

NAJWA AZZAHRA

TI-3C/19

2241720139

## PRAKTIKUM 1

### Langkah 1

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

Connecting to VM Service at ws://127.0.0.1:62914/6mtHmpesIqY=/ws  
Connected to the VM Service.

3  
2  
1

### Langkah 2

Kode tersebut akan menghasilkan output

- 3 adalah panjang dari list.
- 2 adalah elemen awal pada indeks 1 sebelum dimodifikasi.
- 1 adalah elemen pada indeks 1 setelah dimodifikasi.

### Langkah 3

```
1 void main() {
2     final List<String?> list = List.filled(5, null);
3
4     list[1] = "Najwa Azzahra";
5     list[2] = "2241720139";
6
7     assert(list.length == 5);
8     assert(list[1] == "Najwa Azzahra");
9     assert(list[2] == "2241720139");
10
11    print(list.length);
12    print(list[1]);
13    print(list[2]);
14 }
```

5  
Najwa Azzahra  
2241720139

## PRAKTIKUM 2

### Langkah 1

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

{fluorine, chlorine, bromine, iodine, astatine}

### Langkah 2

print(halogens);, Dart akan mencetak isi dari Set tersebut. Karena Set tidak menjamin urutan elemen, urutan elemen dalam output mungkin berbeda setiap kali kode dijalankan. Namun, semua elemen yang ada dalam Set akan dicetak.

### Langkah 3

```
void main() {  
  var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};  
  print(halogens);  
}  
  
var names1 = <String>{};  
Set<String> names2 = {}; // This works, too.  
var names3 = {}; // Creates a map, not a set.  
  
print(names1);  
print(names2);  
print(names3);
```

5 issues

- A function body must be provided. line 10, column 1
- Try adding a function body.
- The name 'print' is already defined. line 11, column 1
- Try renaming one of the declarations.
- A function body must be provided. line 11, column 1
- Try adding a function body.

Sehingga kode program diubah menjadi

```
void main() {  
  var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};  
  print(halogens);  
  
  var names1 = <String>{};  
  Set<String> names2 = {};  
  Set<String> names3 = {};  
  
  names1.add('Najwa Azzahra'); // replace with your name  
  names2.addAll(['2241720139']); // replace with your NIM  
  
  print(names1);  
  print(names2);  
  print(names3);  
}
```

{fluorine, chlorine, bromine, iodine, astatine}  
{Najwa Azzahra}  
{2241720139}  
{}

## PRAKTIKUM 3

### Langkah 1

```
void main() {
var gifts = {
  // Key:    Value
  'first': 'partridge',
  'second': 'turtledoves',
  'fifth': 1
};

var nobleGases = {
  2: 'helium',
  10: 'neon',
  18: 2,
};

print(gifts);
print(nobleGases);
}
```

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

### Langkah 2

- gifts akan mencetak Map dengan tiga entri yang mencakup kunci dan nilai yang ditetapkan.
- nobleGases akan mencetak Map dengan tiga entri yang juga mencakup kunci dan nilai yang ditetapkan.

### Langkah 3

```
void main() {
var gifts = {
  // Key:    Value
  'first': 'partridge',
  'second': 'turtledoves',
  'fifth': 1
};

var nobleGases = {
  2: 'helium',
  10: 'neon',
  18: 2,
};

print(gifts);
print(nobleGases);

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';
}
```

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

2 issues

Semua perubahan pada Map gifts dan nobleGases diterapkan dengan benar. Namun, deklarasi mhs1 dan mhs2 tidak digunakan

Tambahkan elemen nama dan NIM Anda pada tiap variabel di atas (gifts, nobleGases, mhs1, dan mhs2).

```

void main() {
    var gifts = {
        // Key: Value
        'first': 'partridge',
        'second': 'turtledoves',
        'fifth': 'golden rings',
        'sixth': 'Najwa Azzahra',
        'seventh': '2241720139',
    };

    var nobleGases = {
        2: 'helium',
        10: 'neon',
        18: 'argon',
        19: 'Najwa Azzahra',
        20: '2241720139',
    };

    print('gifts: $gifts');
    print('nobleGases: $nobleGases');

    var mhs1 = Map<String, String>();
    mhs1['first'] = 'partridge';
    mhs1['second'] = 'turtledoves';
    mhs1['fifth'] = 'golden rings';
    mhs1['Najwa Azzahra'] = '2241720139';

    var mhs2 = Map<int, String>();
    mhs2[2] = 'helium';
    mhs2[10] = 'neon';
    mhs2[18] = 'argon';
    mhs2[19] = 'Najwa Azzahra';
    mhs2[20] = '2241720139';

    print('mhs1: $mhs1');
    print('mhs2: $mhs2');
}

```

```

gifts: {first: partridge, second: turtledoves, fifth: golden rings, sixth: Najwa Azzahra, seventh: 2241720139}
nobleGases: {2: helium, 10: neon, 18: argon, 19: Najwa Azzahra, 20: 2241720139}
mhs1: {first: partridge, second: turtledoves, fifth: golden rings, Najwa Azzahra: 2241720139}
mhs2: {2: helium, 10: neon, 18: argon, 19: Najwa Azzahra, 20: 2241720139}

```

## PRAKTIKUM 4

### Langkah 1

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

compileDDC  
main.dart:4:7: Error: Undefined name 'list1'.  
print(list1);  
^^^^

- Variabel list1 digunakan dalam print(list1);, tetapi variabel ini tidak ada dalam kode. \
- list2 dideklarasikan dengan benar, tetapi ada kesalahan penamaan variabel saat mencetak.
- Penggunaan operator spread (...) dalam list2 sudah benar. Ini menyebarkan elemen dari list ke dalam list2.

### Langkah 2

```
void main() {
  var list = [1, 2, 3];
  var list2 = [0, ...list];

  print(list); |
  print(list2);
  print(list2.length);
}
```

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

### Langkah 3

```
void main() {
  var list = [1, 2, 3];
  var list2 = [0, ...list];

  print(list);
  print(list2);
  print(list2.length);
  list1 = [1, 2, null];
  print(list1);
  var list3 = [0, ...?list1];
  print(list3.length);
}
```

3 issues

Undefined name 'list1'. line 8, col 3  
Try correcting the name to one that is defined, or defining the name.  
Undefined name 'list1'. line 9, col 7  
Try correcting the name to one that is defined, or defining the name.  
Undefined name 'list1'. line 10, col 21  
Try correcting the name to one that is defined, or defining the name.

- list1 dalam print(list1); dan list3, tetapi list1 tidak dideklarasikan sebelum digunakan. Selain itu, penugasan list1 seharusnya menggunakan var atau final untuk deklarasi variabel.
- Penggunaan ...?list1 untuk operator spread dengan null safety adalah benar, tetapi variabel list1 harus dideklarasikan dengan benar sebelumnya.

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

```
void main() {
  var list = [1, 2, 3];
  var listWithNIM = [2241720139];

  // Menggunakan Spread Operators untuk menggabungkan list dan listWithNIM
  var list2 = [0, ...list, ...listWithNIM];

  print('Original list: $list');
  print('List with NIM: $listWithNIM');
  print('Combined list (list2): $list2');
  print('Length of combined list: ${list2.length}');
}
```

Original list: [1, 2, 3]  
List with NIM: [2241720139]  
Combined list (list2): [0, 1, 2, 3, 2241720139]  
Length of combined list: 5

## Langkah 4

```
compileDDC
main.dart:13:49: Error: Undefined name 'promoActive'.
  var nav = ['Home', 'Furniture', 'Plants', if (promoActive) 'Outlet'];
                                              ^^^^^^^^^^^^^
```

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

```
void main() {
  var list = [1, 2, 3];
  var listWithNIM = [2241720139];

  // Menggunakan Spread Operators untuk menggabungkan list dan listWithNIM
  var list2 = [0, ...list, ...listWithNIM];

  print('Original list: $list');
  print('List with NIM: $listWithNIM');
  print('Combined list (list2): $list2');
  print('Length of combined list: ${list2.length}');

  bool promoActive = true;

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

Original list: [1, 2, 3]  
List with NIM: [2241720139]  
Combined list (list2): [0, 1, 2, 3, 2241720139]  
Length of combined list: 5  
[Home, Furniture, Plants, Outlet]

## Langkah 5

Tunjukkan hasilnya jika variabel login mempunyai kondisi lain.

```
void main() {
  var list = [1, 2, 3];
  var listWithNIM = [2241720139];

  // Menggunakan Spread Operators untuk menggabungkan list dan listWithNIM
  var list2 = [0, ...list, ...listWithNIM];

  print('Original list: $list');
  print('List with NIM: $listWithNIM');
  print('Combined list (list2): $list2');
  print('Length of combined list: ${list2.length}');

  bool promoActive = true;

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

  String login = 'Cust';

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

Original list: [1, 2, 3]  
List with NIM: [2241720139]  
Combined list (list2): [0, 1, 2, 3, 2241720139]  
Length of combined list: 5  
[Home, Furniture, Plants, Outlet]  
[Home, Furniture, Plants, Inventory]

## Langkah 6

```
void main() {
  var list = [1, 2, 3];
  var listWithNIM = [2241720139];

  // Menggunakan Spread Operators untuk menggabungkan list dan listWithNIM
  var list2 = [0, ...list, ...listWithNIM];

  print('Original list: $list');
  print('List with NIM: $listWithNIM');
  print('Combined list (list2): $list2');
  print('Length of combined list: ${list2.length}');

  bool promoActive = true;

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

  String login = 'Cust';

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

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

Original list: [1, 2, 3]  
List with NIM: [2241720139]  
Combined list (list2): [0, 1, 2, 3, 2241720139]  
Length of combined list: 5  
[Home, Furniture, Plants, Outlet]  
[Home, Furniture, Plants, Inventory]  
[#0, #1, #2, #3]

Tidak terjadi error

Collection For adalah fitur dalam Dart yang memungkinkan kamu untuk membuat list baru dengan menggunakan ekspresi for dalam list literal. Ini membuat kode lebih ringkas dan mudah dibaca.

## PRAKTIKUM 5

### Langkah 1

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

compileDDC  
main.dart:3:13: Error: Expected ',' after this.  
print(record)  
^

### Langkah 2

Terdapat kesalahan pada penulisan record

```
void main() {  
  var record = ['first', 2, true, 'last'];  
  print(record);  
}
```

[first, 2, true, last]

### Langkah 3

```
void main() {  
  var record = ['first', 2, true, 'last'];  
  print(record);  
  
  // Define a function to swap two elements  
  List<Object> tukar(List<Object> list, int i, int j) {  
    var temp = list[i];  
    list[i] = list[j];  
    list[j] = temp;  
    return list;  
  }  
  
  record = tukar(record, 0, 3);  
  print(record);  
}
```

[first, 2, true, last]  
[last, 2, true, first]

### Langkah 4

```
void main() {  
  var record = ['first', 2, true, 'last'];  
  print(record);  
  
  // Define a function to swap two elements  
  List<Object> tukar(List<Object> list, int i, int j) {  
    var temp = list[i];  
    list[i] = list[j];  
    list[j] = temp;  
    return list;  
  }  
  
  record = tukar(record, 0, 3);  
  print(record);  
  
  // Initialize mahasiswa with a value  
  (String, int) mahasiswa = ('Najwa Azzahra', 2241720139);  
  print(mahasiswa);  
}
```

[first, 2, true, last]  
[last, 2, true, first]  
(Najwa Azzahra, 2241720139)

### Langkah 5



```
1 void main() {
2   var record = ['first', 2, true, 'last'];
3   print(record);
4
5   // Define a function to swap two elements
6   List<Object> tukar(List<Object> list, int i, int j) {
7     var temp = list[i];
8     list[i] = list[j];
9     list[j] = temp;
10    return list;
11  }
12
13  record = tukar(record, 0, 3);
14  print(record);
15
16  // Initialize mahasiswa with a value
17  (String, int) mahasiswa = ('Najwa Azzahra', 2241720139);
18  print(mahasiswa);
19  var mahasiswa2 = ('Najwa Azzahra', a: 2, b: 2241720139, 'last');
20
21  print(mahasiswa2.$1);
22  print(mahasiswa2.a);
23  print(mahasiswa2.b);
24  print(mahasiswa2.$2);
25 }
```

[first, 2, true, last]  
[last, 2, true, first]  
(Najwa Azzahra, 2241720139)  
Najwa Azzahra  
2  
2241720139  
last

## 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!
  - Function adalah pernyataan yang mendefinisikan blok kode yang dapat digunakan untuk melakukan suatu tugas atau perhitungan tertentu. Fungsi adalah cara untuk mengemas kode yang melakukan operasi spesifik sehingga dapat dipanggil dan digunakan berulang kali tanpa menulis ulang kode tersebut.
3. Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!
  - Parameter Positional: Parameter positional adalah parameter yang harus diberikan dalam urutan tertentu saat memanggil fungsi. Ada dua jenis parameter positional:
    - Parameter Positional Wajib: Parameter ini harus diberikan ketika fungsi dipanggil.

```
void printFullName(String firstName, String lastName) {  
  print('Full name: $firstName $lastName');  
}
```

```
void main() {  
  printFullName(' Najwa ', 'Azzahra'); // Output: Full name: Najwa Azzahra  
}
```
    - Parameter Positional Opsional: Parameter ini tidak harus diberikan saat memanggil fungsi, dan diberikan dalam tanda kurung kotak []

```
void printInfo(String name, [int age]) {  
  if (age != null) {  
    print('Name: $name, Age: $age');  
  }
```

```

    } else {
        print('Name: $name');
    }
}

void main() {
    printInfo('Alice'); // Output: Name: Alice
    printInfo('Bob', 30); // Output: Name: Bob, Age: 30
}

```

- Parameter Nama: Parameter nama digunakan untuk menyebutkan parameter yang diberikan saat memanggil fungsi. Parameter nama ditandai dengan { } dalam definisi fungsi dan digunakan dengan nama saat memanggil fungsi.

- Parameter Nama Wajib: Harus menyertakan parameter nama ketika memanggil fungsi, dan itu dilakukan dengan menyebutkan nama parameter.

```

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

```

```

void main() {
    printDetails(name: 'Charlie', age: 25); // Output: Name: Charlie, Age: 25
}

```

- Parameter Nama Opsional: Parameter ini tidak harus diberikan, dan bisa diberi nilai default. Untuk parameter nama opsional, dapat menggunakan tanda kurung kurawal { } dan memberikan nilai default di dalam fungsi.

```

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

```

```

void main() {
    printGreeting(); // Output: Hello, Guest!
    printGreeting(greeting: 'Hi', name: 'Dave'); // Output: Hi, Dave!
}

```

- Parameter dengan Default Value: Parameter dengan nilai default memungkinkan untuk menentukan nilai default jika parameter tersebut tidak diberikan saat fungsi dipanggil. Nilai default ditentukan dalam definisi fungsi.

```

void printMessage(String message, [int repeatCount = 1]) {
    for (int i = 0; i < repeatCount; i++) {
        print(message);
    }
}

```

```

void main() {
    printMessage('Hello'); // Output: Hello
}

```

```
    printMessage('Hello', 3); // Output: Hello (ditampilkan 3 kali)
}
```

- Parameter Variadic: untuk menangani jumlah parameter yang tidak ditentukan dalam fungsi menggunakan parameter positional opsional berbentuk List.

```
void printNumbers(int first, int second, [int? third, int? fourth]) {
    print('Numbers: $first, $second, $third, $fourth');
}
```

```
void main() {
    printNumbers(1, 2); // Output: Numbers: 1, 2, null, null
    printNumbers(1, 2, 3, 4); // Output: Numbers: 1, 2, 3, 4
}
```

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

- Menyimpan Fungsi dalam Variabel: Fungsi dapat disimpan dalam variabel sehingga Anda dapat memanggilnya melalui variabel tersebut.

```
void sayHello() {
    print('Hello, world!');
}
```

```
void main() {
    var greet = sayHello; // Menyimpan referensi ke fungsi sayHello dalam variabel greet
    greet(); // Memanggil fungsi sayHello melalui variabel greet
}
```

- Meneruskan Fungsi sebagai Argumen: Fungsi dapat diteruskan sebagai argumen ke fungsi lain.

```
void executeFunction(Function func) {
    func(); // Memanggil fungsi yang diteruskan sebagai argumen
}
```

```
void sayHello() {
    print('Hello, world!');
}
```

```
void main() {
    executeFunction(sayHello); // Meneruskan fungsi sayHello ke executeFunction
}
```

- Mengembalikan Fungsi dari Fungsi Lain: Fungsi dapat dikembalikan sebagai nilai dari fungsi lain.

```
Function makeMultiplier(int multiplier) {
    return (int value) => value * multiplier; // Mengembalikan fungsi anonim
}
```

```
void main() {
```

```

    var multiplyByTwo = makeMultiplier(2); // Mendapatkan fungsi yang
mengalikan dengan 2
    print(multiplyByTwo(5)); // Output: 10
}

```

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

- Fungsi anonim (Anonymous Functions) adalah fungsi yang tidak memiliki nama. Fungsi ini biasanya didefinisikan langsung di tempat di mana fungsi tersebut akan digunakan, seperti saat memanggil fungsi lain atau menetapkan nilai variabel. Fungsi anonim sering digunakan dalam situasi di mana fungsi tersebut hanya diperlukan untuk tugas tertentu dan tidak akan digunakan di tempat lain

```

void performOperation(Function operation) {
    operation(); // Memanggil fungsi yang diteruskan sebagai argumen
}

```

```

void main() {
    performOperation() {
        print("This is an anonymous function!");
    });
}

```

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

- Lexical Scope adalah konsep di mana variabel yang diakses dalam fungsi ditentukan oleh lokasi fungsi itu didefinisikan dalam kode, bukan di mana fungsi itu dipanggil. Dalam konteks ini, scope atau lingkup variabel bersifat statis dan ditentukan pada waktu kompilasi berdasarkan struktur kode.

```

void outerFunction() {
    var outerVariable = 'I am outer';

    void innerFunction() {
        print(outerVariable); // Akses variabel outerVariable dari scope
outerFunction
    }
}

```

```

    innerFunction();
}

```

```

void main() {
    outerFunction(); // Output: I am outer
}

```

- Lexical Closures adalah konsep yang memungkinkan sebuah fungsi untuk "menutup" atau "menyimpan" akses ke variabel dari scope di mana fungsi tersebut didefinisikan, bahkan setelah fungsi tersebut selesai dieksekusi. Ini berarti fungsi dapat terus mengakses variabel dari lingkungan asalnya meskipun eksekusi telah pindah ke luar scope tersebut.

```

Function makeCounter() {
    int count = 0;

    // Fungsi anonim yang "menyimpan" akses ke variabel count
    return () {
        count++;
        return count;
    };
}

void main() {
    var counter = makeCounter(); // Mendapatkan fungsi closure
    print(counter()); // Output: 1
    print(counter()); // Output: 2
}

```

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

➤ Menggunakan Tuple (List)

```

List<dynamic> getPersonInfo() {
    String name = 'Najwa';
    int age = 20;
    bool isEmployed = true;

    // Mengembalikan beberapa nilai dalam bentuk List
    return [name, age, isEmployed];
}

void main() {
    var personInfo = getPersonInfo();

    print('Name: ${personInfo[0]}'); // Output: Name: Najwa
    print('Age: ${personInfo[1]}'); // Output: Age: 20
    print('Employed: ${personInfo[2]}'); // Output: Employed: true
}

```

➤ Menggunakan Map

```

Map<String, dynamic> getPersonDetails() {
    return {
        'name': 'Bob',
        'age': 25,
        'isStudent': false,
    };
}

void main() {
    var details = getPersonDetails();
}

```

```

print('Name: ${details['name']}'); // Output: Name: Bob
print('Age: ${details['age']}'); // Output: Age: 25
print('Student: ${details['isStudent']}'); // Output: Student: false
}

```

➤ Menggunakan Class

```

class Person {
    String name;
    int age;
    bool isEmployed;

    Person(this.name, this.age, this.isEmployed);
}

Person getPersonDetails() {
    return Person('Charlie', 28, true);
}

void main() {
    var person = getPersonDetails();

    print('Name: ${person.name}'); // Output: Name: Charlie
    print('Age: ${person.age}'); // Output: Age: 28
    print('Employed: ${person.isEmployed}'); // Output: Employed: true
}

```