

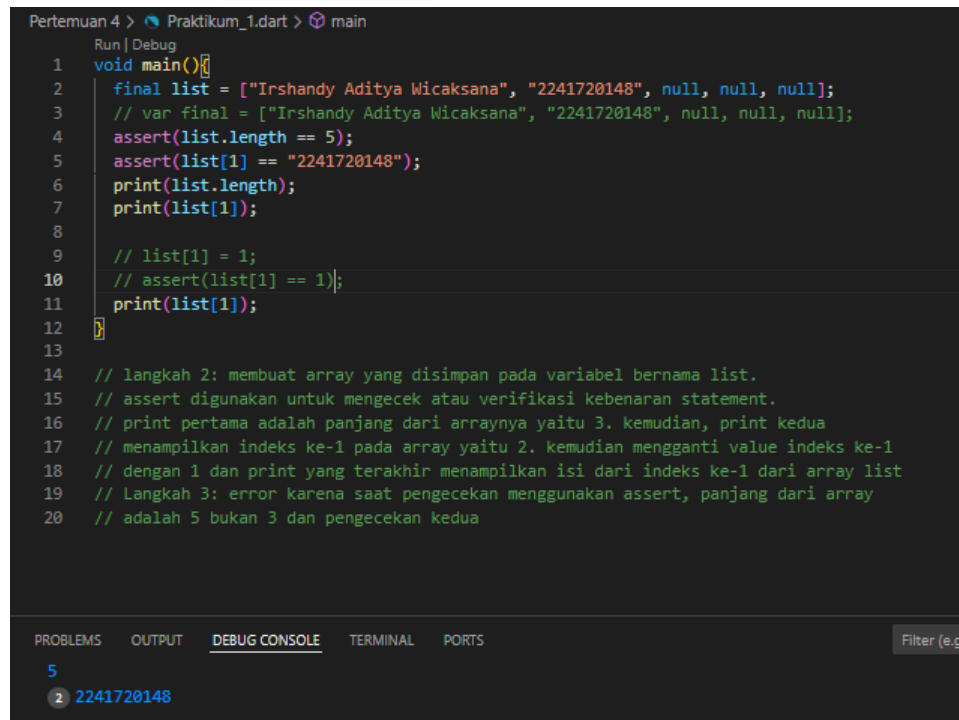
Irshandy Aditya Wicaksana

TI-3A / 16

2241720148

1. Laporan Praktikum:

Praktikum 1:



The screenshot shows an IDE window titled 'Pertemuan 4 > Praktikum_1.dart > main'. The code is in Dart and defines a list with five elements: a name, an ID, and three null values. It includes assertions for the list's length and the value at index 1, followed by print statements. Comments explain the steps: creating the array, using assert for verification, and noting an error in the second assertion because the array length is 5, not 3. The bottom panel shows the 'DEBUG CONSOLE' with the output '5' and '2241720148'.

```
Pertemuan 4 > Praktikum_1.dart > main
Run | Debug
1 void main() {
2   final list = ["Irshandy Aditya Wicaksana", "2241720148", null, null, null];
3   // var final = ["Irshandy Aditya Wicaksana", "2241720148", null, null, null];
4   assert(list.length == 5);
5   assert(list[1] == "2241720148");
6   print(list.length);
7   print(list[1]);
8
9   // list[1] = 1;
10  // assert(list[1] == 1);
11  print(list[1]);
12 }
13
14 // langkah 2: membuat array yang disimpan pada variabel bernama list.
15 // assert digunakan untuk mengecek atau verifikasi kebenaran statement.
16 // print pertama adalah panjang dari arraynya yaitu 3. kemudian, print kedua
17 // menampilkan indeks ke-1 pada array yaitu 2. kemudian mengganti value indeks ke-1
18 // dengan 1 dan print yang terakhir menampilkan isi dari indeks ke-1 dari array list
19 // Langkah 3: error karena saat pengecekan menggunakan assert, panjang dari array
20 // adalah 5 bukan 3 dan pengecekan kedua
```

PROBLEMS OUTPUT **DEBUG CONSOLE** TERMINAL PORTS Filter (e.g.)

5
2 2241720148

// langkah 2: membuat array yang disimpan pada variabel bernama list. assert digunakan untuk mengecek atau verifikasi kebenaran statement. print pertama adalah panjang dari arraynya yaitu 3. kemudian, print kedua menampilkan indeks ke-1 pada array yaitu 2. kemudian mengganti value indeks ke-1 dengan 1 dan print yang terakhir menampilkan isi dari indeks ke-1 dari array list

Langkah 3: error karena saat pengecekan menggunakan assert, panjang dari array adalah 5 bukan 3 dan pengecekan kedua

Praktikum 2:

```
Run | Debug
1 void main(){
2     var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};
3     print(halogens);
4
5     var names1 = <String>{};
6     Set<String> names2 = {}; // This works, too.
7     var names3 = {}; // Creates a map, not a set.
8
9     names1.add("Irshandy Aditya Wicaksana");
10    names1.add("2241720148");
11
12    names2.addAll({"Irshandy Aditya", "2241720148"});
13
14    print(names1);
15    print(names2);
16    print(names3);
17 }
18
19 // langkah 2: menampilkan data set pada variabel halogens
20 // langkah 3: tidak error
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text)

```
{fluorine, chlorine, bromine, iodine, astatine}
{Irshandy Aditya Wicaksana, 2241720148}
{Irshandy Aditya, 2241720148}
{}

Exited.
```

Langkah 2: menampilkan data set pada variabel halogens

Langkah 3: tidak error

Praktikum 3:

```
1 void main(){
15     var mhs1 = Map<String, String>();
16     gifts['first'] = 'partridge';
17     gifts['second'] = 'turtledoves';
18     gifts['fifth'] = 'golden rings';
19     gifts['six'] = 'Irshandy Aditya Wicaksana';
20     gifts['seven'] = '2241720148';
21
22     var mhs2 = Map<int, String>();
23     nobleGases[2] = 'helium';
24     nobleGases[10] = 'neon';
25     nobleGases[18] = 'argon';
26     nobleGases[26] = 'Irshandy Aditya Wicaksana';
27     nobleGases[34] = '2241720148';
28
29     mhs1['nama'] = 'Irshandy Aditya Wicaksana';
30     mhs1['nim'] = '2241720148';
31     mhs2[10] = 'Irshandy Aditya Wicaksana';
32     mhs2[16] = '2241720148';
33
34     print(gifts);
35     print(nobleGases);
36     print(mhs1);
37     print(mhs2);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, !exclude, \escape)

```
{first: partridge, second: turtledoves, fifth: golden rings, six: Irshandy Aditya Wicaksana, seven: 2241720148}
{2: helium, 10: neon, 18: argon, 26: Irshandy Aditya Wicaksana, 34: 2241720148}

Exited.
{nama: Irshandy Aditya Wicaksana, nim: 2241720148}
{10: Irshandy Aditya Wicaksana, 16: 2241720148}
```

Langkah 2: tidak ada error

Langkah 3: menimpa value dari key yang dituju contohnya value dari key fifth pada array gifts dari angka 1 berubah menjadi golden rings

Praktikum 4:

```
1 void main(){
11 print(list1);
12 var list3 = [0, ...?list1];
13 print(list3.length);
14 var nim = [4,1,7,2,0,1,4,8];
15 var list = [2,2, ...nim];
16 print(list);
17
18 // 4
19 var promoActive = true;
20 var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
21 print(nav);
22
23 // 5
24 var login = 'Admin';
25 var nav2 = ['Home', 'Furniture', 'Plants', if (login case 'Manager') 'Inventory'];
26 print(nav2);
27
28 // 6
29 var listOfInts = [1, 2, 3];
30 var listOfStrings = ['#0', for (var i in listOfInts) '#$i'];
31 assert(listOfStrings[1] == '#1');
32 print(listOfStrings);
33 }
34
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2]
3
[2, 2, 4, 1, 7, 2, 0, 1, 4, 8]
[Home, Furniture, Plants, Outlet]
[Home, Furniture, Plants]
[#0, #1, #2, #3]

Exited.
```

Langkah 2: error karena tidak ada list1

Langkah 3: error karena array integer tidak bisa null

Langkah 4: variabel promoActive belum diinisialisasi. apabila true maka Outlet akan ditambahkan pada array, jika false tidak ditampilkan

Langkah 5: jika kondisi login adalah Manager, maka akan menampilkan Inventory pada saat print nav2, jika memiliki kondisi selain Manager, maka inventory tidak ditampilkan.

Langkah 6: untuk menggunakan perulangan for secara ringkas dan efisien tanpa harus yang memungkinkan menghasilkan elemen-elemen dalam koleksi berdasarkan kondisi dalam loop tanpa harus menambahkan elemen satu per satu secara manual

Praktikum 5:

```
Pertemuan 4 > Praktikum_5.dart > ...
1 void main(){
2
3 // 3
4 (int, int) tukar((int, int) record) {
5     var (a, b) = record;
6     return (b, a);
7 }
8 var hasil = tukar((record.a, 10));
9 print(hasil);
10
11 // 4
12 // Record type annotation in a variable declaration:
13 (String, int) mahasiswa;
14 mahasiswa = ('Irshandy Aditya Wicaksana', 2241720148);
15 print(mahasiswa);
16
17 // 5
18 var mahasiswa2 = ('Irshandy Aditya Wicaksana', a: 2241720148, b: true, 'last');
19
20 print(mahasiswa2.$1); // Prints 'first'
21 print(mahasiswa2.a); // Prints 2
22 print(mahasiswa2.b); // Prints true
23 print(mahasiswa2.$2); // Prints 'last'
24 }
25
26 PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
27 (first, last, a: 2, b: true)
28 (10, 2)
29
30 Exited.
31 (Irshandy Aditya Wicaksana, 2241720148)
32 Irshandy Aditya Wicaksana
33 2241720148
34 true
35 last
```

Langkah 2: record menyimpan semua tipe data dalam 1 variabel, error: kurang titik koma.

Langkah 3: menukar value dari parameter pertama dan kedua.

Langkah 4: Peletakan tipe data pada record harus sesuai seperti pada saat melakukan deklarasi variabel

Langkah 5: Menunjukkan cara mengambil data yang ada pada records.

2. Jelaskan yang dimaksud Functions dalam bahasa Dart!

Jawaban :

Functions adalah sebuah blok kode yang dapat diberi nama dan digunakan untuk melakukan tugas tertentu yang memungkinkan kita untuk memisahkan logika program menjadi potongan kode yang lebih kecil, terstruktur, dan dapat digunakan kembali.

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

Jawaban :

Terdapat 4 jenis parameter yaitu:

- Positional Parameter: Parameter yang wajib diisi saat fungsi dipanggil. Contoh: void printInfo(String name, int age).
- Named Parameter: Parameter yang memungkinkan kita untuk memberikan argumen berdasarkan nama parameternya, sehingga saat memanggil fungsi, urutan argumen tidak penting. Contoh:

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

void main() {
    printInfo(name: 'John', age: 25); // Name: John, Age: 25
}
```

- Optional Positional Parameters: Parameter yang tidak wajib diisi pada saat hendak digunakan. Apabila tidak diisi, maka nilai dari parameter ini akan null. Contoh :

```
void greet(String name, [String? greeting]) {
    if (greeting != null) {
        print('$greeting, $name!');
    } else {
        print('Hello, $name!');
    }
}

void main() {
    greet('Alice'); // Hello, Alice!
    greet('Alice', 'Hi'); // Hi, Alice!
}
```

- Default Parameter: Parameter yang diberikan nilai default. Apabila parameter tidak diisi pada saat fungsi digunakan, maka isi dari parameter adalah nilai default itu sendiri. Contoh:

```
void greet(String name, {String greeting = 'Hello'}) {
    print('$greeting, $name!');
}

void main() {
    greet('Alice'); // Hello, Alice!
    greet('Alice', greeting: 'Hi'); // Hi, Alice!
}
```

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

Jawaban :

Functions sebagai first-class object berarti fungsi diperlakukan seperti objek biasa dan dapat dimanipulasi dengan cara yang sama seperti objek lain. Dengan begitu, fungsi dapat disimpan dalam variabel, dikirim sebagai argument ke fungsi lain, dikembalikan dari fungsi lain, dan disimpan dalam koleksi seperti list dan map.

Contoh sintak:

```
Function createFunction() {
    return (int x) => x * 2;
}
```

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

Jawaban :

Anonymous functions adalah fungsi yang tidak memiliki nama dan biasanya digunakan untuk operasi sederhana atau sekali pakai. Contoh:

```
(parameter) {
    // kode yang dieksekusi
};
```

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

Jawaban :

Lexical scope adalah aturan penentuan akses variabel berdasarkan posisi fungsi dan kode saat penulisan.. Sedangkan lexical closures adalah fungsi yang mengakses variabel dari lingkup luar, menyimpan referensi ke variabel tersebut bahkan setelah lingkup asalnya selesai.

Contoh lexical scope:

```
void main() {
    int x = 10;

    void innerFunction() {
        // Bisa mengakses variabel x karena berada di lingkup luar (main)
        print(x);
    }

    innerFunction(); // Output: 10
}
```

Contoh Lexical closures:

```
Function createCounter() {
    int counter = 0;

    // Mengembalikan closure
    return () {
        counter++;
        return counter;
    };
}

void main() {
    var myCounter = createCounter();

    print(myCounter()); // Output: 1
    print(myCounter()); // Output: 2
    print(myCounter()); // Output: 3
}
```

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

Jawaban :

Ada beberapa cara untuk membuat return multiple value di Functions. Salah satunya menggunakan record. Contoh cara pembuatannya ada pada praktikum 5:

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

(int, int) tukar((int, int) record) {
    var (a, b) = record;
    return (b, a);
}

var hasil = tukar((record.a, 10));
print(hasil);
```