

# **POLITEKNIK NEGERI MALANG**

***“Pemrograman Mobile ”***

***D4 – Teknik Informatika***

Dosen pengampu: Sofyan Noor Arief, S.ST., M.Kom.



**Disusun oleh:**

**Nama: Mukhamad Faruq Al Fahmi**

**Kelas: 3D**

**Absen: 25**

**Nim: 2141720066**

## Prtaktikum 1: Eksperimen tipe data list

### a. Hasil Run Langkah 2

```
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum1.dart
3
2
1
```

kode pada praktikum 1 ini adalah contoh sederhana penggunaan List dalam Dart, serta penggunaan assert untuk memeriksa kondisi yang seharusnya benar selama eksekusi program. Jika ada yang tidak sesuai, program akan memunculkan pesan kesalahan.

### b. Kode program

```
final finallist = List.filled(5, null);

finallist[1] = 'Nama: Mukhamad Faruq Al Fahmi';
finallist[1] = 'NIM: 2141720066';
assert(finallist[1] == 'Nama: Mukhamad Faruq Al Fahmi');
assert(finallist[2] == "NIM: 2141720066");

print(finallist[1]);
print(finallist[2]);
```

### c. kode di atas terjadi eror karena data null tidak bisa di isi dengan tipe data string

### d. Pembeneran

```
final finallist = List<String?>.filled(6, null);

finallist[1] = 'Nama: Mukhamad Faruq Al Fahmi';
finallist[2] = 'NIM: 2141720066';
assert(finallist[1] == 'Nama: Mukhamad Faruq Al Fahmi');
assert(finallist[2] == "NIM: 2141720066");

print(finallist[1]);
print(finallist[2]);
print(finallist[5]);
```

### e. Hasil

```
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum1.dart
Nama: Mukhamad Faruq Al Fahmi
NIM: 2141720066
null
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4>
```

## Praktikum 2: Eksperimen Tipe Data Set

### a. Langkah 1

```
void main() {  
  var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine',  
                  'astatine'};  
  
  print(halogens);  
}
```

#### Output

```
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum2.dart  
{fluorine, chlorine, bromine, iodine, astatine}
```

Pada kode di atas membuat set bernama halogens kemudian mengisi variabel tersebut dan menampilkan hasilnya

### b. Langkah 3

```
var names1 = <String>{};  
Set<String> names2 = {}; // This works, too.  
var names3 = {}; // Creates a map, not a set.  
  
print(names1);  
print(names2);  
print(names3);
```

kode di atas tidak terjadi eror namun hanya menampilkan kurung kurawal

### c. Tambahkan elemen nama dan NIM Anda pada kedua variabel Set tersebut dengan dua fungsi berbeda yaitu `.add()` dan `.addAll()`.

```
var names1 = <String>{};  
Set<String> names2 = {}; // This works, too.  
//var names3 = {}; // Creates a map, not a set.  
names1.add('Mukhamad Faruq Al Fahmi');  
names1.add('2141720066');  
  
names2.addAll(names1);  
  
print(names1);  
print(names2);
```

#### Hasil

```
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum2.dart  
{fluorine, chlorine, bromine, iodine, astatine}  
{Mukhamad Faruq Al Fahmi, 2141720066}  
{Mukhamad Faruq Al Fahmi, 2141720066}
```

### Paktikum 3: Eksperimen Tipe Data Maps

a. Langkah 1

```
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum3.dart  
{first: partridge, second: turtledoves, fifth: 1}  
{2: helium, 10: neon, 18: 2}
```

Dari kode yang di salin diaman amembuat variabel yang merupakan map dan setiap elemen map terdapat sepasang kunci (key) dan value (nilai)

b. Langkah 3

- a. Penjelasan: kode pada langkah ini tidak eror namun saat di run nilai akan tetap seperti langkah 1 karena pada langkah ini variabel 1 dan 2 tidak di beri nilai
- b. Perbaikan

```
// perbaikan  
var gifts = {  
  // Key:    Value  
  'first': 'partridge',  
  'second': 'turtledoves',  
  'fifth': 1,  
  'nama': 'Mukhamad Faruq Al Fahmi',  
  'NIM': 2141720066  
};  
  
var nobleGases = {  
  2: 'helium',  
  10: 'neon',  
  18: 2,  
  'nama': 'Mukhamad Faruq Al Fahmi',  
  'NIM': 2141720066  
};  
  
var mhs1 = Map<String, String>();  
gifts['first'] = 'partridge';  
gifts['second'] = 'turtledoves';  
gifts['fifth'] = 'golden rings';  
gifts['nama'] = 'Mukhamad Faruq Al Fahmi';  
gifts['NIM'] = '2141720066';  
  
var mhs2 = Map<int, String>();  
nobleGases[2] = 'helium';  
nobleGases[10] = 'neon';  
nobleGases[18] = 'argon';  
nobleGases['nama'] = 'Mukhamad Faruq Al Fahmi';  
nobleGases['NIM'] = '2141720066';
```

```
print('/n');
print("gifts: $gifts\n");
print("nobleGases: $nobleGases\n");
print("Mahasiswa 1: $mhs1\n");
print("Mahasiswa 2: $mhs2\n");
```

#### Praktikum 4: Eksperimen Tipe Data List: Spread dan Control-flow Operators

a. Langkah 1

Langkah 1 menyalin kode program

```
var list = [1, 2, 3];
var list2 = [0, ...list];
print(list1);
print(list2);
print(list2.length);
```

b. Langkah 2

Kode program di atas terdapat eror dimana nama pada 'print(list1);' tidak terdefinisi sehingga terjadi eror, seharusnya di ganti menjadi "print(list)"

Hasil:

```
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum4.dart
[1, 2, 3]
[0, 1, 2, 3]
4
```

Kode tersebut merupakan contoh penggunaan operator spread( tanda titik tiga (...))

list2 mendapatkan nilai dari list1 karena nilai list1 telah digabungkan dengan nilai list2 dengan bantuan operator spread

c. Langkah 3

```
list1 = [1, 2, null];
print(list1);
var list3 = [0, ...?list1];
print(list3.length);
```

Pada praktikum ini terjadi eroro karena terdapat beberapa kesalahan dimana seharusnya " list1 = [1, 2, null];" dibuat menjadi variabel dan yang kedua menghilangkan tanda tanya pada "var list3 = [0, ...?list1];"

- Perbaikan dan penambahan Nama dan NIM

```
var list1 = [1, 2, null];
print(list1);

var list3 = [0, ...list1];
print(list3.length);

var NIM = [2141720066];
```

```
var list4 = ["Mukhamad Faruq Al Fahmi", ...NIM];
print(list4);
}
```

- Output

```
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum4.dart
[1, 2, null]
4
[Mukhamad Faruq Al Fahmi, 2141720066]
```

d. Langkah 4

- Perbaiki kode

```
print("promoActive = true\n");
bool promoActive = true;
print('Promo Active: $promoActive');
var nav = ['Home', 'Furniture', 'Plants', if (promoActive)
'Outlet'];
print(nav);

print("\npromoActive2 = false\n");
bool promoActive2 = false;
print('Promo Active: $promoActive2');
// ignore: dead_code
var nav2 = ['Home', 'Furniture', 'Plants', if (promoActive2)
'Outlet'];
print(nav2);
```

Hasil

```
Promo Active: true
[Home, Furniture, Plants, Outlet]

promoActive2 = false

Promo Active: false
[Home, Furniture, Plants]
```

e. Langkah 5

Pada tahap ini terjadi eror karena perlu di tambahkan case dengan variabel login  
Hasil Ketika sudah di benarkan

```
Case True
[Home, Furniture, Plants, Inventory]

Case false/kondisi selain manager
[Home, Furniture, Plants]
```

f. Langkah 6

Hasil

```
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum4.dart  
[#0, #1, #2, #3]
```

Penjelasan

Kode pada Langkah 6 menampilkan nilai pada variabel listOfInts menambahkan #0 sebagai nilai awal dan menambahkan # pada nilai lainnya

Manfaat Collection

Sebagai klarifikasi, jika Anda merujuk kepada "collection for" dalam Dart, mungkin Anda berbicara tentang fitur "collection if" dan "collection for" yang memungkinkan Anda membuat koleksi (List, Set, atau Map) dengan kondisi atau iterasi tertentu.

## Praktikum 5: Eksperimen Tipe Data Records

a. Langkah 1

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

b. Langkah 2

```
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum5.dart  
(first, last, a: 2, b: true)
```

tipe data records akan mencetak nilai/elemen yang bernilai String terlebih dahulu, tipe data selain String akan dicetak sesuai urutan dalam inisialisasi

c. Langkah 3

Kode:

```
void main() {  
  var records = (30, 50);  
  print("Record awal: $records");  
  
  var hasilTukar = tukar(records);  
  print("Record setelah ditukar: $hasilTukar");  
}  
  
(int, int) tukar((int, int) record) {  
  var (a, b) = record;  
  return (b, a);  
}
```

Hasil:

```
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum5.dart  
Record awal: (30, 50)  
Record setelah ditukar: (50, 30)
```

d. Langkah 4

```
(String, int) mahasiswa;  
print(mahasiswa);
```

dari kode di atas terdapat eror karena var mahasiswa belum di inisialisasi nilainya

perbaikan

```
(String, int) mahasiswa;  
mahasiswa = ("Mukhamad Faruq Al Fahmi", 2141720066); //  
inisialisasi nilai dengan nama dan NIM  
print(mahasiswa);
```

hasil

```
Record Setelah ditukar : (30, 30)  
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum5.dart  
(Mukhamad Faruq Al Fahmi, 2141720066)
```

e. Langkah 5

Kode

```
var mahasiswa2 = ('first', a: 2141720066, b: true, 'Mukhamad  
Faruq Al Fahmi');  
  
print(mahasiswa2.$1); // Prints 'first'  
print(mahasiswa2.a); // Prints 2  
print(mahasiswa2.b); // Prints true  
print(mahasiswa2.$2); // Prints 'Last'
```

hasil

```
PS D:\Code Learn\Flutter\Code Flutter\jobsheet_4> dart praktikum5.dart  
first  
2141720066  
true  
Mukhamad Faruq Al Fahmi
```



## 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!
3. Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!
4. Jelaskan maksud Functions sebagai first-class objects beserta contoh sintaknya!
5. Apa itu Anonymous Functions? Jelaskan dan berikan contohnya!
6. Jelaskan perbedaan Lexical scope dan Lexical closures! Berikan contohnya!
7. Jelaskan dengan contoh cara membuat return multiple value di Functions!
8. Kumpulkan berupa link commit repo GitHub pada tautan yang telah disediakan di grup Telegram!

## JAWAB:

2. Dalam bahasa Dart, fungsi (functions) adalah blok kode yang memiliki nama, menerima parameter (atau argumen), dan mengembalikan nilai. Fungsi dapat digunakan untuk mengelompokkan dan mengorganisasi logika program menjadi bagian-bagian yang dapat digunakan kembali.

3. Jenis-jenis parameter di function

1. Required Parameters:

```
void printMessage(String message) {  
  print(message);  
}
```

2. Optional Parameters:

a. Named Parameters:

```
void printDetails({String name, int age}) {  
  print('Name: $name, Age: $age');  
}
```

b. Positional Parameters:

```
void printDetails(String name, [int age]) {  
  print('Name: $name, Age: $age');  
}
```

3. Default Parameters:

```
void printDefaultDetails(String name, {int age = 30}) {  
  print('Name: $name, Age: $age');  
}
```

#### 4. Functions sebagai First-Class Objects:

Dalam Dart, fungsi dianggap sebagai first-class objects, yang berarti fungsi dapat:

- Disimpan dalam variabel.
- Dilewatkan sebagai parameter ke fungsi lain.
- Dikembalikan sebagai nilai dari fungsi lain.

Contoh sintak

```
void greet() {  
  print('Hello!');  
}  
  
void main() {  
  var myFunction = greet;  
  myFunction(); // Memanggil fungsi melalui variabel.  
}
```

#### 5. Anonymous Functions:

Anonymous functions (fungsi tanpa nama) disebut juga lambda atau closures. Mereka dapat digunakan untuk membuat fungsi yang sederhana dan langsung digunakan tanpa perlu mendefinisikan nama.

Contoh sintak

```
void main() {  
  var multiply = (int a, int b) {  
    return a * b;  
  };  
  
  print(multiply(2, 3)); // Output: 6  
}
```

#### 6. Lexical Scope dan Lexical Closures:

##### Lexical Scope:

Lexical scope berarti variabel-variabel di dalam suatu fungsi dapat diakses oleh fungsi tersebut dan oleh fungsi-fungsi yang didefinisikan di dalamnya.

Contoh sintak

```
void outerFunction() {  
  var outerVariable = 'I am from outer function';  
  
  void innerFunction() {  
    print(outerVariable); // Bisa mengakses variabel dari outer function.  
  }  
  
  innerFunction();  
}
```

### Lexical Closures:

Closures adalah fungsi-fungsi yang dapat mengakses variabel-variabel dari lingkungan di mana mereka dibuat, bahkan jika fungsi tersebut dijalankan di lingkungan yang berbeda.

Contoh sintak

```
Function outerFunction() {  
  var outerVariable = 'I am from outer function';  
  
  return () {  
    print(outerVariable); // Closure, dapat mengakses variabel dari outer function.  
  };  
}
```

### 7. Return Multiple Values di Functions:

Dart tidak secara langsung mendukung pengembalian multiple values dari fungsi, tetapi Anda dapat menggunakan Map atau List untuk mencapainya.

Contoh dengan Map:

```
Map<String, dynamic> getPersonInfo() {  
  return {'name': 'John', 'age': 25};  
}  
  
void main() {  
  var personInfo = getPersonInfo();  
  print(personInfo['name']); // Output: John  
  print(personInfo['age']); // Output: 25  
}
```

Contoh dengan List:

```
List<dynamic> getPersonInfo() {  
  return ['John', 25];  
}  
  
void main() {  
  var personInfo = getPersonInfo();  
  print(personInfo[0]); // Output: John  
  print(personInfo[1]); // Output: 25  
}
```