic> getDetails() {

  return [42, 'Hello'];

}

void main() {

  var details = getDetails();

  print('Number: ${details[0]}, Message: ${details[1]}');

}

Map<String, dynamic> getDetails() {

  return {

    'number': 42,

    'message': 'Hello'

  };

}

void main() {

  var details = getDetails();

  print('Number: ${details['number']}, Message: ${details['message']}');

}

* + menggunakan kelas custom

class Details {

  final int number;

  final String message;

  Details(this.number, this.message);

}

Details getDetails() {

  return Details(42, 'Hello');

}

void main() {

  var details = getDetails();

  print('Number: ${details.number}, Message: ${details.message}');

}

1. Kumpulkan berupa link commit repo GitHub pada tautan yang telah disediakan di grup Telegram!