**Prak 1**

Run



Assert digunakan untuk mengecek sebuah kondisi, dalam length untuk mengecek panjang list array, list[x] == y mengechek apakah index x bernilai y jika bernilai salah maka akan mengembalikan error, print untuk menampilkan output.

Nilai 3 didapat dari print(list.length) yang adalah 3

Nilai 2 didapat dari print(list[1]) yang mengambil nilai dari index 1

Nilai 1 didapat dari print(list[1]) ini terjadi setelah terjadinya list[1] = 1; yang merubah nilai index 1 dari 2 menjadi 1

Perubahan

void main(List<String> arguments) {

  final List<String?> list = List.filled(5, null);

  list[0] = 'Achmad Danni MPA';

  list[1] = '22417600095';

  print(list);

}

Output



**Prak 2**

Run



Var halogens = {…} digunakan untuk menciptakan set variabel bernama halogens, set dalam dart dinotasikan dengan {}

Print digunakan untuk menampilkan output.



Output kosong dikarenakan var nameX belum di isi.

Penambahan add dan addall

  var names1 = <String>{};

  Set<String> names2 = {};

  var names3 = {};

  names1.add('A Danni MPA');

  names2.addAll(['A Danni MPA', '2241760095']);

  print(names1);

  print(names2);

  print(names3);

Output



**Prak 3**

Run



Experimen tipe data maps, maps dapat menyimpan nilai dengan tipe data berbeda.

Penambahan

  var mhs1 = Map<String, String>();

  mhs1['first'] = 'partridge';

  mhs1['second'] = 'turtledoves';

  mhs1['fifth'] = 'golden rings';

  var mhs2 = Map<int, String>();

  mhs2[2] = 'helium';

  mhs2[10] = 'neon';

  mhs2[18] = 'argon';

  print('mhs1: $mhs1');

  print('mhs2: $mhs2');

Output



**Prak 4**

Run langkah 1



List1 memiliki nilai 1,2,3

List2 memiliki nilai 0,1,2,3 hal ini disebabkan oleh inisialisasi 0, …list1 yang mengambil semua nilai didalam list1

4 adalah length list2 yaitu 0(1) ditambah list1(3)

Langkah 3

Error terjadi karena null tidak dapat diassign ke var int perbaikan:

  List<int?> list3 = [1, 2, null];

  print(list3);

  var list4 = [0, ...?list3];

  print(list4.length);

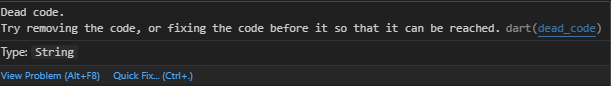
Run



Langkah 4



Output sama namun saat di set false terjadi:



Langkah 5

Jika Manager



Selain Manager



Langkah 6



Collection for digunakan untuk generate element secara dinamis bergantung pada collection lainnya

**Prak 5**

Run



Output keluar yang positional element lalu named element.

Langkah 3



Digunakan untuk menukar posisi

Langkah 4



Bisa digunakan untuk menyimpan 2 data sekaligus

Langkah 5



Pengeluaran menggunakan $ untuk positional, dan nama untuk yang memiliki nama.

**Tugas Praktikum**

1. Sudah
2. Function adalah bagian kode yang dapat digunakan secara berulang yang bertujuan agar kode lebih ringkas dan tidak redundant
3. **Parameter Positional**  
   Parameter yang dikirimkan sesuai urutan

Contoh:

*void greet(String name, int age) {  
 print('Hello, $name! You are $age years old.');  
}*

*greet('Alice', 30);*

P**arameter Positional Optional**

Sama dengan parameter positional cuman jika tidak di inputkan devault ke null

Contoh:

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

*print('Hello, $name!');*

*if (age != null) {*

*print('You are $age years old.');*

*}*

*}*

*greet('Alice');*

**Parameter Bernama**

Parameter yang di identifikasi nama

Contoh:

*void greet({required String name, int age = 30}) {*

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

*}*

*greet(name: 'Alice');*

**Parameter Bernama Optional**

Parameter bernama yang tidak wajib di isi dan akan menjadi null(jika tidak di isi)

Contoh:

*void greet({String? name, int age = 30}) {*

*print('Hello, ${name ?? 'Guest'}! You are $age years old.');*

*}*

*greet();*

1. First class function adalah function yang dapat ditetapkan ke variabel, diteruskan sebagai argumen, dan dikembalikan dari fungsi lain.

Contoh:

*void sayHello() {*

*print('Hello!');*

*}*

*var greet = sayHello;*

*greet();*

*void executeFunction(void Function() func) {*

*func();*

*}*

*executeFunction(sayHello);*

1. Anonymous function adalah function tanpa nama, sering digunakan untuk tugas jangka pendek, terutama sebagai argumen untuk fungsi tingkat tinggi seperti .forEach() atau .map().

Contoh:

*var multiply = (int a, int b) => a \* b;*

*print(multiply(3, 4)); // Outputs: 12*

*var list = [1, 2, 3];*

*list.forEach((element) {*

*print('Element: $element');*

*});*

1. Lexical Scope: Merujuk pada visibilitas variabel yang didefinisikan dalam suatu fungsi, blok, atau modul, tergantung di mana variabel tersebut didefinisikan dalam kode sumber.

Contoh:

*void main() {*

*int outer = 10;*

*void inner() {*

*print(outer);*

*}*

*inner();*

*}*

Lexical Closure: Suatu fungsi yang menangkap variabel dari lingkup leksikal di sekitarnya, bahkan setelah lingkup tersebut selesai dieksekusi.

Contoh:

*void main() {*

*Function makeAdder(int addBy) {*

*return (int i) => i + addBy;*

*}*

*var add5 = makeAdder(5);*

*print(add5(2));*

*}*

1. Di Dart, Anda dapat menggunakan tupel (catatan) atau koleksi seperti daftar dan peta untuk mengembalikan beberapa nilai dari suatu fungsi.

Contoh list:

*List<int> getCoordinates() {*

*return [10, 20];*

*}*

*void main() {*

*var coordinates = getCoordinates();*

*print('X: ${coordinates[0]}, Y: ${coordinates[1]}');*

*}*

1. [GitHub](https://github.com/AchmadDanniMPA/A-Danni-MPA_01_Mobile)