

Nama : Muhammad Bagus Indrawan

Kelas : TI-3A/20

NIM : 2241720217

## Praktikum 1: Eksperimen Tipe Data List

### Langkah 1:

Ketik atau salin kode program berikut ke dalam void main().

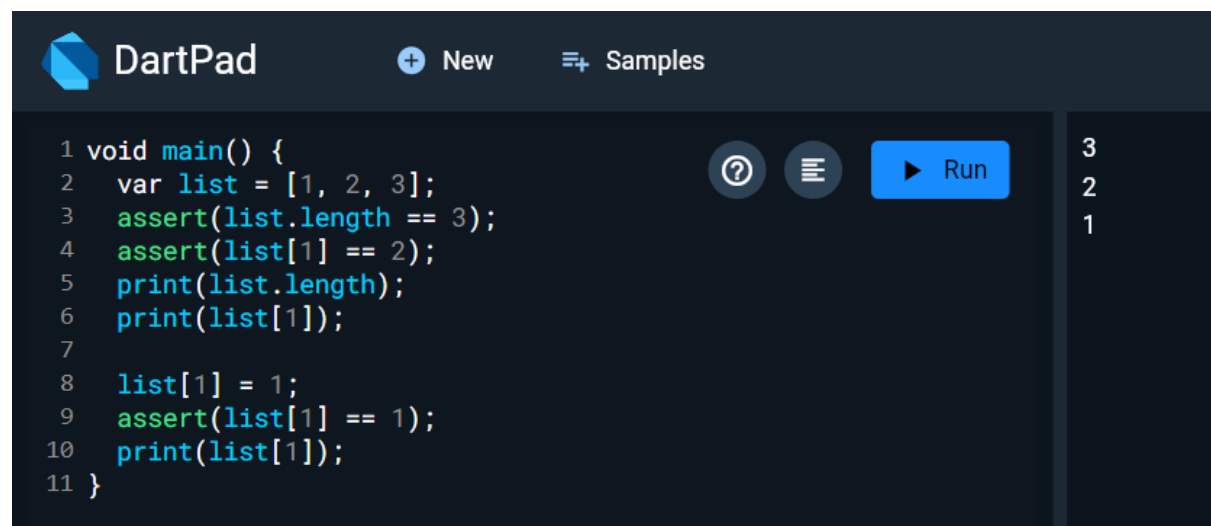
```
var list = [1, 2, 3];
assert(list.length == 3);
assert(list[1] == 2);
print(list.length);
print(list[1]);

list[1] = 1;
assert(list[1] == 1);
print(list[1]);
```

### Langkah 2:

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan!

Kode Dart di atas membuat list dengan nilai [1, 2, 3], memverifikasi panjangnya (3) dan nilai elemen indeks 1 (2) menggunakan assert. Kemudian mencetak panjang dan elemen tersebut. Setelah mengubah nilai elemen indeks 1 menjadi 1, kode memeriksa lagi dengan assert dan mencetak nilai baru. Jika semua kondisi assert terpenuhi, outputnya seperti dibawah



The screenshot shows the DartPad web interface. The code editor on the left 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 }
```

On the right side, the output of the code is displayed as a vertical list of values: 3, 2, and 1. The interface also includes a 'Run' button and a 'Samples' menu.

### 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.

Muncul output seperti berikut :



```
1 void main() {
2   final List<String?> list = List.filled(5, null);
3   list[1] = 'Muhammad Bagus Indrawan';
4   list[2] = '2241720217';
5
6   for (var i = 0; i < list.length; i++) {
7     print('Index $i: ${list[i]}');
8   }
9 }
```

Index 0: null  
Index 1: Muhammad Bagus Indrawan  
Index 2: 2241720217  
Index 3: null  
Index 4: null

## Praktikum 2: Eksperimen Tipe Data Set

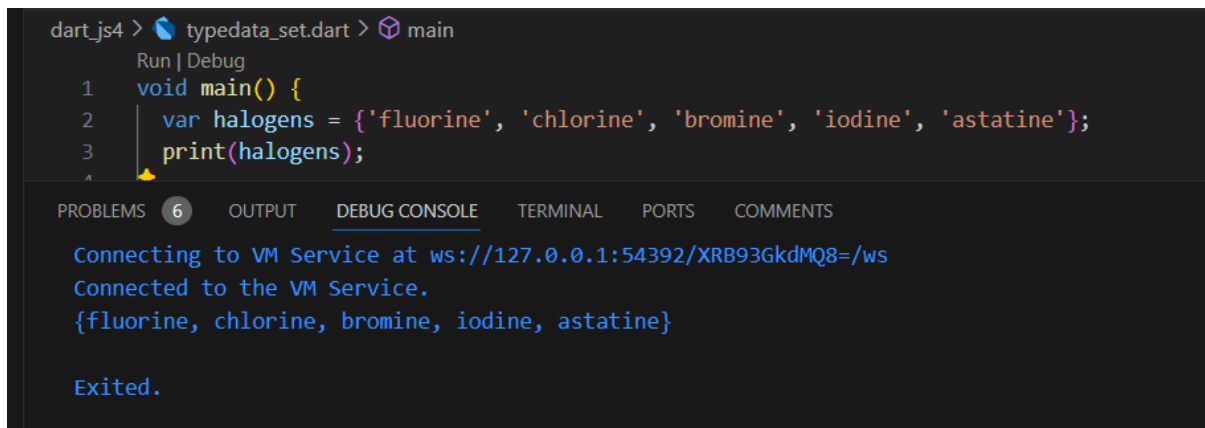
### Langkah 1:

Ketik atau salin kode program berikut ke dalam fungsi main().

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

### Langkah 2:

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.



```
dart.js4 > typedata_set.dart > main
Run | Debug
1 void main() {
2   var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};
3   print(halogens);
4 }

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:54392/XRB93GkdMQ8=/ws
Connected to the VM Service.
{fluorine, chlorine, bromine, iodine, astatine}
Exited.
```

Muncul output daftar unsur halogen yang dicetak dalam bentuk set, tidak ada error.

### Langkah 3:

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

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

print(names1);
print(names2);
print(names3);
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki namun tetap menggunakan ketiga variabel

tersebut. Tambahkan elemen nama dan NIM Anda pada kedua variabel Set tersebut dengan dua fungsi berbeda yaitu `.add()` dan `.addAll()`. Untuk variabel Map dihapus, nanti kita coba di praktikum selanjutnya.

Tidak terjadi error, hanya saja terdapat 3 string yang belum terisi.

```
dart_js4 > typedata_set.dart > ...
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 = {};
7   var names3 = {};
8
9   print(names1);
10  print(names2);
11  print(names3);
12 }
13 |

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:54949/CPvldzFQISA=/ws
Connected to the VM Service.
{fluorine, chlorine, bromine, iodine, astatine}
3 {}
Exited.
```

Berikut adalah modifikasi kode dan outputnya :

```
dart_js4 > typedata_set.dart > main
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 = {};
7   var names3 = {};
8
9   names1.add('Muhammad Bagus Indrawan');
10  names1.add('2241720217');
11
12  names2.addAll(['TI3A', '20']);
13
14  print(names1);
15  print(names2);
16  print(names3);
17 }
18

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:54880/JBDhMqzXDgg=/ws
Connected to the VM Service.
{fluorine, chlorine, bromine, iodine, astatine}
{Muhammad Bagus Indrawan, 2241720217}
{TI3A, 20}
{}
Exited.
```

## Praktikum 3: Eksperimen Tipe Data Maps

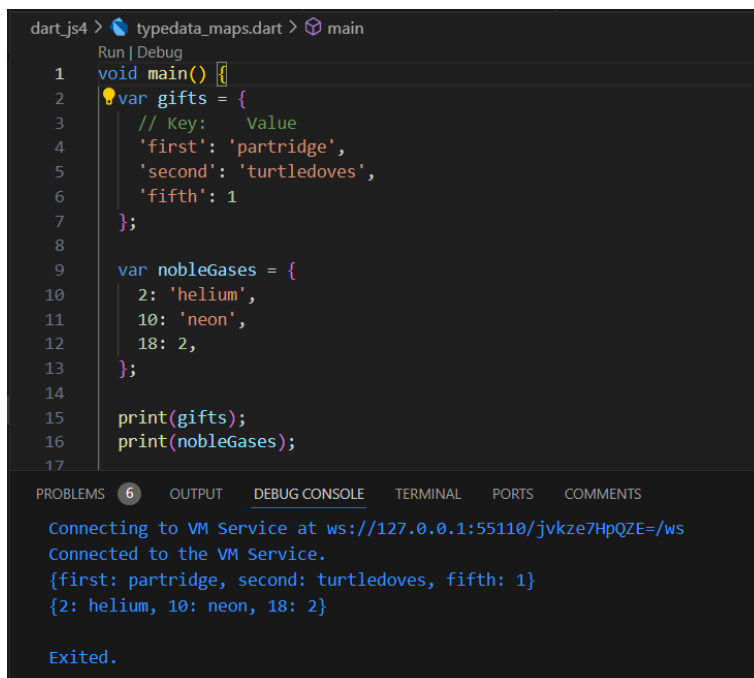
### Langkah 1:

Ketik atau salin kode program berikut ke dalam fungsi main().

```
var gifts = {  
  // Key:    Value  
  'first': 'partridge',  
  'second': 'turtledoves',  
  'fifth': 1  
};  
  
var nobleGases = {  
  2: 'helium',  
  10: 'neon',  
  18: 2,  
};  
  
print(gifts);  
print(nobleGases);
```

### Langkah 2:

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.



```
dart_js4 > typedata_maps.dart > main  
Run | Debug  
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  
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS  
Connecting to VM Service at ws://127.0.0.1:55110/jvkze7HpQZE=/ws  
Connected to the VM Service.  
{first: partridge, second: turtledoves, fifth: 1}  
{2: helium, 10: neon, 18: 2}  
Exited.
```

Saat kode dijalankan, hasil yang ditampilkan adalah isi kedua map. Map gifts akan mencetak daftar hadiah, sedangkan nobleGases akan menampilkan nama-nama gas mulia dengan nomor atomnya. Kode ini berjalan tanpa error.

### Langkah 3:

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

```

var mhs1 = Map<String, String>();
gifts['first'] = 'partridge';
gifts['second'] = 'turtledoves';
gifts['fifth'] = 'golden rings';

var mhs2 = Map<int, String>();
nobleGases[2] = 'helium';
nobleGases[10] = 'neon';
nobleGases[18] = 'argon';

```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Tambahkan elemen nama dan NIM Anda pada tiap variabel di atas (gifts, nobleGases, mhs1, dan mhs2). Dokumentasikan hasilnya dan buat laporannya!

```

dart_js4 > typedata_maps.dart > main
Run | Debug
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  var mhs1 = Map<String, String>();
19  gifts['first'] = 'partridge';
20  gifts['second'] = 'turtledoves';
21  gifts['fifth'] = 'golden rings';

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```

Connecting to VM Service at ws://127.0.0.1:55835/DyxjV1faNWg=/ws
Connected to the VM Service.
{first: partridge, second: turtledoves, fifth: 1}
{2: helium, 10: neon, 18: 2}
2 {}
Exited.

```

Tidak terjadi error, hanya saja terdapat map kosong pada output yang terjadi karena mhs1 dan mhs2 didefinisikan sebagai map kosong, tetapi tidak diisi dengan elemen apapun sebelum dicetak.

Berikut adalah kode perbaikannya :

```
dart_js4 > typedata_maps.dart > main
Run | Debug
1 void main() {
2   var gifts = {
3     // Key: Value
4     'first': 'partridge',
5     'second': 'turtledoves',
6     'fifth': 'golden rings' // Memperbarui nilai di sini
7   };
8
9   var nobleGases = {
10    2: 'helium',
11    10: 'neon',
12    18: 'argon' // Memperbarui nilai di sini
13  };
14
15  print(gifts);
16  print(nobleGases);
17
18  var mhs1 = Map<String, String>();
19  mhs1['name'] = 'Muhammad Bagus Indrawan';
20  mhs1['NIM'] = '2241720217';
21
22  var mhs2 = Map<int, String>();
23  mhs2[1] = 'Muhammad Bagus Indrawan';
24  mhs2[2] = '2241720217';
25
26  print(mhs1);
27  print(mhs2);
28
29 }
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE ... Filter (e.g. text, lexclude,...)

Connecting to VM Service at ws://127.0.0.1:56114/WOVJIADK9EE=ws  
Connected to the VM Service.  
{first: partridge, second: turtledoves, fifth: golden rings}  
{2: helium, 10: neon, 18: argon}  
{name: Muhammad Bagus Indrawan, NIM: 2241720217}  
{1: Muhammad Bagus Indrawan, 2: 2241720217}  
Exited.

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

### Langkah 1:

Ketik atau salin kode program berikut ke dalam fungsi main().

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

### Langkah 2:

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

```
dart_js4 > typedata_list_sco.dart > main
Run | Debug
1 void main() {
2   var list = [1, 2, 3];
3   var list2 = [0, ...list1];
4   print(list1);
5   print(list2);
6   print(list2.length);
}
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS Filter (e.g. text, lexclude, \escape)

Connecting to VM Service at ws://127.0.0.1:56374/7rujeIBweKU=ws  
Connected to the VM Service.  
dart\_js4/typedata\_list\_sco.dart:3:22: Error: Undefined name 'list1'.  
var list2 = [0, ...list1];  
^^^^^  
dart\_js4/typedata\_list\_sco.dart:4:9: Error: Undefined name 'list1'.  
print(list1);  
^^^^^  
dart\_js4/typedata\_list\_sco.dart:3:22: Error: An expression whose value can be 'null' must be null-checked before it can be dereferenced.  
var list2 = [0, ...list1];  
^  
dart\_js4/typedata\_list\_sco.dart:3:22: Error: Unexpected type 'invalid-type' of a spread. Expected 'dynamic' or an Iterable.  
var list2 = [0, ...list1];  
^  
Failed to connect/initialize debugger for http://127.0.0.1:56372/hdmqHJH56VQ=/:  
getVM: (-32000) Service connection disposed  
Exited.

Terjadi error karena kesalahan dalam penamaan variabel “list” yang seharusnya dinamai list1, sehingga variabel menjadi tidak terbaca, berikut adalah perbaikan kodenya

```
dart_js4 > typedata_list_sco.dart > main
Run | Debug
1 void main() {
2   var list1 = [1, 2, 3];
3   var list2 = [0, ...list1];
4   print(list1);
5   print(list2);
6   print(list2.length);
}

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:55437/Z0wdjvAXnx8=/ws
Connected to the VM Service.
[1, 2, 3]
[0, 1, 2, 3]
4
Exited.
```

Kode yang diberikan mendefinisikan dua list di Dart, yaitu list1 dan list2. List list1 berisi tiga elemen, yaitu 1, 2, dan 3. Kemudian, list list2 dibuat dengan menggunakan spread operator (...) untuk menggabungkan elemen 0 di awal dengan semua elemen dari list1, sehingga list2 berisi elemen 0, 1, 2, dan 3. Kode ini kemudian mencetak isi dari list1, list2, dan panjang dari list2, yang hasilnya menunjukkan bahwa list1 berisi [1, 2, 3], list2 berisi [0, 1, 2, 3], dan panjang list2 adalah 4. Tidak terjadi error.

### Langkah 3:

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

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

Apa yang terjadi ? Jika terjadi error, silakan perbaiki.

Tambahkan variabel list berisi NIM Anda menggunakan Spread Operators. Dokumentasikan hasilnya dan buat laporannya!

```
dart_js4 > typedata_list_sco.dart > main
Run | Debug
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   // langkah 2 -> perbaikan pada penamaan atribut list1
8
9   list1 = [1, 2, 3];
10  print(list1);
11  var list3 = [0, ...?list1];
12  print(list3.length);
13 }

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:56556/44GjsqlZXs4=/ws
Connected to the VM Service.
[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2, 3]
4
Exited.
```

Kode yang diberikan mendefinisikan beberapa list di Dart menggunakan spread operator. List list1 berisi elemen 1, 2, dan 3, dan list list2 dibuat dengan menambahkan 0 di depan, menghasilkan [0, 1, 2, 3]. Kode mencetak kedua list dan panjang list2 dengan benar. List list3 juga dibuat dengan menggunakan spread operator, tetap menghasilkan [0, 1, 2, 3].

Menambahkan variabel list berisi NIM Anda menggunakan Spread Operators

```
dart_js4 > typedata_list_sco.dart > main
Run | Debug
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   // langkah 2 -> perbaikan pada penamaan atribut list1
9   list1 = [1, 2, 3];
10  print(list1);
11  var list3 = [0, ...?list1];
12  print(list3.length);
13
14  // Menambahkan NIM menggunakan Spread Operator
15  var nim = ['2241720217'];
16  var listWithNim = [...list3, ...nim];
17  print(listWithNim);
18 }
19

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:56712/AewE08Mj8wo=/ws
Connected to the VM Service.
[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2, 3]
4
[0, 1, 2, 3, 2241720217]
Exited.
```



List baru `listWithNim` dibuat untuk menambahkan NIM dengan cara menggabungkan `list3` dan list yang berisi NIM. Jika NIM asli ditambahkan, output menunjukkan hasil penggabungan semua list seperti di atas.

#### Langkah 4:

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

```
var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
print(nav);
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Tunjukkan hasilnya jika variabel `promoActive` ketika `true` dan `false`.

```
dart js4 > typedata_list_sco.dart > main
1 void main() {
2
3
4
5
6
7 // langkah 2 -> perbaikan pada penamaan atribut list1
8 list1 = [1, 2, 3];
9 print(list1);
10
11 var list3 = [0, ...?list1];
12 print(list3.length);
13
14 // Menambahkan NIM menggunakan Spread Operator
15 var nim = ['2241720217'];
16 var listWithNim = [...list3, ...nim];
17 print(listWithNim);
18
19 var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
20 print(nav);
21 }
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:56920/vCGVX71gvJ4=/ws  
Connected to the VM Service.  
dart\_js4/typedata\_list\_sco.dart:19:49: Error: Undefined name 'promoActive'.  
var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];  
^^^^^^^^^^^^^^

Kode di atas terjadi error karena variabel `promoActive` belum diinisialisasi. Berikut adalah hasil perbaikannya.

- Variable `promoActive` = `true`

```
19 var promoActive = true;
20 var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
21 print(nav);
22 }
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:57077/FI2\_y9oasUQ=/ws  
Connected to the VM Service.  
[1, 2, 3]  
[0, 1, 2, 3]  
4  
[1, 2, 3]  
4  
[0, 1, 2, 3, 2241720217]  
[Home, Furniture, Plants, Outlet]  
  
Exited.

- Variable promoActive = false

```
19   var promoActive = false;
20   var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
21   print(nav);
22 }
23
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:57077/FI2\_y9oasUQ=/ws  
Connected to the VM Service.  
[1, 2, 3]  
[0, 1, 2, 3]  
4  
[1, 2, 3]  
4  
[0, 1, 2, 3, 2241720217]  
[Home, Furniture, Plants, Outlet]  
  
Exited.

### Langkah 5:

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

```
var nav2 = ['Home', 'Furniture', 'Plants', if (login case 'Manager') 'Inventory'];
print(nav2);
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Tunjukkan hasilnya jika variabel login mempunyai kondisi lain.

```
23   var nav2 = ['Home', 'Furniture', 'Plants', if (login case 'Manager') 'Inventory'];
24   print(nav2);
25 }
26
```

Terminal (Ctrl+)

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:57321/h-XTikHt8Lg=/ws  
Connected to the VM Service.  
dart\_js4/typeddata\_list\_sco.dart:23:50: Error: Undefined name 'login'.  
var nav2 = ['Home', 'Furniture', 'Plants', if (login case 'Manager') 'Inventory'];  
^^^^

Kode di atas terjadi error karena variabel login belum diinisialisasi. Berikut adalah hasil perbaikannya.

```
22
23   var login = 'Manager';
24   var nav2 = ['Home', 'Furniture', 'Plants', if (login case 'Manager') 'Inventory'];
25   print(nav2);
26 }
27
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:57404/79T8NgPw39M=/ws  
Connected to the VM Service.  
[1, 2, 3]  
[0, 1, 2, 3]  
4  
[1, 2, 3]  
4  
[0, 1, 2, 3, 2241720217]  
[Home, Furniture, Plants]  
[Home, Furniture, Plants, Inventory]  
  
Exited.

Dan berikut adalah hasilnya jika variabel login mempunyai kondisi lain.

```
23     var login = 'User';
24     var nav2 = ['Home', 'Furniture', 'Plants', if (login case 'Manager') 'Inventory'];
25     print(nav2);
26 }
27
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:57466/\_MwRm7XWQys=/ws  
Connected to the VM Service.  
[1, 2, 3]  
[0, 1, 2, 3]  
4  
[1, 2, 3]  
4  
[0, 1, 2, 3, 2241720217]  
2 [Home, Furniture, Plants]  
  
Exited.

#### Langkah 6:

Tambahkan kode program berikut, lalu coba eksekusi (Run) kode Anda.

```
var listOfInts = [1, 2, 3];
var listOfStrings = ['#0', for (var i in listOfInts) '#$i'];
assert(listOfStrings[1] == '#1');
print(listOfStrings);
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Jelaskan manfaat **Collection For** dan dokumentasikan hasilnya.

### Praktikum 5: Eksperimen Tipe Data Records

#### Langkah 1:

Ketik atau salin kode program berikut ke dalam fungsi main().

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

#### Langkah 2:

Silakan coba eksekusi (Run) kode pada langkah 1 tersebut. Apa yang terjadi? Jelaskan! Lalu perbaiki jika terjadi error.

```
dart_js4 > typedata_records.dart > main
Run | Debug
1 void main() {
2   var record = ('first', a: 2, b: true, 'last');
3   print(record)
}

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:55537/4NPYZMPDDQ8=/ws
Connected to the VM Service.
dart_js4/typedata_records.dart:3:15: Error: Expected ';' after this.
  print(record)
                ^

Exited (254).
```

Kode di atas terjadi error karena pada perintah sintaks print pada akhir perintahnya kurang tanda semicolon ";".

Kode yang benar :

```
dart_js4 > typedata_records.dart > main
Run | Debug
1 void main() {
2   var record = ('first', a: 2, b: true, 'last');
3   print(record);
}

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:62888/CZj077HC220=/ws
Connected to the VM Service.
(first, last, a: 2, b: true)

Exited.
```

Pada kode program di atas, tipe data record digunakan untuk menyimpan beberapa nilai yang memiliki tipe data berbeda dalam satu struktur tunggal. Tipe data record memudahkan pengelompokan beberapa nilai dengan berbagai tipe data seperti string, integer, dan boolean tanpa memerlukan kelas atau struktur data yang lebih kompleks. Ketika `print(record)` dieksekusi, seluruh isi record akan ditampilkan, yakni `('first', a: 2, b: true, 'last')`.

### Langkah 3:

Tambahkan kode program berikut di luar scope `void main()`, lalu coba eksekusi (Run) kode Anda.

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

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Gunakan fungsi tukar() di dalam main() sehingga tampak jelas proses pertukaran value field di dalam Records.

```
dart_js4 > typedata_records.dart > main
Run | Debug
1 void main() {
2   var record = ('first', a: 2, b: true, 'last');
3   print(record);
4 }
5
6 (int, int) tukar((int, int) record) {
7   var (a, b) = record;
8   return (b, a);
9 }

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:63169/FfBfVbf9P7A=/ws
Connected to the VM Service.
(first, last, a: 2, b: true)

Exited.
```

Setelah memasukkan kode program pada langkah ke-3 tidak akan terjadi error. Namun fungsi tukar juga tidak dipanggil oleh variabel manapun, jadi untuk menunjukkan proses pertukaran value field di dalam records perlu dibuat variabel record2 yang menyimpan nilai int. Kemudian akan dijalankan fungsi tukar terhadap variabel record2 pada variabel hasil, sehingga nilai int pada variabel record2 akan ditukar posisinya menggunakan fungsi tukar dan variabel setelah ditukar tersebut akan disimpan dalam variabel hasil dan ditampilkan keduanya, yakni sebelum diubah posisinya dan setelah diubah seperti gambar di atas.

Fungsi tukar() didefinisikan untuk menukar dua nilai integer yang terdapat dalam record. Fungsi ini menerima record dengan dua elemen integer, menggunakan teknik destructuring untuk mengakses nilai-nilai tersebut, dan kemudian mengembalikan record baru dengan posisi kedua elemen yang sudah ditukar.

#### Langkah 4:

Tambahkan kode program berikut di dalam scope void main(), lalu coba eksekusi (Run) kode Anda.

```
// Record type annotation in a variable declaration:
(String, int) mahasiswa;
print(mahasiswa);
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Inisialisasi field nama dan NIM Anda pada variabel record mahasiswa di atas. Dokumentasikan hasilnya dan buat laporannya!

```
dart_js4 > typedata_records.dart > main
Run | Debug
1 void main() {
2   var record = ('first', a: 2, b: true, 'last');
3   print(record);
4   (String, int) mahasiswa;
5   print(mahasiswa);
6 }
7
8 (int, int) tukar((int, int) record) {
9   var (a, b) = record;
10  return (b, a);
}

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:63340/S0nmcw96aW8=/ws
Connected to the VM Service.
dart_js4/typedata_records.dart:5:9: Error: Non-nullable variable 'mahasiswa' must be assigned before it can be used.
  print(mahasiswa);
        ^^^^^^^^^
```

Berdasarkan output di atas, kode terjadi error karena saat pemanggilan variabel mahasiswa pada perintah print, variabel mahasiswa masih belum memiliki nilai, sehingga untuk menjalankan kode program di atas perlu dimasukkan nilai dari variabel mahasiswa terlebih dahulu yang sesuai dengan tipe data ditentukan yakni (String, int).

Kode yang benar :

```
dart_js4 > typedata_records.dart > main
Run | Debug
1 void main() {
2   var record = ('first', a: 2, b: true, 'last');
3   print(record);
4   (String, int) mahasiswa;
5   mahasiswa = ('Muhammad Bagus Indrawan', 2241720217);
6   print(mahasiswa);
7 }
8
9 (int, int) tukar((int, int) record) {
10  var (a, b) = record;
}

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:63415/mcfuVi8JqJ4=/ws
Connected to the VM Service.
(first, last, a: 2, b: true)
(Muhammad Bagus Indrawan, 2241720217)

Exited.
```

## Langkah 5:

Tambahkan kode program berikut di dalam scope void main(), lalu coba eksekusi (Run) kode Anda.

```
var mahasiswa2 = ('first', a: 2, b: true, 'last');

print(mahasiswa2.$1); // Prints 'first'
print(mahasiswa2.a); // Prints 2
print(mahasiswa2.b); // Prints true
print(mahasiswa2.$2); // Prints 'last'
```

Apa yang terjadi ? Jika terjadi error, silakan perbaiki. Gantilah salah satu isi record dengan nama dan NIM Anda, lalu dokumentasikan hasilnya dan buat laporannya!

```
dart.js4 > typedata_records.dart > main
1 void main() {
2   print(mahasiswa);
3
4   var mahasiswa2 = ('first', a: 2, b: true, 'last');
5
6   print(mahasiswa2.$1); // Prints 'first'
7   print(mahasiswa2.a); // Prints 2
8   print(mahasiswa2.b); // Prints true
9   print(mahasiswa2.$2); // Prints 'last'
10 }
11
12
13
14
15
16
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:63442/pgs-O\_TQeEE=/ws  
Connected to the VM Service.  
(first, last, a: 2, b: true)  
(Muhammad Bagus Indrawan, 2241720217)  
first  
2  
true  
last  
  
Exited.

Pada kode program di atas, sebuah record mahasiswa2 dideklarasikan dengan elemen posisional dan elemen bernama. Elemen posisional ditunjukkan pada elemen 'first' dan 'last', serta elemen bernama yakni a: 2 dan b: true. Pada perintah print, mahasiswa2.\$1 dan mahasiswa2.\$2 digunakan untuk mengakses elemen posisional dari variabel mahasiswa2 posisi pertama dan kedua, sedangkan mahasiswa2.a dan mahasiswa2.b digunakan untuk mengakses elemen dengan nama a dan b yang sudah ada pada variabel mahasiswa2.

```
dart.js4 > typedata_records.dart > main
1 void main() {
2   print(mahasiswa);
3
4   var mahasiswa2 = ('first', nama: 'Muhammad Bagus Indrawan', nim: 2241720217, 'last');
5
6   print(mahasiswa2.$1);
7   print(mahasiswa2.nama);
8   print(mahasiswa2.nim);
9   print(mahasiswa2.$2);
10 }
11
12
13
14
15
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:63767/3PNTlnQ0oCQ=/ws  
Connected to the VM Service.  
(first, last, a: 2, b: true)  
(Muhammad Bagus Indrawan, 2241720217)  
first  
Muhammad Bagus Indrawan  
2241720217  
last  
  
Exited.

## Tugas Praktikum

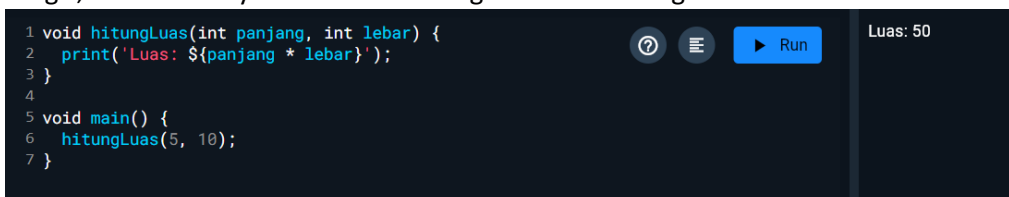
1. Silakan selesaikan Praktikum 1 sampai 5, lalu dokumentasikan berupa screenshot hasil pekerjaan Anda beserta penjelasannya!
2. Jelaskan yang dimaksud Functions dalam bahasa Dart!

Dalam bahasa Dart, Functions adalah blok kode yang dapat dipanggil untuk melakukan suatu tugas tertentu. Sebuah fungsi biasanya terdiri dari kumpulan instruksi yang menjalankan suatu operasi, menerima input (parameter), dan bisa mengembalikan output (nilai hasil). Di Dart, fungsi bisa dideklarasikan menggunakan kata kunci `void` jika tidak mengembalikan nilai, atau dengan tipe data tertentu jika fungsi tersebut mengembalikan nilai.

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

1. Positional Parameter


Positional parameter adalah parameter yang wajib disediakan saat memanggil fungsi, dan urutannya harus sesuai dengan deklarasi fungsi.



```
1 void hitungLuas(int panjang, int lebar) {  
2   print('Luas: ${panjang * lebar}');  
3 }  
4  
5 void main() {  
6   hitungLuas(5, 10);  
7 }
```

2. Named Parameter

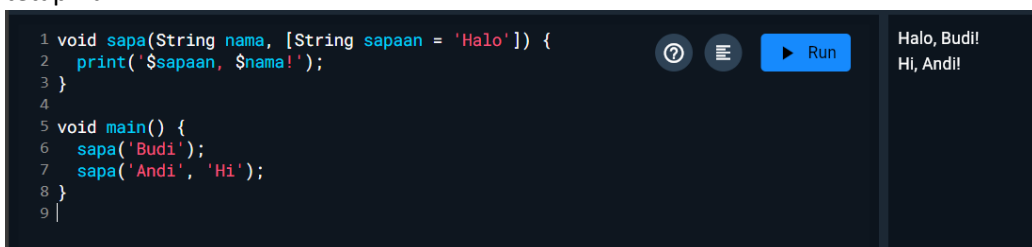
Named parameter adalah parameter yang disebutkan dengan nama saat fungsi dipanggil, sehingga urutan tidak menjadi masalah. Named parameter sering digunakan untuk membuat kode lebih mudah dipahami.



```
1  
2  
3 void hitungVolume({required int panjang, required int lebar, required int  
   tinggi}) {  
4   print('Volume: ${panjang * lebar * tinggi}');  
5 }  
6  
7 void main() {  
8   hitungVolume(panjang: 5, lebar: 10, tinggi: 2);  
9 }
```

3. Optional Positional Parameter

Optional positional parameter adalah parameter yang tidak wajib diberikan saat fungsi dipanggil. Jika parameter tidak diberikan, bisa menggunakan nilai default atau tetap `null`.



```
1 void sapa(String nama, [String sapaan = 'Halo']) {  
2   print('$sapaan, $nama!');  
3 }  
4  
5 void main() {  
6   sapa('Budi');  
7   sapa('Andi', 'Hi');  
8 }  
9 |
```

4. Optional Named Parameter (dengan nilai default)

Sama seperti named parameter, tapi dengan nilai default jika tidak diberikan.



```

1 void buatPesanan({String item = 'Teh', int jumlah = 1}) {
2   print('Pesanan: $jumlah $item');
3 }
4
5 void main() {
6   buatPesanan();
7   buatPesanan(item: 'Kopi', jumlah: 2);
8 }
9

```

Pesanan: 1 Teh  
Pesanan: 2 Kopi

#### 5. Required Named Parameter

Required Named Parameter adalah parameter yang berfungsi untuk memastikan bahwa named parameter tertentu harus diberikan, kita dapat menggunakan `required` dalam deklarasi fungsi.

```

1 void hitungPersegi({required int sisi}) {
2   print('Luas persegi: ${sisi * sisi}');
3 }
4
5 void main() {
6   hitungPersegi(sisi: 5);
7 }
8

```

Luas persegi: 25

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

**Functions sebagai first-class objects** berarti fungsi dalam suatu bahasa pemrograman diperlakukan seperti objek lainnya. Ini artinya, fungsi bisa disimpan dalam variabel, diteruskan sebagai argumen ke fungsi lain, atau dikembalikan dari fungsi lain.

**Contoh implementasi :**

```

dart_js4 > function_fco.dart > main
1 // Mendefinisikan fungsi
2 void sayHello() {
3   print("Hello, World!");
4 }
5
6 // Fungsi yang menerima fungsi sebagai parameter
7 void executeFunction(Function func) {
8   func(); // Menjalankan fungsi yang diteruskan
9 }
10
11 // Fungsi yang mengembalikan fungsi
12 Function createMultiplier(int multiplier) {
13   return (int value) => value * multiplier; // Mengembalikan fungsi
14 }
15
16 Run | Debug
17 void main () {
18   // Menyimpan fungsi dalam variabel
19   var greeting = sayHello;
20
21   // Memanggil fungsi melalui variabel
22   greeting(); // Output: Hello, World!
23
24   // Menggunakan fungsi sebagai argumen
25   executeFunction(sayHello); // Output: Hello, World!
26
27   // Menggunakan fungsi yang dikembalikan
28   var double = createMultiplier(2);
29   print(double(5)); // Output: 10
30 }

```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:50650/tiqHPRh0so=/ws  
Connected to the VM Service.

2 Hello, World!  
10

Exited.

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

Fungsi anonim, atau juga dikenal sebagai lambda atau closure, adalah fungsi yang tidak memiliki nama. Fungsi ini sering digunakan ketika Anda memerlukan fungsi sekali pakai yang sederhana dan tidak perlu mendefinisikan nama fungsi. Dalam Dart, fungsi anonim berguna saat Anda ingin membuat fungsi cepat yang akan digunakan sebagai argumen dalam fungsi lain, seperti dalam operasi `forEach`, `map`, atau pengurutan (*sorting*).

```
dart_js4 > anonymous_function.dart > main
Run | Debug
1 void main() {
2   var list = [1, 2, 3, 4, 5];
3
4   // Menggunakan fungsi anonim dalam forEach
5   list.forEach((item) {
6     print(item * 2); // Menggandakan setiap elemen dan mencetaknya
7   });
8
9   // Menggunakan sintaks panah untuk fungsi anonim
10  var doubled = list.map((item) => item * 2).toList();
11
12  print(doubled); // Output: [2, 4, 6, 8, 10]
13
14
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:50763/CYaCI40f-48=/ws
Connected to the VM Service.
2
4
6
8
10
[2, 4, 6, 8, 10]
Exited.
```

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

**Lexical scope** (juga dikenal sebagai *static scope*) adalah konsep di mana cakupan (*scope*) variabel ditentukan berdasarkan lokasi fisik kode dalam file saat ditulis. Dengan kata lain, variabel dan fungsi hanya dapat diakses di dalam blok kode tempat mereka dideklarasikan dan lingkup-lingkup di dalamnya, dan tidak bisa diakses dari luar cakupan tersebut.

#### Contoh Lexical Scope :

```
dart_js4 > lexical_scope.dart > main
Run | Debug
1 void main() {
2   var a = 10; // Variabel di scope utama
3
4   void innerFunction() {
5     print(a); // Bisa mengakses variabel a karena berada dalam lexical scope
6   }
7
8   innerFunction(); // Output: 10
9   print(a); // Output: 10
10
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:50888/IRgr1XZie6g=/ws
Connected to the VM Service.
2 10
Exited.
```

**Lexical closure** adalah konsep yang lebih lanjut dari lexical scope, di mana sebuah fungsi (closure) dapat "menangkap" atau "mengikat" variabel-variabel dari lingkup luar tempat fungsi tersebut didefinisikan, bahkan setelah lingkup luar tersebut selesai dieksekusi.

**Contoh Lexical Closure :**

```
dart_js4 > lexical_closure.dart > main
Run | Debug
1 void main() {
2   var a = 5; // Variabel dalam lexical scope utama
3
4   Function multiplyBy(int multiplier) {
5     // Closure yang menangkap variabel dari lingkup luar
6     return (int value) {
7       return value * multiplier * a; // Mengakses variabel multiplier dan a
8     };
9   }
10
11   var multiplyBy2 = multiplyBy(2); // Closure menangkap multiplier = 2 dan a = 5
12   print(multiplyBy2(10)); // Output: 100 (10 * 2 * 5)
13
14   a = 6; // Mengubah nilai a
15   print(multiplyBy2(10)); // Output: 120 (10 * 2 * 6)
16 }
17

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Connecting to VM Service at ws://127.0.0.1:50941/vVrHtm15biA=/ws
Connected to the VM Service.
100
120
Exited.
```

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

Dalam Dart, ada beberapa cara untuk **mensimulasikan return multiple values**. Metode umum untuk melakukan ini adalah dengan mengembalikan tuple-like structure menggunakan list, map, atau class.

Contoh pembuatan:

- Menggunakan List  
List bisa digunakan untuk mengembalikan beberapa nilai, di mana setiap nilai disimpan sebagai elemen dalam List.

```
dart_js4 > return_multiple_value.dart > ...
1 //List
2 List<int> getTwoNumbers() {
3   return [5, 10]; // Mengembalikan dua angka sebagai List
4 }
5
Run | Debug
6 void main() {
7   var numbers = getTwoNumbers();
8   print(numbers[0]); // Output: 5
9   print(numbers[1]); // Output: 10
10 }
```

- Menggunakan Map  
Map digunakan untuk mengembalikan beberapa nilai dengan memberi label atau nama pada setiap nilai.

```
dart_js4 > return_multiple_value.dart > ...  
12 //Map  
13 Map<String, dynamic> getUserInfo() {  
14     return {  
15         'name': 'Muhammad Bagus Indrawan',  
16         'age': 21  
17     };  
18 }  
19  
Run | Debug  
20 void main() {  
21     var userInfo = getUserInfo();  
22     print(userInfo['name']); // Output: Muhammad Bagus Indrawan  
23     print(userInfo['age']); // Output: 21  
24 }
```

- Menggunakan Class  
Jika Anda ingin lebih terstruktur dan mendeskripsikan lebih baik nilai-nilai yang dikembalikan, Anda bisa membuat **class** untuk menampung nilai-nilai tersebut.

```
dart_js4 > return_multiple_value.dart > ...  
26 //Class  
27 class Person {  
28     String name;  
29     int age;  
30  
31     Person(this.name, this.age); // Constructor untuk menyimpan nilai  
32 }  
33  
34 Person getPersonInfo() {  
35     return Person('Muhammad Bagus Indrawan', 21);  
36 }  
37  
Run | Debug  
38 void main() {  
39     var person = getPersonInfo();  
40     print(person.name); // Output: Muhammad Bagus Indrawan  
41     print(person.age); // Output: 21  
42 }
```

8. Kumpulkan berupa link commit repo GitHub pada tautan yang telah disediakan di grup Telegram!