

Pengantar Bahasa Pemrograman Dart

BAGIAN 3

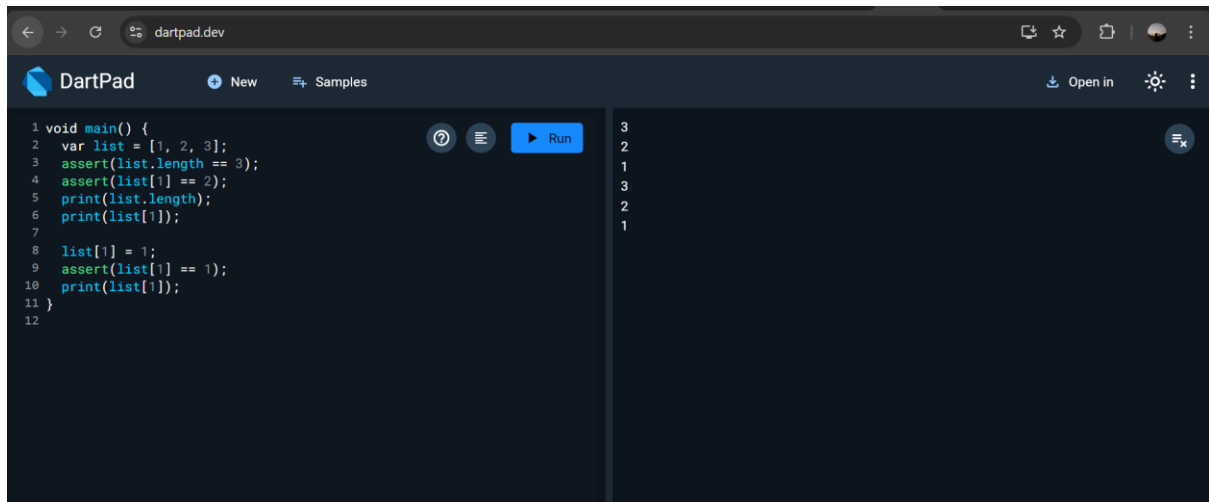


ANANDA AZ HARUDDIN SALIMA

2241720071

PRAKTIKUM 1: EKSPERIMEN TIPE DATA LIST

Langkah 1:



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

```
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 }  
12
```

The right pane shows the output of the code, which is:

```
3  
2  
1  
3  
2  
1
```

Langkah 2

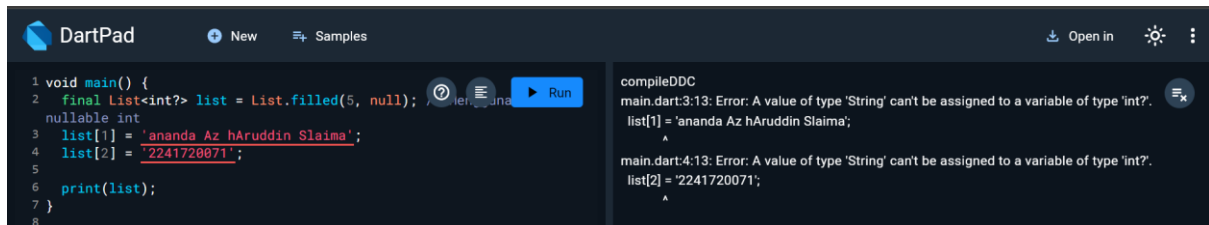
Penjelasan Langkah 1:

1. **Deklarasi List:** List berisi tiga elemen: 1, 2, dan 3.
2. **Assert Statements:**
 - `assert(list.length == 3)` memeriksa apakah panjang list adalah 3. Jika tidak, akan menimbulkan kesalahan.
 - `assert(list[1] == 2)` memeriksa apakah elemen indeks ke-1 (yang kedua) adalah 2.
3. **Print Statements:**
 - `print(list.length)` mencetak panjang list, yang seharusnya 3.
 - `print(list[1])` mencetak elemen di indeks ke-1, yang seharusnya 2.
4. **Mengubah Elemen:** `list[1] = 1;` mengganti nilai elemen di indeks ke-1 dengan 1.
5. **Assert Statement:** `assert(list[1] == 1)` memeriksa apakah elemen di indeks ke-1 sekarang adalah 1.
6. **Print Statement:** `print(list[1])` mencetak nilai elemen ke-1 yang baru, yang seharusnya 1.

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.

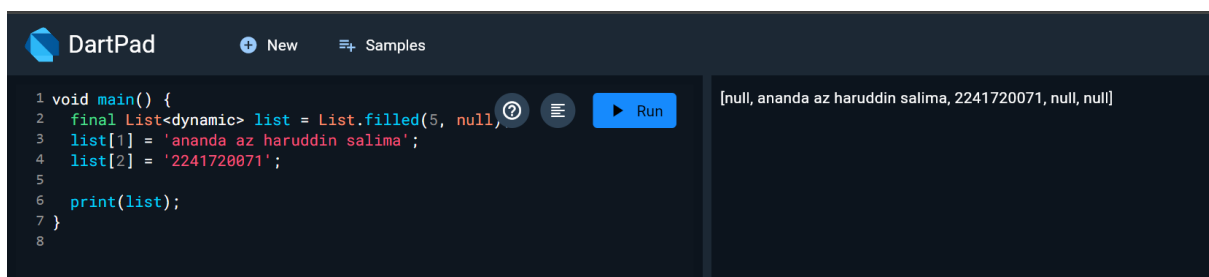


```
1 void main() {
2   final List<int?> list = List.filled(5, null);
3   list[1] = 'ananda Az hAruddin Slaima';
4   list[2] = '2241720071';
5
6   print(list);
7 }
8
```

compileDDC
main.dart:3:13: Error: A value of type 'String' can't be assigned to a variable of type 'int?'.
list[1] = 'ananda Az hAruddin Slaima';
^
main.dart:4:13: Error: A value of type 'String' can't be assigned to a variable of type 'int?'.
list[2] = '2241720071';
^

string (nama dan NIM) di dalam list yang dideklarasikan sebagai List<int?>, maka akan terjadi kesalahan tipe data (type error).

Perbaiki



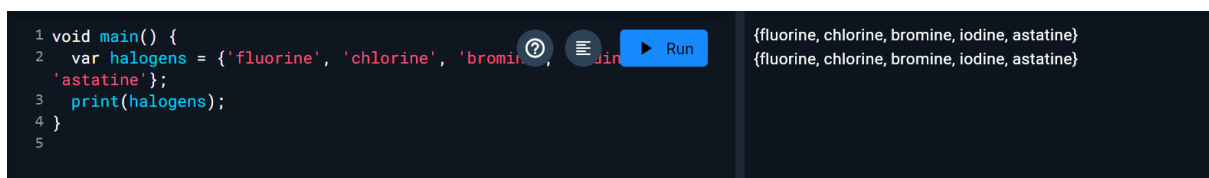
```
1 void main() {
2   final List<dynamic> list = List.filled(5, null);
3   list[1] = 'ananda az haruddin salima';
4   list[2] = '2241720071';
5
6   print(list);
7 }
8
```

[null, ananda az haruddin salima, 2241720071, null, null]

program akan berjalan tanpa kesalahan, dan outputnya akan menunjukkan isi list yang mencakup null di indeks lainnya dan nama serta NIM Anda di indeks ke-1 dan ke-2.

PRAKTIKUM 2: EKSPERIMEN TIPE DATA SET

Langkah 1



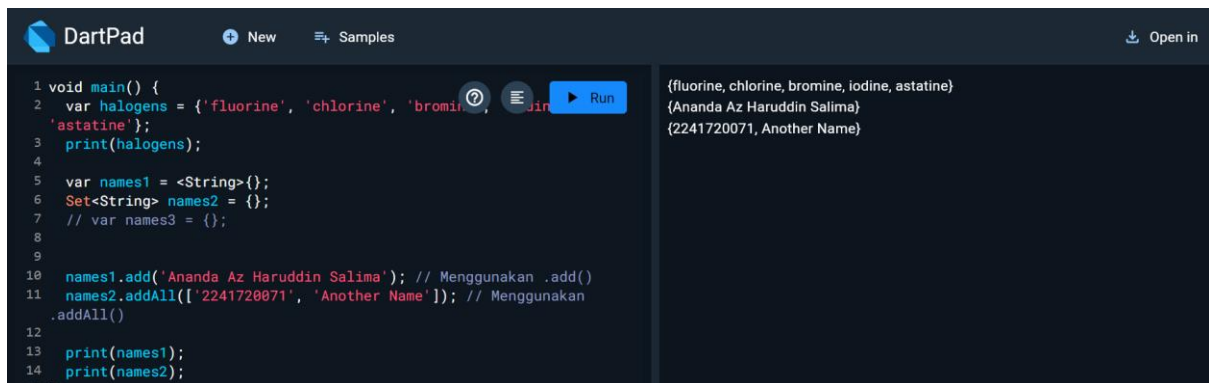
```
1 void main() {
2   var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};
3   print(halogens);
4 }
5
```

{fluorine, chlorine, bromine, iodine, astatine}
{fluorine, chlorine, bromine, iodine, astatine}

Langkah 2

- ☐ Deklarasi Set: Kode di atas mendeklarasikan sebuah set bernama halogens yang berisi beberapa elemen.
- ☐ Print Statement: print(halogens); akan mencetak isi dari set tersebut.

Langkah 3



```
1 void main() {
2   var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};
3   print(halogens);
4
5   var names1 = <String>{};
6   Set<String> names2 = {};
7   // var names3 = {};
8
9
10  names1.add('Ananda Az Haruddin Salima'); // Menggunakan .add()
11  names2.addAll(['2241720071', 'Another Name']); // Menggunakan
    .addAll()
12
13  print(names1);
14  print(names2);
15 }
```

{fluorine, chlorine, bromine, iodine, astatine}
{Ananda Az Haruddin Salima}
{2241720071, Another Name}

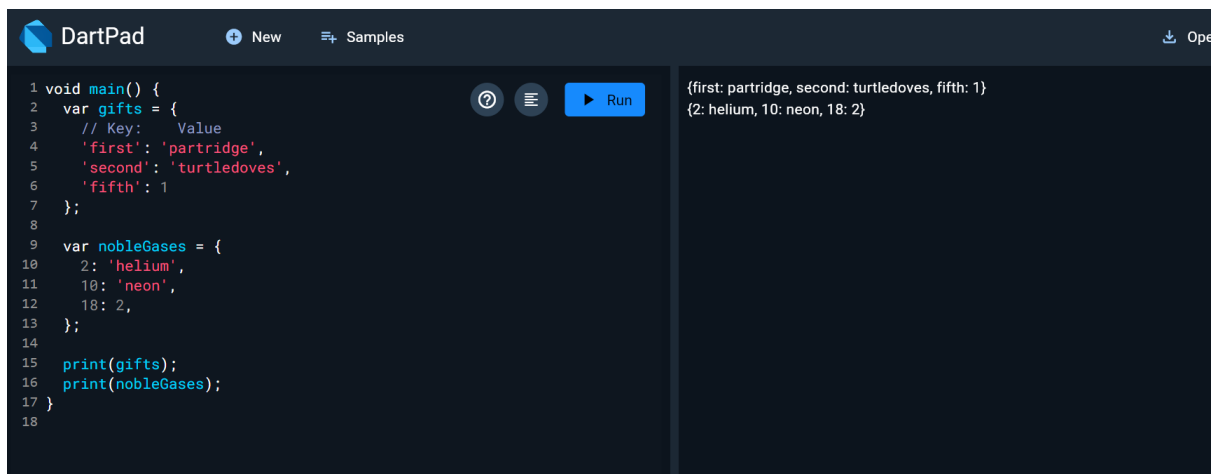
- **Menambahkan Elemen:**

- `names1.add('David M. H.');` menambahkan nama ke `names1` menggunakan metode `.add()`.
- `names2.addAll(['123456789', 'Another Name']);` menambahkan beberapa elemen ke `names2` menggunakan metode `.addAll()`.

- **Output Akhir:** Setelah menjalankan kode ini, kita akan melihat output yang mencakup isi dari `halogens`, `names1`, dan `names2`.

PRAKTIKUM 3: EKSPERIMEN TIPE DATA MAPS

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
```

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

Langkah 2

Penjelasan Langkah 1:

1. **Deklarasi Map:**

- `gifts` adalah map yang menggunakan string sebagai key dan bisa memiliki nilai yang berbeda tipe (string dan integer).
- `nobleGases` adalah map yang menggunakan integer sebagai key dan bisa memiliki nilai yang berbeda (string dan integer).

2. Print Statement:

- o `print(gifts);` akan mencetak isi map `gifts`.
- o `print(nobleGases);` akan mencetak isi map `nobleGases`.

Langkah 3



The screenshot shows the DartPad interface. On the left, the code editor contains the following Dart code:

```
3 // Key: Value
4 'first': 'partridge',
5 'second': 'turtledoves',
6 'fifth': 'golden rings', // Memperbarui nilai
7 };
8
9 var nobleGases = {
10 2: 'helium',
11 10: 'neon',
12 18: 'argon', // Memperbarui nilai
13 };
14
15 print(gifts);
16 print(nobleGases);
17
18 var mhs1 = Map<String, String>();
19 mhs1['name'] = 'David M. H.'; // Ganti dengan nama Anda
20 mhs1['NIM'] = '123456789'; // Ganti dengan NIM Anda
21
22 var mhs2 = Map<int, String>();
23 mhs2[1] = 'David M. H.'; // Ganti dengan nama Anda
24 mhs2[2] = '123456789'; // Ganti dengan NIM Anda
25
```

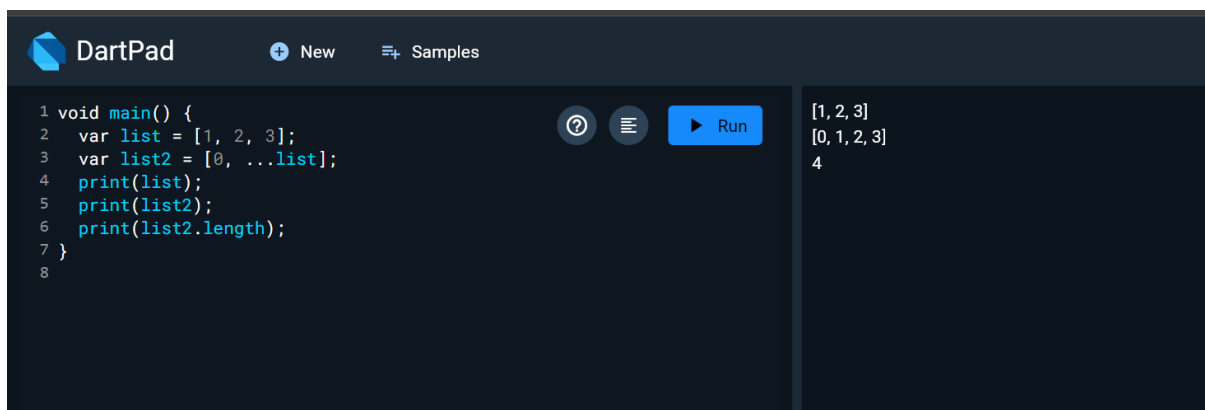
On the right, the output console shows the following results:

```
{first: partridge, second: turtledoves, fifth: golden rings}
{2: helium, 10: neon, 18: argon}
{name: David M. H., NIM: 123456789}
{1: David M. H., 2: 123456789}
```

- **Tujuan:** Mempelajari penggunaan struktur data Map di Dart.
- **Hasil:**
 - Map `gifts` dan `nobleGases` berhasil dibuat dan dimodifikasi.
 - Elemen nama dan NIM berhasil ditambahkan ke dalam map `mhs1` dan `mhs2`.
- **Kesimpulan:** Praktikum ini memberikan pemahaman yang lebih baik tentang penggunaan Map dalam Dart, serta cara menambahkan dan memodifikasi elemen dalam struktur data tersebut.

PRAKTIKUM 4: EKSPERIMEN TIPE DATA LIST: SPREAD DAN CONTROL-FLOW OPERATORS

Langkah 1



The screenshot shows the DartPad interface. On the left, the code editor contains the following Dart code:

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

On the right, the output console shows the following results:

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

Langkah 2

- **Spread Operator** (...) digunakan untuk memasukkan semua elemen dari `list` ke dalam `list2`.
- `print(list)` akan mencetak isi dari `list`, dan `print(list2)` akan mencetak `list2` yang berisi elemen dari `list` serta elemen awal 0.
- `print(list2.length)` akan mencetak panjang dari `list2`.

Langkah 3

```
1 void main() {
2   var list1 = [1, 2, null];
3   print(list1);
4   var list3 = [0, ...?list1];
5   print(list3.length);
6 }
7
```

[1, 2, null]
4

- **Spread Operator dengan Null Safety** (...?) digunakan untuk menggabungkan elemen dari `list1` ke dalam `list3`. Jika `list1` adalah `null`, maka tidak akan ada elemen yang ditambahkan, sehingga ini lebih aman.
- `print(list1)` akan mencetak `list1`, dan `print(list3.length)` akan mencetak panjang dari `list3`.

Langkah 4

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

[Home, Furniture, Plants, Outlet]

Penjelasan Langkah 4:

- Jika `promoActive` adalah `true`, elemen `Outlet` akan ditambahkan ke dalam list `nav`.
- Jika `promoActive` adalah `false`, elemen `Outlet` tidak akan ditambahkan.

Langkah 5

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

[Home, Furniture, Plants, Inventory]

Penjelasan Langkah 5:

- Jika login adalah 'Manager', maka elemen 'Inventory' akan ditambahkan ke list nav2.
- Jika login bukan 'Manager', elemen 'Inventory' tidak akan ditambahkan.

Langkah 6

```
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
```

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

Penjelasan Langkah 6:

- for digunakan untuk membuat koleksi dinamis. Setiap elemen di listOfInts akan di-loop dan dikonversi ke string dengan awalan #.
- Hasilnya adalah list baru listOfStrings yang berisi string

Manfaat Collection For:

- Mempermudah pembuatan koleksi baru berdasarkan iterasi dari koleksi lain.
- Meminimalisasi kode yang berulang dan meningkatkan efisiensi.

PRAKTIKUM 5: EKSPERIMEN TIPE DATA RECORDS

Langkah 1

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

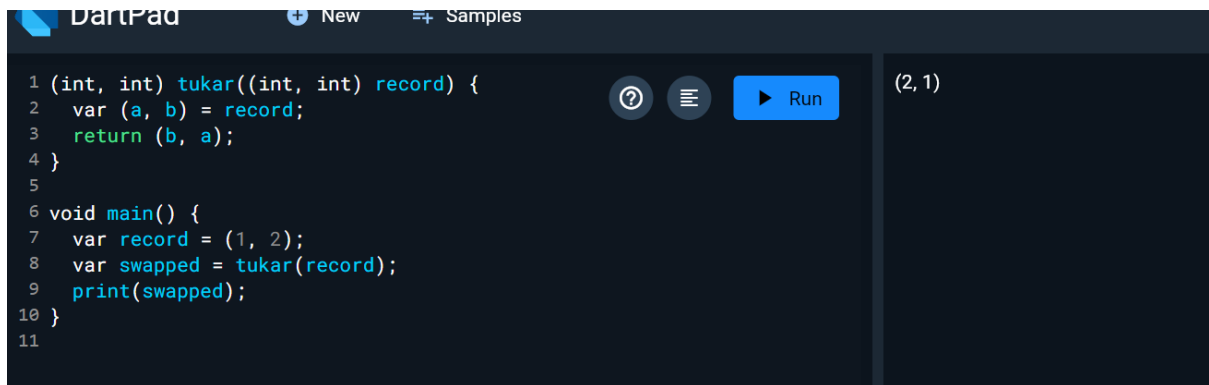
(first, last, a: 2, b: true)

Langkah 2

Penjelasan Langkah 1:

- Kode ini menggunakan tipe data **Record**. record berisi elemen bernama (a, b) dan elemen tanpa nama ('first' dan 'last').
- Saat menjalankan kode ini, Dart akan mencetak seluruh isi dari record.

Langkah 3



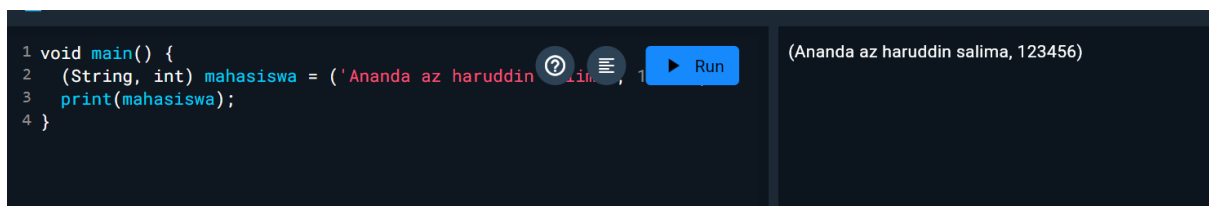
```
1 (int, int) tukar((int, int) record) {
2   var (a, b) = record;
3   return (b, a);
4 }
5
6 void main() {
7   var record = (1, 2);
8   var swapped = tukar(record);
9   print(swapped);
10 }
11
```

(2, 1)

Penjelasan Langkah 2:

- Fungsi `tukar()` menerima `record` dengan tipe `(int, int)` dan mengembalikan `record` dengan urutan nilai yang ditukar.
- `var (a, b) = record` memecah `record` menjadi dua variabel `a` dan `b`, kemudian mengembalikannya dalam urutan terbalik

Langkah 4



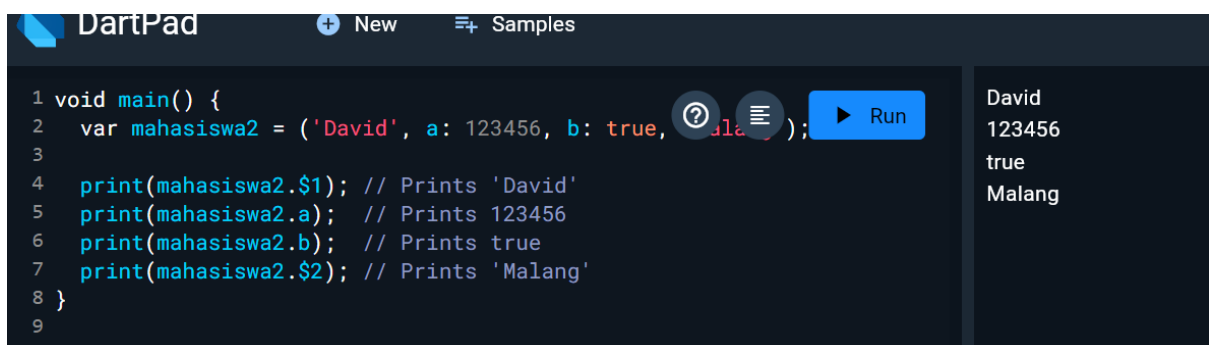
```
1 void main() {
2   (String, int) mahasiswa = ('Ananda az haruddin salima', 123456);
3   print(mahasiswa);
4 }
```

(Ananda az haruddin salima, 123456)

Penjelasan Langkah 3:

- Deklarasi `(String, int) mahasiswa` berarti variabel ini merupakan `record` dengan tipe data `String` dan `int`.
- Inisialisasi dengan nama dan NIM `('ananda a zharuddin salima', 123456)` kemudian mencetaknya.

Langkah 5



```
1 void main() {
2   var mahasiswa2 = ('David', a: 123456, b: true, c: 'Malang');
3
4   print(mahasiswa2.$1); // Prints 'David'
5   print(mahasiswa2.a); // Prints 123456
6   print(mahasiswa2.b); // Prints true
7   print(mahasiswa2.$2); // Prints 'Malang'
8 }
9
```

David
123456
true
Malang

Penjelasan Langkah 4:

- Elemen bernama (a dan b) dapat diakses dengan namanya, sedangkan elemen tidak bernama dapat diakses dengan penomoran (\$1, \$2).
- Pada contoh ini, elemen pertama (\$1) berisi 'David', elemen kedua (a) berisi 123456, dan seterusnya.

TUGAS PRACTICUM

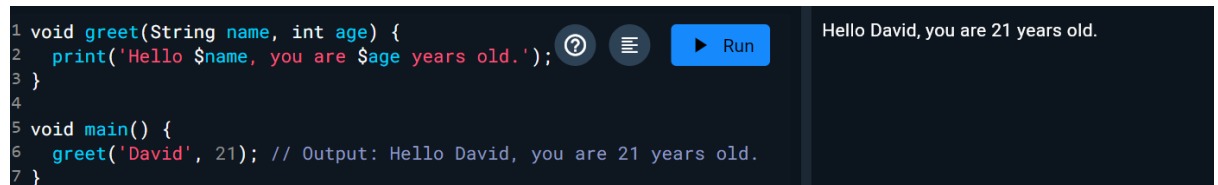
1. Jelaskan yang dimaksud Functions dalam bahasa Dart!

Functions adalah blok kode yang dapat digunakan kembali dan dieksekusi saat dipanggil.

2. Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!

Positional Parameters: Parameter yang harus diberikan dalam urutan tertentu.

```
1 void greet(String name, int age) {  
2   print('Hello $name, you are $age years old.');
```



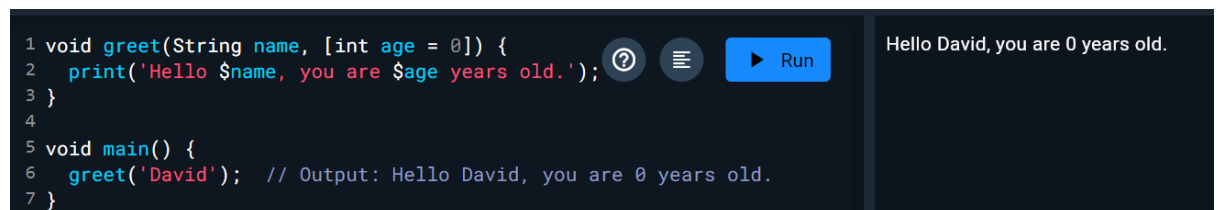
The image shows a code editor with a function `greet` that takes two positional parameters: `String name` and `int age`. In the `main` function, `greet` is called with the arguments `'David'` and `21`. The output on the right shows the result of the function call.

```
3 }  
4  
5 void main() {  
6   greet('David', 21); // Output: Hello David, you are 21 years old.  
7 }
```

Hello David, you are 21 years old.

Optional Positional Parameters: Parameter yang opsional, ditandai dengan []. Jika tidak diisi, mereka akan bernilai null secara default atau bisa diberi nilai default.

```
1 void greet(String name, [int age = 0]) {  
2   print('Hello $name, you are $age years old.');
```



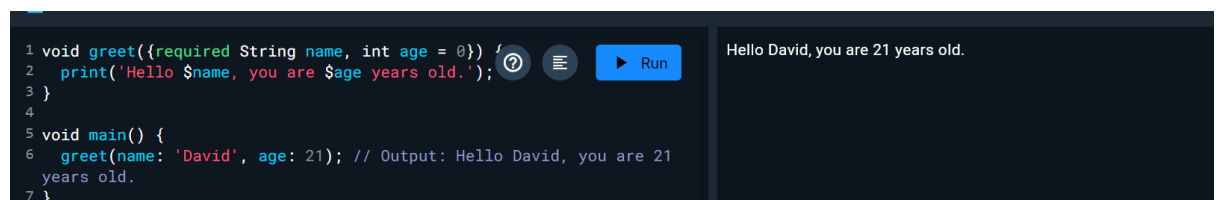
The image shows a code editor with a function `greet` that has an optional positional parameter `age` with a default value of `0`. In the `main` function, `greet` is called with only the `name` argument `'David'`. The output on the right shows the result of the function call.

```
3 }  
4  
5 void main() {  
6   greet('David'); // Output: Hello David, you are 0 years old.  
7 }
```

Hello David, you are 0 years old.

Named Parameters: Parameter yang diberi nama dan bisa dipanggil dalam urutan bebas. Biasanya digunakan untuk meningkatkan keterbacaan kode.

```
1 void greet({required String name, int age = 0}) {  
2   print('Hello $name, you are $age years old.');
```



The image shows a code editor with a function `greet` that has named parameters: `String name` and `int age`. In the `main` function, `greet` is called with named arguments `name: 'David'` and `age: 21`. The output on the right shows the result of the function call.

```
3 }  
4  
5 void main() {  
6   greet(name: 'David', age: 21); // Output: Hello David, you are 21  
   years old.  
7 }
```

Hello David, you are 21 years old.

Optional Named Parameters: Sama seperti named parameters, tetapi bersifat opsional.

```
1 void greet({String name = 'Guest', int age = 0}) {  
2   print('Hello $name, you are $age years old.');
```



The image shows a code editor with a function `greet` that has optional named parameters: `String name` with a default value of `'Guest'` and `int age` with a default value of `0`. In the `main` function, `greet` is called without any arguments. The output on the right shows the result of the function call.

```
3 }  
4  
5 void main() {  
6   greet(); // Output: Hello Guest, you are 0 years old.  
7 }  
8
```

Hello Guest, you are 0 years old.

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

Dart memperlakukan functions sebagai first-class objects, yang berarti functions bisa disimpan dalam variabel, dilewatkan sebagai argumen ke fungsi lain, atau dikembalikan oleh fungsi lain.

```
void sayHello(String name) {  
  print('Hello $name');  
}  
  
void main() {  
  // Menyimpan function dalam variabel  
  var greet = sayHello;  
  
  // Menggunakan variabel untuk memanggil function  
  greet('David'); // Output: Hello David  
}
```

Run

Hello David

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

Anonymous functions adalah functions yang tidak memiliki nama. Mereka sering digunakan sebagai argumen untuk fungsi lain, misalnya dalam fungsi `forEach` atau sebagai callback.

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

Run

Number: 1
Number: 2
Number: 3
Number: 4
Number: 5

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

Lexical Scope: Merujuk pada cara Dart menentukan di mana variabel dapat diakses berdasarkan di mana mereka dideklarasikan. Variabel hanya dapat diakses dalam lingkup tempat mereka dideklarasikan.

```
1 void main() {  
2   int outerVariable = 100;  
3  
4   void innerFunction() {  
5     print(outerVariable); // outerVariable bisa diakses  
6   }  
7  
8   innerFunction(); // Output: 100  
9 }
```

Run

100

Lexical Closures: Closure adalah fungsi yang "menangkap" variabel dari lingkup di mana fungsi tersebut didefinisikan. Closure dapat tetap mengakses variabel dari luar lingkup setelah fungsi tersebut selesai dieksekusi.

```
1 Function createCounter() {  
2   int count = 0;  
3  
4   return () {  
5     count += 1;  
6     return count;  
7   };  
8 }  
9  
10 void main() {  
11   var counter = createCounter();  
12   print(counter()); // Output: 1  
13   print(counter()); // Output: 2  
14 }  
15
```

Run

1
2

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

Dart tidak secara langsung mendukung pengembalian beberapa nilai, tetapi dapat dicapai dengan menggunakan **tuple** (via Record), **List**, atau **Map**.

- Dengan list

```
1 List<int> getCoordinates() {
2   return [10, 20];
3 }
4
5 void main() {
6   var coords = getCoordinates();
7   print('x: ${coords[0]}, y: ${coords[1]}'); // Output: x: 10, y: 20
8 }
9
```

x: 10, y: 20

- Dengan tuple

```
1 (int, int) getCoordinates() {
2   return (10, 20);
3 }
4
5 void main() {
6   var (x, y) = getCoordinates();
7   print('x: $x, y: $y'); // Output: x: 10, y: 20
8 }
9
```

x: 10, y: 20

- Dengan map

```
1 Map<String, int> getCoordinates() {
2   return {'x': 10, 'y': 20};
3 }
4
5 void main() {
6   var coords = getCoordinates();
7   print('x: ${coords['x']}, y: ${coords['y']}'); // Output: x: 10,
8   y: 20
9 }
```

x: 10, y: 20