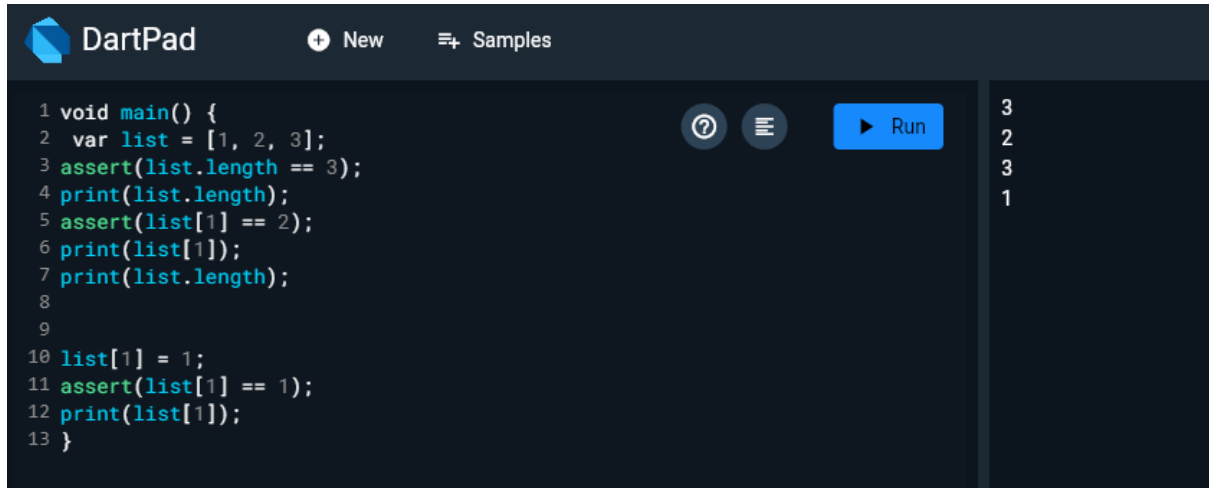


## TUGAS MOBILE PROGRAMMING

(Pemograman Dasar Dart 3)

RAFIKAH NADHIF MAULIDINA – 230444040025

### PRAKTIKUM 1 : EKSPERIMEN TYPE DATA LIST



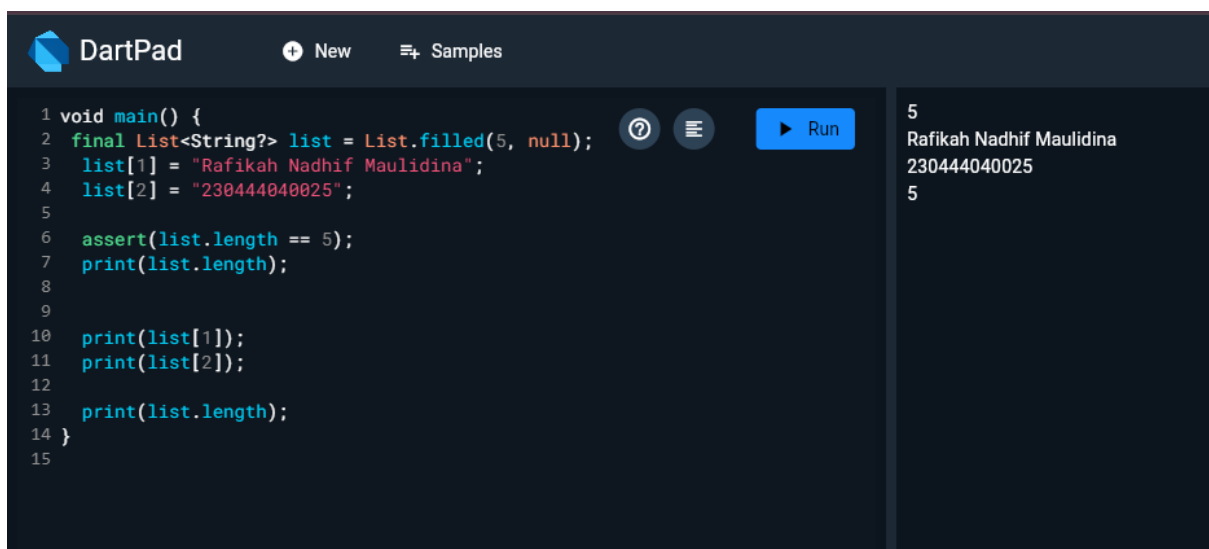
```
1 void main() {
2   var list = [1, 2, 3];
3   assert(list.length == 3);
4   print(list.length);
5   assert(list[1] == 2);
6   print(list[1]);
7   print(list.length);
8
9
10  list[1] = 1;
11  assert(list[1] == 1);
12  print(list[1]);
13 }
```

3  
2  
3  
1

Yang terjadi adalah

1. Var list adalah tipe data list yang berisi 3 elemen [1, 2, 3]
2. Mengecek Panjang llist dengan asset yang dimana Panjang nya 3
3. list[1] adalah 2, sehingga tidak error saat hasil keluar
4. elemen ke 1 diubah dari 2 menjadi 1, yang dimana print nya sesuai jadi tidak terjadi error

ubah Langkah 1 menjadi variable final



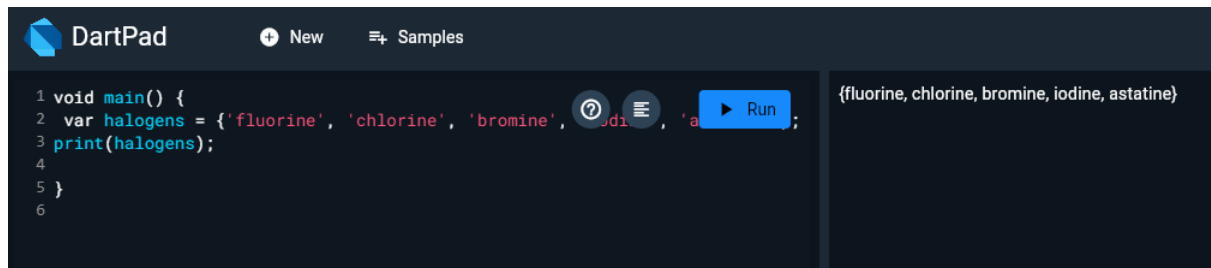
```
1 void main() {
2   final List<String?> list = List.filled(5, null);
3   list[1] = "Rafikah Nadhif Maulidina";
4   list[2] = "230444040025";
5
6   assert(list.length == 5);
7   print(list.length);
8
9
10  print(list[1]);
11  print(list[2]);
12
13  print(list.length);
14 }
15
```

5  
Rafikah Nadhif Maulidina  
230444040025  
5

1. list tipe data numerik, tidak bisa diisi dengan huruf
2. final memodifikasi isi list

3. `string(?)` untuk mendukung nilai null
4. `List.filled(5, null)` list tetap ukurannya 5, setiap elemennya awalan null

## PRAKTIKUM 2 : EKSPERIMEN TIPE DATA SET



```

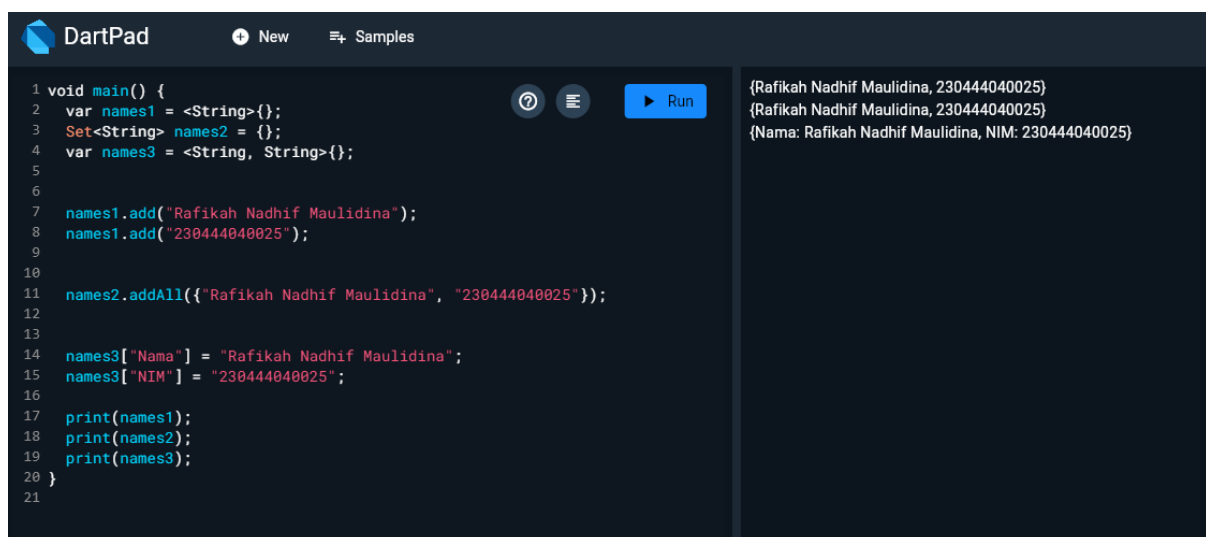
1 void main() {
2   var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};
3   print(halogens);
4 }
5
6

```

{fluorine, chlorine, bromine, iodine, astatine}

Yang terjadi adalah:

1. Halogens adalah set dalam dart yang menggunakan {}
2. yang dimana elemennya tidak berulang
3. elemennya tidak memiliki indeks seperti list
4. data nya tidak selalu menjamin berurutan
5. kalau mau berurutan pakainya list [] bukan set{}



```

1 void main() {
2   var names1 = <String>{};
3   Set<String> names2 = {};
4   var names3 = <String, String>{};
5
6   names1.add("Rafikah Nadhif Maulidina");
7   names1.add("230444040025");
8
9   names2.addAll({"Rafikah Nadhif Maulidina", "230444040025"});
10
11   names3["Nama"] = "Rafikah Nadhif Maulidina";
12   names3["NIM"] = "230444040025";
13
14   print(names1);
15   print(names2);
16   print(names3);
17 }
18
19
20
21

```

{Rafikah Nadhif Maulidina, 230444040025}

{Rafikah Nadhif Maulidina, 230444040025}

{Nama: Rafikah Nadhif Maulidina, NIM: 230444040025}

Yang terjadi adalah :

1. `var names3` bertipe `<String, String>{};` bukan set
2. menambahkan elemen Add pada `names1` dan `addAll` pada `names2`
3. `addAll` digunakan untuk menambah beberapa elemen sekaligus ke dalam set
4. `names3` map, jadi pakai key-value

### PRAKTIKUM 3 : EKSPERIMEN TIPE DATA MAPS

```
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 }
19
```

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

Yang terjadi adalah :

1. gifts merupakan tipe map<string, dynamic>
2. key nya berupa string dan value nya ada yang string ada yang integer
3. key digunakan untuk mengakses value
4. nobleGases merupakan tipe map<int, dynamic>
5. key nya berupa int dan value nya berupa string dan int
6. print gifts dan noblegases untuk menampilkan output

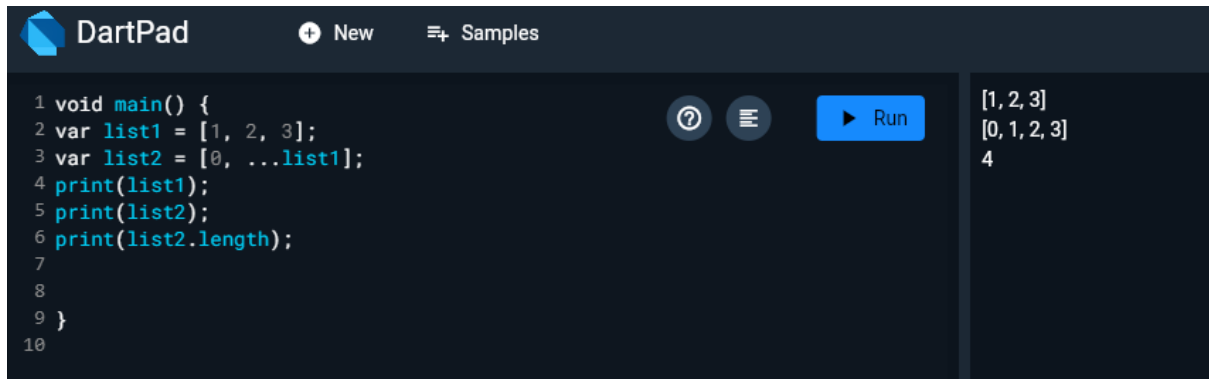
```
7 };
8
9 var nobleGases = {
10  2: 'helium',
11  10: 'neon',
12  18: 2,
13 };
14 var mhs1 = Map<String, String>();
15 gifts['first'] = 'partridge';
16 gifts['second'] = 'turtledoves';
17 gifts['fifth'] = 'golden rings';
18
19
20 var mhs2 = Map<int, String>();
21 nobleGases[2] = 'helium';
22 nobleGases[10] = 'neon';
23 nobleGases[18] = 'argon';
24
25 print(gifts);
26 print(nobleGases);
27 print(mhs1);
28 print(mhs2);
29
30 }
31
```

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

Yang dihasilkan adalah :

1. menambahkan mhs 1 (key nya string, value nya string) dan mhs 2 (key nya string, valuenya int)
2. mengubah nilai di gifts sebelum nya int, sekarang jadi 'golden rings' (string)
3. mengubah nilai nobleGases, sebelum nya 18 punya nilai 2, sekarang diubah menjadi 'argon'

## PRAKTIKUM 4 : EKSPERIMEN TIPE DATA LIST (SPREAD & CONTROL-FLOW OPERATORS)



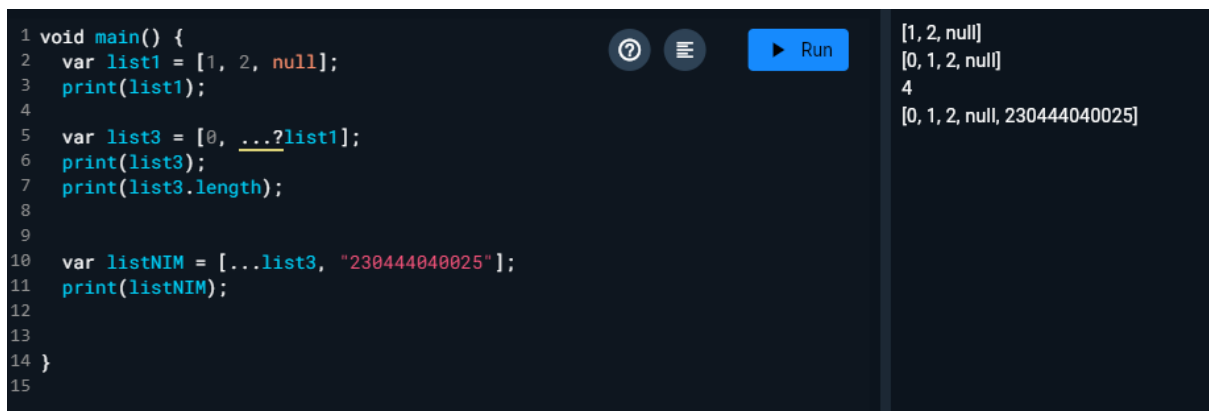
```
1 void main() {
2   var list1 = [1, 2, 3];
3   var list2 = [0, ...list1];
4   print(list1);
5   print(list2);
6   print(list2.length);
7
8
9 }
10
```

Output:

```
[1, 2, 3]
[0, 1, 2, 3]
4
```

Yang terjadi adalah :

1. di soal list1 akan error karna tidak ada yang dipanggil, maka saya betulkan menjadi list1
2. list1 sederhana yang berisi 3 angka
3. (...list1) spread operator untuk menyalin list1 ke list2, dimulai dari 0 lalu dilanjutkan list1
4. list2.length akan menghitung Panjang output yang dihasilkan



```
1 void main() {
2   var list1 = [1, 2, null];
3   print(list1);
4
5   var list3 = [0, ...?list1];
6   print(list3);
7   print(list3.length);
8
9
10  var listNIM = [...list3, "230444040025"];
11  print(listNIM);
12
13
14 }
15
```

Output:

```
[1, 2, null]
[0, 1, 2, null]
4
[0, 1, 2, null, 230444040025]
```

Yang terjadi adalah

1. list1 berisi null
2. ?list1, jika list1 tidak null maka isi list1 akan disebarkan ke list3
3. Jika list1 null maka akan dilewati
4. Var listNIM menggunakan spread operator untuk menambahkan semua elemen dari list3

```
1 void main() {
2   bool promoActive = true;
3
4   var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
5   print(nav);
6 }
7
```

[Home, Furniture, Plants, Outlet]

Yang terjadi adalah

1. Jika true maka outlet akan ditambahkan ke dalam list
2. Jika false maka outlet tidak dimasukkan ke list
3. Gambar diatas menunjukkan output nya true
4. Gambar dibawah menunjukkan outputnya false

```
1 void main() {
2   bool promoActive = false;
3   var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
4
5   print(nav);
6 }
7
8
```

[Home, Furniture, Plants]

Login supervisor

```
1 void main() {
2   String login = 'Supervisor';
3   var nav2 = ['Home', 'Furniture', 'Plants', if (login == 'Supervisor')
4     'Inventory'];
5   print(nav2);
6 }
```

[Home, Furniture, Plants, Inventory]

Yang ditampilkan adalah :

1. Variable login, jika login == supervisor maka inventory akan masuk ke nav2

Login admin

```
1 void main() {
2   String login = 'Admin';
3   var nav2 = ['Home', 'Furniture', 'Plants', if (login == 'Supervisor')
4     'Inventory'];
5   print(nav2);
6 }
```

[Home, Furniture, Plants]

2. jika login admin maka inventory tidak masuk

```
1 void main() {
2   var listofInts = [1, 2, 3];
3   var listofStrings = ['#0', for (var i in listofInts) '#$i'];
4   assert(listofStrings[1] == '#1');
5   print(listofStrings);
6 }
7
8
```

[#0, #1, #2, #3]

Yang dihasilkan :

1. Var lisofints berisi angka 1, 2 , 3

2. Menggunakan collection for dalam var listofstrings yang dimana #0 sebagai elemen pertama dan di looping melalui listofints dan menambahkan elemen baru
3. Assert listofstring memastikan elemen ke 1 adalah 1, jika bukan akan terjadi error

## MANFAAT COLLECTION FOR

Memanfaatkan perulangan untuk deklarasi langsung di dalam list

1. Kode lebih ringkas untuk dibaca
2. Lebih singkat dan mudah dipahami
3. Meningkatkan performa karna bisa menambahkan elemen ke list tanpa menggunakan add()

## PRAKTIKUM 5 : EKSPERIMEN TIPE DATA RECORDS

```
1 void main() {
2     var record = ('first', a: 2, b: true, 'last');
3     print(record);
4 }
5
```

(first, last, a: 2, b: true)

Yang dihasilkan :

1. Records bisa berisi nilai yang memiliki nama / tanpa nama
2. Record tetap mempertahankan urutan field

```
1 (int, int) tukar((int, int) record) {
2     var (a, b) = record;
3     return (b, a);
4 }
5
6 void main() {
7     var angka = (10, 20);
8     print("Sebelum ditukar: $angka");
9
10    var hasil = tukar(angka);
11    print("Setelah ditukar: $hasil");
12 }
13
```

Sebelum ditukar: (10, 20)  
Setelah ditukar: (20, 10)

Yang terjadi adalah

1. Fungsi tukar, menerima records dengan (int, int) dan mengembalikan record baru dengan nilai yang sudah ditukar
2. Didalam main, var angka adalah record awal sebelum ditukar
3. Sedangkan var hasil adalah record akhir setelah ditukar

```
1 void main() {
2
3   (String, int) mahasiswa = ("Rafikah Nadhif Maulidina", 230444040025);
4   print(mahasiswa);
5   print("Nama: ${mahasiswa.$1}");
6   print("NIM: ${mahasiswa.$2}");
7 }
8
```

(Rafikah Nadhif Maulidina, 230444040025)  
Nama: Rafikah Nadhif Maulidina  
NIM: 230444040025

Yang terjadi adalah

1. Deklarasi kode yang dimana mahasiswa mempunyai record dengan 2 fields, nama (string), nim (int)
2. Inisialisasi record menyimpan data nama dan nim
3. Print mahasiswa mencetak semuanya
4. Print nama dan nim mengakses field pertama(nama) dan kedua(nim)

```
1 void main() {
2   (String, int) mahasiswa = ("Rafikah Nadhif Maulidina", 230444040025);
3   print(mahasiswa);
4   print("Nama: ${mahasiswa.$1}");
5   print("NIM: ${mahasiswa.$2}");
6
7   var mahasiswa2 = ('first', a: 2, b: true, 'last');
8
9
10
11  print(mahasiswa2.$1);
12  print(mahasiswa2.a);
13  print(mahasiswa2.b);
14  print(mahasiswa2.$2);
15 }
16
```

(Rafikah Nadhif Maulidina, 230444040025)  
Nama: Rafikah Nadhif Maulidina  
NIM: 230444040025  
first  
2  
true  
last

Yang dihasilkan :

1. Record mahasiswa2 dengan positional fields dan named
2. Positional \$1 = first & \$2 = last
3. Named a = 2, b = true

---

### 1. Function dalam Bahasa Dart :

Blok kode untuk melakukan perintah tertentu dan bisa dipanggil di parameter tertentu, tentunya bisa menghindari duplikat kode

### 2. Jenis jenis parameter function dalam Bahasa dart :

- positional parameters : sesuai urutan Ketika function dipanggil
- optional position parameters : menggunakan []
- named parameter : menggunakan {} pemanggilan function dengan parameter yang tidak harus berurutan
- default parameters : menggunakan nilai default jika tidak ada parameters

### 3. Maksud function sebagai first-class object dan contoh sintaksnya :

- disimpan dalam variable
- dikirim ke function sebagai parameter lain
- dikembalikan sebagai function

Contoh :

```
1 void sayHello() {
2   print("Hello, Rafikah!");
3 }
4 void main() {
5   var greet = sayHello;
6   greet();
7 }
8
9
```

Run

Hello, Rafikah!

#### 4. Anonymous function dan contohnya :

Function tanpa nama sebagai parameter sekali pakai

Contoh :

```
1 void main() {
2   var numbers = [1, 2, 3, 4];
3
4   numbers.forEach((number) {
5     print("Number: $number");
6   });
7 }
8
```

Run

Number: 1  
Number: 2  
Number: 3  
Number: 4

#### 5. Perbedaan lexical scope dan lexical closures dan contohnya :

-lexical scope : hanya bisa mengakses variable dalam cakupan yang dimana function itu di deklarasikan

Contoh lexical scope

```
1 void main() {
2   int x = 10;
3
4   void printX() {
5     print(x);
6   }
7
8   printX();
9 }
10
```

Run

10

-lexical closure : function yang menggunakan variable dari luar cakupannya tapi masiih bisa mengakses meskipun scope aslinya tidak ada

Contoh lexical closure

```
1 Function makeMultiplier(int multiplier) {
2   return (int number) => number * multiplier;
3 }
4
5 void main() {
6   var timesTwo = makeMultiplier(2);
7   print(timesTwo(5));
8 }
9
```

Run

10

#### 6. Contoh membuat return multiple value di function :

-return value list



```
1 List<int> getCoordinates() {
2     return [20, 30];
3 }
4
5 void main() {
6     var coords = getCoordinates();
7     print("X: ${coords[0]}, Y: ${coords[1]}");
8 }
9
```

X: 20, Y: 30

-return value map

```
1 Map<String, dynamic> getUser() {
2     return {"name": "Rafikah", "age": 19};
3 }
4
5 void main() {
6     var user = getUser();
7     print("Name: ${user['name']}, Age: ${user['age']}");
8 }
9
```

Name: Rafikah, Age: 19

-return value records

```
1 (String, int) getPerson() {
2     return ("Rafikah", 19);
3 }
4
5 void main() {
6     var (name, age) = getPerson();
7     print("Name: $name, Age: $age");
8 }
9
```

Name: Rafikah, Age: 19