

Mobile Programming



Malik Abdul Azis

2241720240

3I Class

13


STUDY PROGRAM D-IV INFORMATIC ENGINEER
INFORMATION TECHNOLOGY DEPARTMENT
MALANG STATE POLYTECHNIC

Soekarno Hatta Street No.9, Jatimulyo, Lowokwaru District, Malang City, East Java

65141

Practicum 1

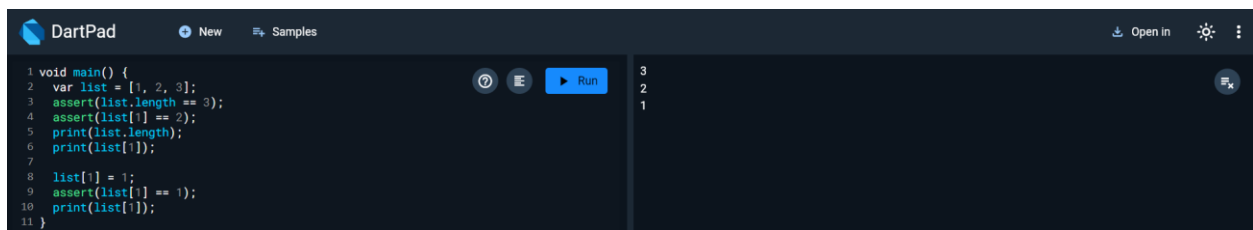
- Langkah 1



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

- Langkah 2

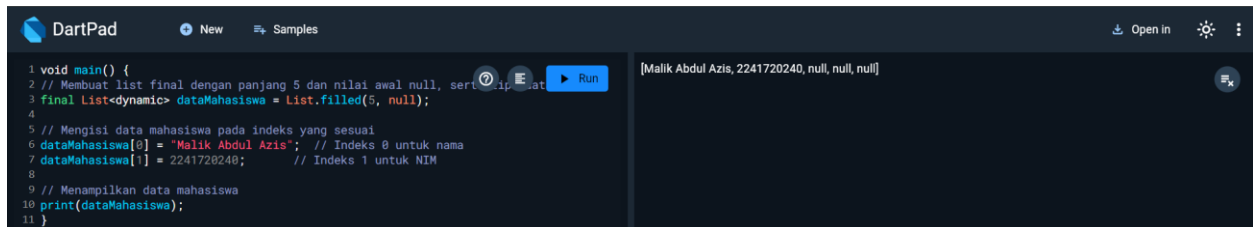
Ini menunjukkan bahwa panjang list adalah 3, elemen ke-1 awalnya bernilai 2, kemudian diubah menjadi 1, dan nilai terakhir yang dicetak adalah 1.



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

3
2
1

- Langkah 3

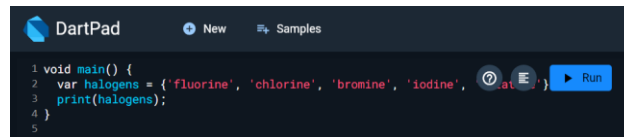


```
1 void main() {  
2   // Membuat list final dengan panjang 5 dan nilai awal null, sertipipat  
3   final List<dynamic> dataMahasiswa = List.filled(5, null);  
4  
5   // Mengisi data mahasiswa pada indeks yang sesuai  
6   dataMahasiswa[0] = "Malik Abdul Azis"; // Indeks 0 untuk nama  
7   dataMahasiswa[1] = 2241720240; // Indeks 1 untuk NIM  
8  
9   // Menampilkan data mahasiswa  
10  print(dataMahasiswa);  
11 }
```

[Malik Abdul Azis, 2241720240, null, null, null]

Practicum 2

- Langkah 1

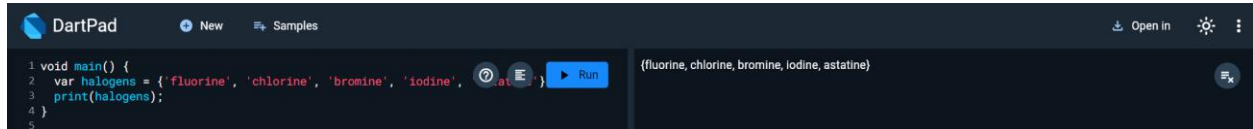


The screenshot shows the DartPad web interface. The code editor contains the following Dart code:

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

Below the code editor, there are icons for running the code (a play button) and a 'Run' button. The output area on the right is empty.

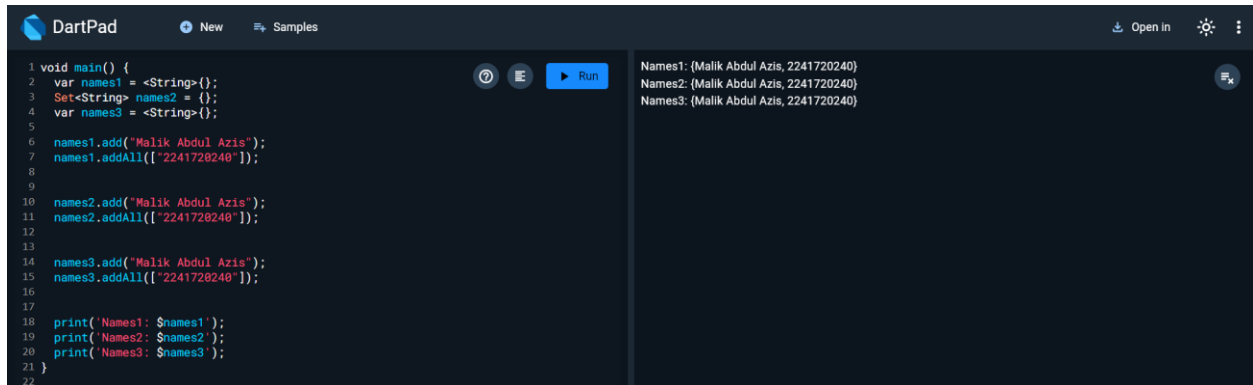
- Langkah 2



The screenshot shows the DartPad web interface with the same code as in Langkah 1. The output area on the right now displays the result of the program execution:

```
(fluorine, chlorine, bromine, iodine, astatine)
```

- Langkah 3



The screenshot shows the DartPad web interface with a more complex Dart program. The code editor contains the following Dart code:

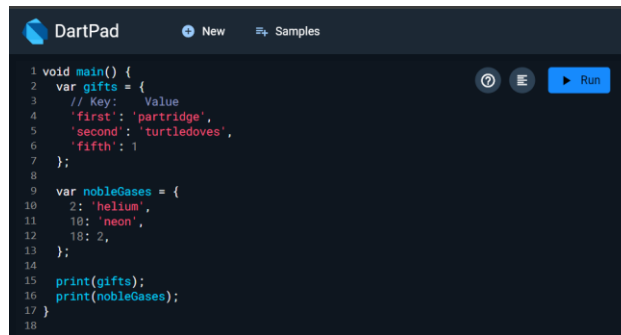
```
1 void main() {  
2   var names1 = <String>{};  
3   Set<String> names2 = {};  
4   var names3 = <String>{};  
5  
6   names1.add("Malik Abdul Azis");  
7   names1.addAll(["2241720240"]);  
8  
9  
10  names2.add("Malik Abdul Azis");  
11  names2.addAll(["2241720240"]);  
12  
13  
14  names3.add("Malik Abdul Azis");  
15  names3.addAll(["2241720240"]);  
16  
17  
18  print('Names1: $names1');  
19  print('Names2: $names2');  
20  print('Names3: $names3');  
21 }  
22
```

Below the code editor, there are icons for running the code (a play button) and a 'Run' button. The output area on the right now displays the result of the program execution:

```
Names1: {Malik Abdul Azis, 2241720240}  
Names2: {Malik Abdul Azis, 2241720240}  
Names3: {Malik Abdul Azis, 2241720240}
```

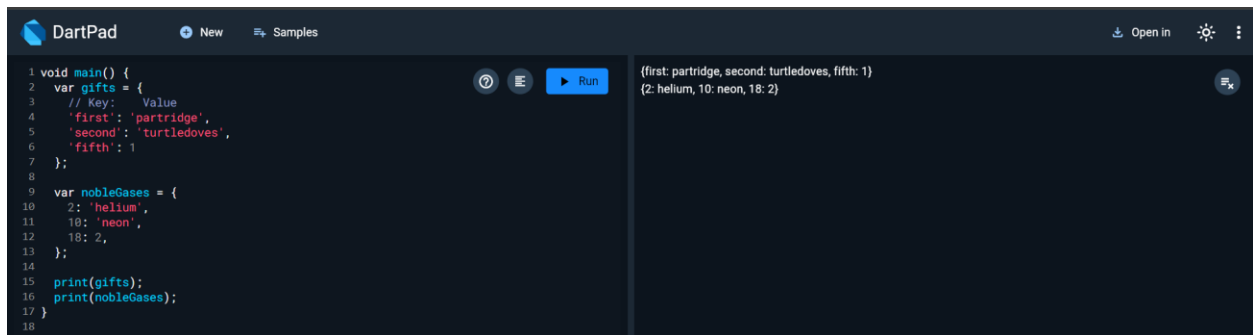
Practicum 3

- Langkah 1



```
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
```

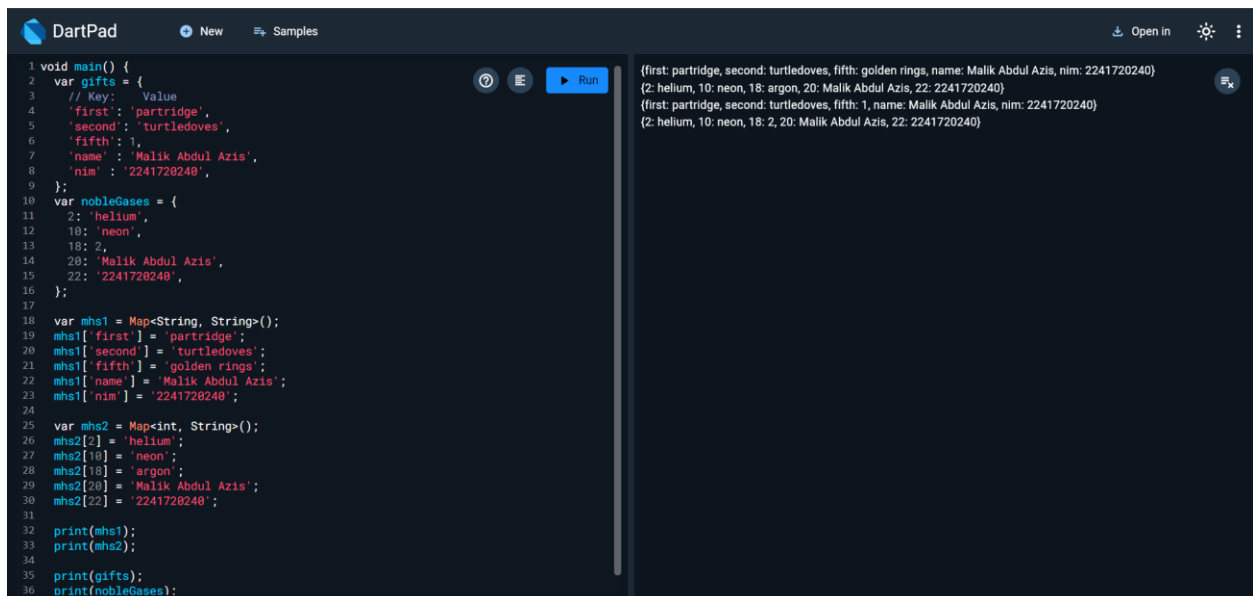
- Langkah 2



```
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
```

(first: partridge, second: turtledoves, fifth: 1)
(2: helium, 10: neon, 18: 2)

- Langkah 3

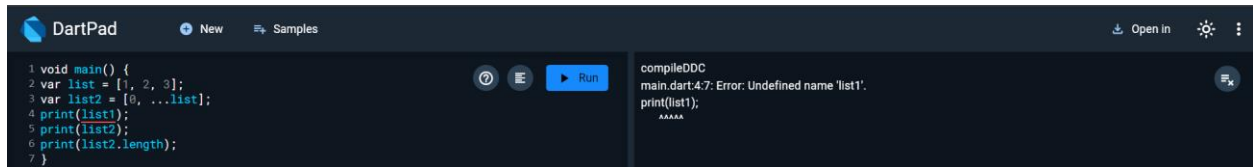


```
1 void main() {  
2   var gifts = {  
3     // Key:   Value  
4     'first': 'partridge',  
5     'second': 'turtledoves',  
6     'fifth': 1,  
7     'name': 'Malik Abdul Azis',  
8     'nim': '2241720240',  
9   };  
10  var nobleGases = {  
11    2: 'helium',  
12    10: 'neon',  
13    18: 2,  
14    20: 'Malik Abdul Azis',  
15    22: '2241720240',  
16  };  
17  
18  var mhs1 = Map<String, String>();  
19  mhs1['first'] = 'partridge';  
20  mhs1['second'] = 'turtledoves';  
21  mhs1['fifth'] = 'golden rings';  
22  mhs1['name'] = 'Malik Abdul Azis';  
23  mhs1['nim'] = '2241720240';  
24  
25  var mhs2 = Map<int, String>();  
26  mhs2[2] = 'helium';  
27  mhs2[10] = 'neon';  
28  mhs2[18] = 'argon';  
29  mhs2[20] = 'Malik Abdul Azis';  
30  mhs2[22] = '2241720240';  
31  
32  print(mhs1);  
33  print(mhs2);  
34  
35  print(gifts);  
36  print(nobleGases);  
37 }
```

(first: partridge, second: turtledoves, fifth: golden rings, name: Malik Abdul Azis, nim: 2241720240)
(2: helium, 10: neon, 18: argon, 20: Malik Abdul Azis, 22: 2241720240)
(first: partridge, second: turtledoves, fifth: 1, name: Malik Abdul Azis, nim: 2241720240)
(2: helium, 10: neon, 18: 2, 20: Malik Abdul Azis, 22: 2241720240)

Practicum 4

- Langkah 1
Terjadi error “Undefined name”

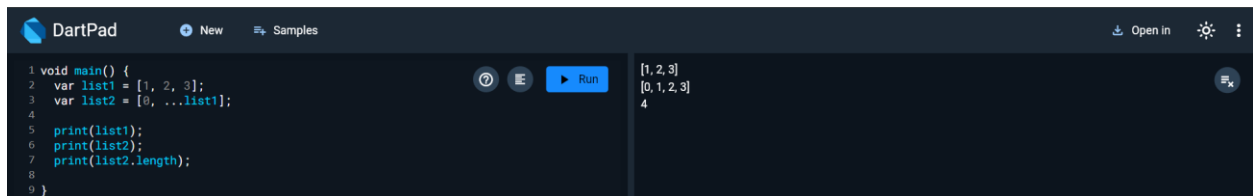


The screenshot shows the DartPad interface. On the left, the code is as follows:

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

On the right, the error message reads: "compileDDC main.dart:4:7: Error: Undefined name 'list1'. print(list1);".

- Langkah 2
Jika diperbaiki seperti berikut, maka output akan keluar



The screenshot shows the DartPad interface with the same code as in Langkah 1. The output on the right is:

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

- Langkah 3



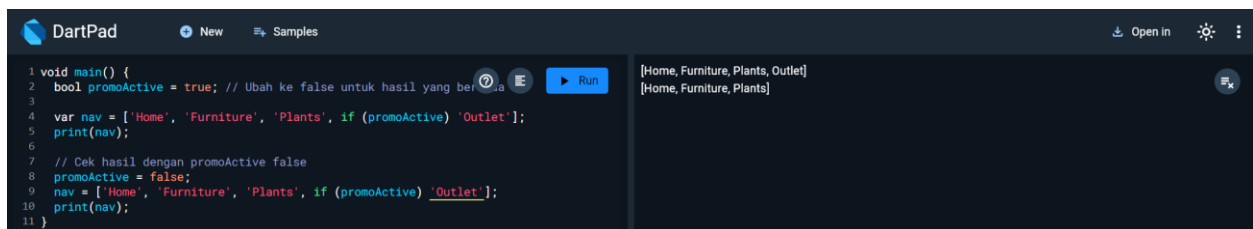
The screenshot shows the DartPad interface with the following code:

```
1 void main() {  
2   var list1 = [1, 2, null];  
3   print(list1);  
4  
5   // Menggunakan spread operator dengan null check  
6   var list3 = [0, ...?list1]; // Menggunakan ? untuk menghindari error jika list1  
   null  
7   print(list3);  
8   print(list3.length);  
9  
10  // Menambahkan variabel list berisi  
11  var nim = [2241720240];  
12  var listNIM = [...nim]; // Menggunakan spread operator untuk menambahkan NIM ke  
   list  
13  print(listNIM);  
14 }
```

The output on the right is:

```
[1, 2, null]  
[0, 1, 2, null]  
4  
[2241720240]
```

- Langkah 4



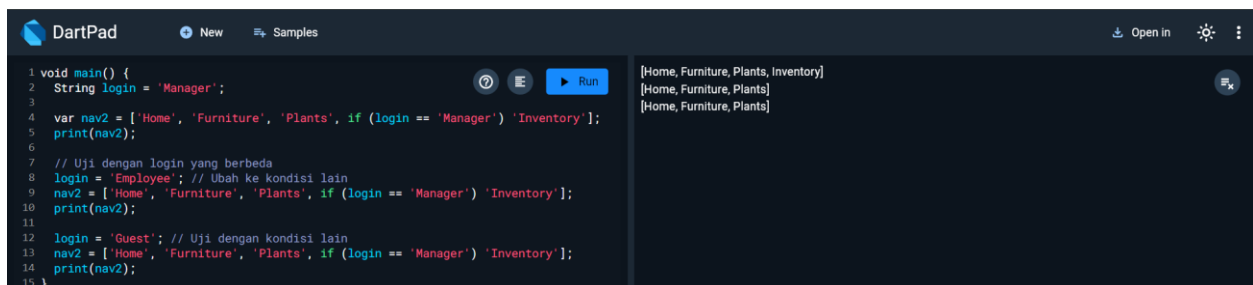
The screenshot shows the DartPad interface with the following code:

```
1 void main() {  
2   bool promoActive = true; // Ubah ke false untuk hasil yang ber...  
3  
4   var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];  
5   print(nav);  
6  
7   // Cek hasil dengan promoActive false  
8   promoActive = false;  
9   nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];  
10  print(nav);  
11 }
```

The output on the right is:

```
[Home, Furniture, Plants, Outlet]  
[Home, Furniture, Plants]
```

- Langkah 5



The screenshot shows the DartPad interface with the following code:

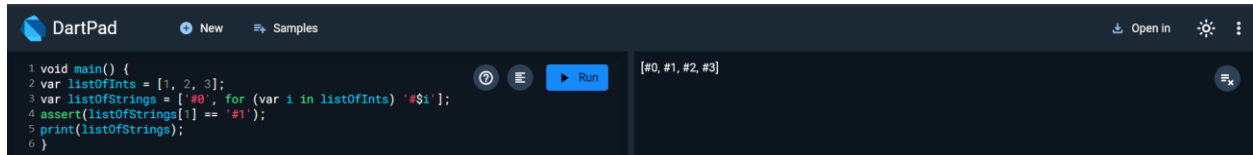
```
1 void main() {  
2   String login = 'Manager';  
3  
4   var nav2 = ['Home', 'Furniture', 'Plants', if (login == 'Manager') 'Inventory'];  
5   print(nav2);  
6  
7   // Uji dengan login yang berbeda  
8   login = 'Employee'; // Ubah ke kondisi lain  
9   nav2 = ['Home', 'Furniture', 'Plants', if (login == 'Manager') 'Inventory'];  
10  print(nav2);  
11  
12  login = 'Guest'; // Uji dengan kondisi lain  
13  nav2 = ['Home', 'Furniture', 'Plants', if (login == 'Manager') 'Inventory'];  
14  print(nav2);  
15 }
```

The output on the right is:

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

- Langkah 6

Dalam kode ini, untuk setiap elemen *i* dalam `listOfInts`, akan dibuat string baru yang berbentuk '#' diikuti oleh nilai *i*.



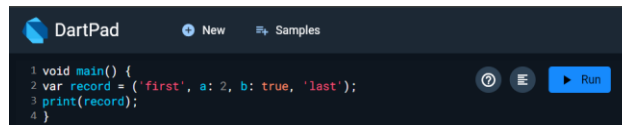
The screenshot shows the DartPad web interface. The top bar includes the DartPad logo, a 'New' button, a 'Samples' dropdown, and links for 'Open in' and settings. The main editor area contains the following Dart code:

```
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 }
```

Below the code is a 'Run' button. To the right of the code editor, the output of the program is displayed: `['#0', '#1', '#2', '#3']`.

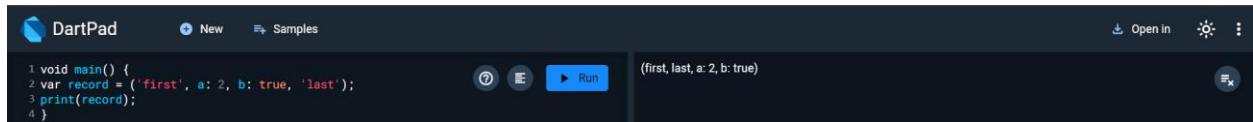
Practicum 5

- Langkah 1



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

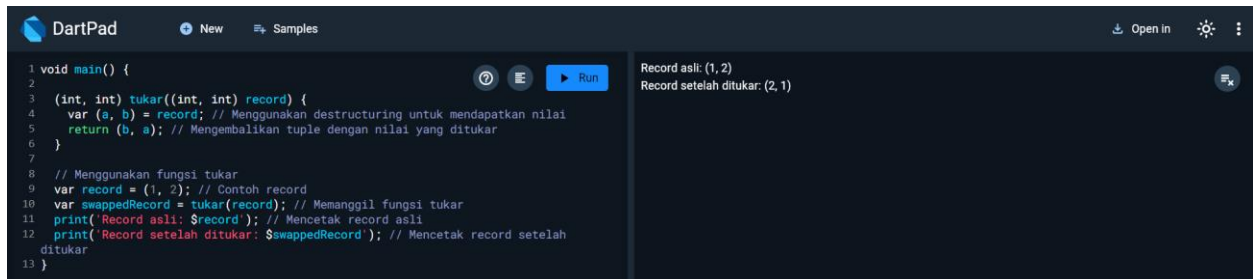
- Langkah 2



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

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

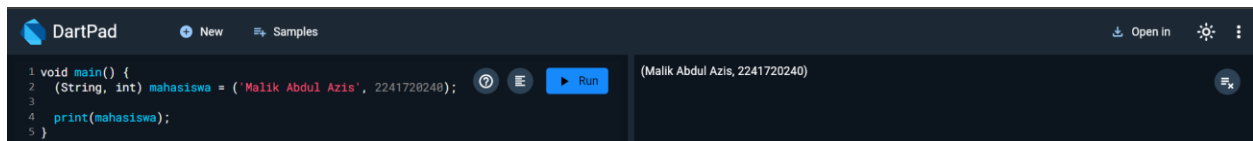
- Langkah 3



```
1 void main() {  
2  
3   (int, int) tukar((int, int) record) {  
4     var (a, b) = record; // Menggunakan destructuring untuk mendapatkan nilai  
5     return (b, a); // Mengembalikan tuple dengan nilai yang ditukar  
6   }  
7  
8   // Menggunakan fungsi tukar  
9   var record = (1, 2); // Contoh record  
10  var swappedRecord = tukar(record); // Memanggil fungsi tukar  
11  print('Record asli: $record'); // Mencetak record asli  
12  print('Record setelah ditukar: $swappedRecord'); // Mencetak record setelah  
   ditukar  
13 }
```

Record asli: (1, 2)
Record setelah ditukar: (2, 1)

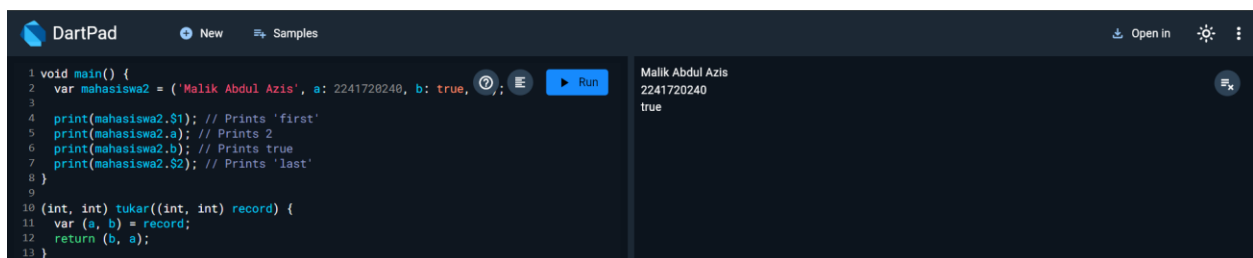
- Langkah 4



```
1 void main() {  
2   (String, int) mahasiswa = ('Malik Abdul Azis', 2241720240);  
3  
4   print(mahasiswa);  
5 }
```

(Malik Abdul Azis, 2241720240)

- Langkah 5



```
1 void main() {  
2   var mahasiswa2 = ('Malik Abdul Azis', a: 2241720240, b: true, 'last');  
3  
4   print(mahasiswa2.$1); // Prints 'first'  
5   print(mahasiswa2.a); // Prints 2  
6   print(mahasiswa2.b); // Prints true  
7   print(mahasiswa2.$2); // Prints 'last'  
8 }  
9  
10 (int, int) tukar((int, int) record) {  
11   var (a, b) = record;  
12   return (b, a);  
13 }
```

Malik Abdul Azis
2241720240
true

Practicum Task

1. Silakan selesaikan Praktikum 1 sampai 5, lalu dokumentasikan berupa screenshot hasil pekerjaan Anda beserta penjelasannya!
Mulai Halaman 2
2. Jelaskan yang dimaksud Functions dalam bahasa Dart!
Fungsi (function) dalam Dart adalah blok kode yang dapat dipanggil berkali-kali untuk melakukan tugas tertentu. Fungsi membuat kode lebih terorganisir, mudah dibaca, dan dapat digunakan kembali.
3. Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!
 - Parameter Wajib (Required Parameter): Parameter yang harus diisi nilainya saat memanggil fungsi.
 - Parameter Opsional (Optional Parameter): Parameter yang boleh diisi atau tidak.
 - Parameter Opsional Posisional: Posisi parameter tidak penting, bisa diisi atau tidak. Ditandai dengan tanda [].
 - Parameter Opsional Bernama: Harus diberi nama saat memanggil fungsi. Ditandai dengan tanda {}.
 - Parameter Default: Parameter yang memiliki nilai default jika tidak diisi saat pemanggilan.
4. Jelaskan maksud Functions sebagai first-class objects beserta contoh sintaknya!
Artinya, fungsi dapat:
 - Ditugaskan ke variabel
 - Dikirim sebagai argumen ke fungsi lain
 - Dikembalikan sebagai nilai dari fungsi

```
void sayHello() {  
  print('Hello!');  
}  
  
void greetFunction(Function greetFunction) {  
  greetFunction();  
}  
  
greetFunction(sayHello);
```

5. Apa itu Anonymous Functions? Jelaskan dan berikan contohnya!
Anonymous Functions adalah fungsi yang tidak memiliki nama. Biasanya digunakan secara singkat, misalnya sebagai callback atau dalam lambda expression.

```
var list = [1, 2, 3];  
list.forEach((element) {  
  print(element);  
});
```

6. Jelaskan perbedaan Lexical scope dan Lexical closures! Berikan contohnya!
 - Lexical Scope: Variabel yang dapat diakses dalam suatu fungsi ditentukan oleh letak fungsi tersebut dalam kode. Variabel yang dideklarasikan di luar fungsi dapat diakses di dalam fungsi jika fungsi tersebut berada dalam scope variabel tersebut.

- Lexical Closures: Sebuah closure adalah fungsi yang memiliki akses ke variabel yang ada di luar scope-nya, bahkan setelah fungsi luarnya selesai dieksekusi.

```
Function makeCounter() {  
  var count = 0;  
  return () => count++;  
}  
  
var counter = makeCounter();  
print(counter()); // Output: 1  
print(counter()); // Output: 2
```

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

Dalam Dart, kita tidak bisa langsung mengembalikan beberapa nilai dari sebuah fungsi. Namun, kita bisa menggunakan:

- List: Mengembalikan sebuah list yang berisi nilai-nilai yang ingin dikembalikan.
- Class: Membuat sebuah kelas yang merepresentasikan nilai-nilai yang ingin dikembalikan, kemudian mengembalikan objek dari kelas tersebut.
- Tuple: (Mulai dari Dart 2.12) Memungkinkan kita untuk mengembalikan beberapa nilai dalam satu ekspresi.

Contoh List

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

Contoh Class

```
class Person {  
  String name;  
  int age;  
  
  Person(this.name, this.age);  
}  
  
Person createPerson() {  
  return Person('John Doe', 30);  
}
```

Contoh Tuple

```
(int, String) getPair() {  
  return (42, 'answer');  
}
```

Github Link

<https://github.com/mhankazis/PemrogramanMobile/tree/main/Week%204>