

**JOBSHEET 3**  
**PEMROGRAMAN MOBILE**



**SERLI PUTRI MAHARANI**

**TI-3E / 26**

**2141720023**

**PROGRAM STUDI D – IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023**

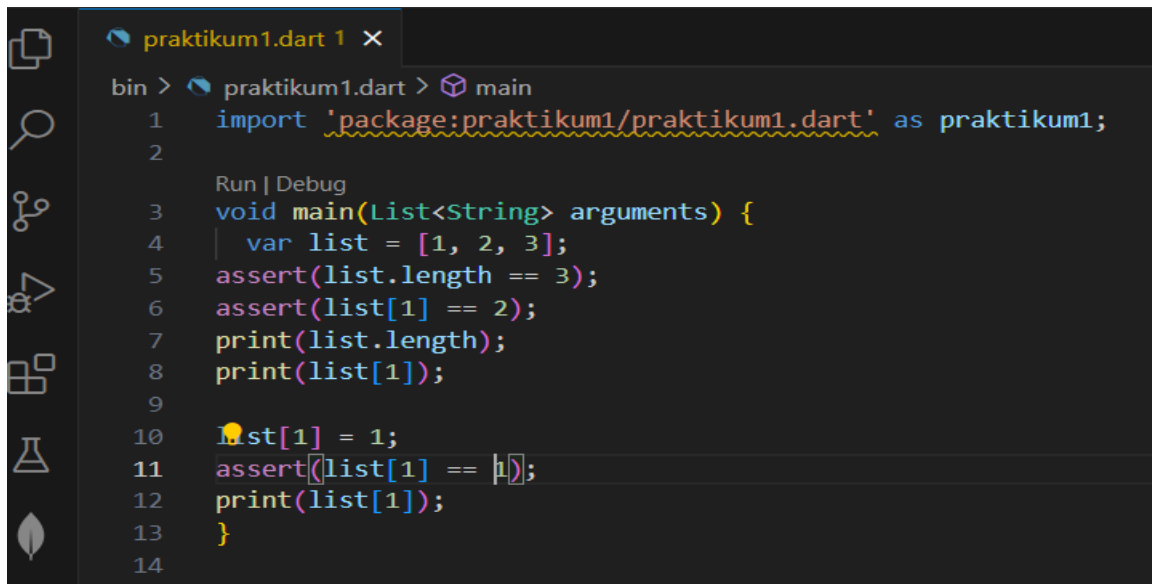
## PRAKTIKUM 1

### 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]);

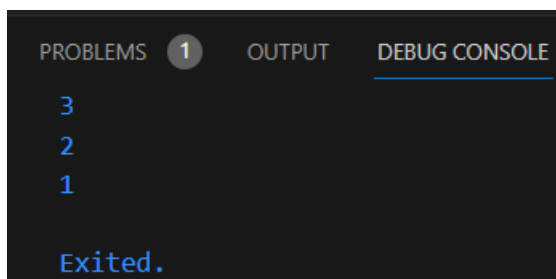
list[1] = 1;
assert(list[1] == 1);
print(list[1]);
```



```
bin > praktikum1.dart > main
1  import 'package:praktikum1/praktikum1.dart' as praktikum1;
2
3  Run | Debug
4  void main(List<String> arguments) {
5      | var list = [1, 2, 3];
6      | assert(list.length == 3);
7      | assert(list[1] == 2);
8      | print(list.length);
9      | print(list[1]);
10     |
11     | list[1] = 1;
12     | assert(list[1] == 1);
13     | print(list[1]);
14     }
```

### Langkah 2

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



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE
3
2
1
Exited.
```

### 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

```
14 //LANGKAH 3
15 final List<String?> list = List.filled(5, null);
16
17 // indeks ke 1 dengan nama
18 list[1] = 'Serli Putri Maharani';
19
20 // indeks ke 2 dengan NIM
21 list[2] = '2141720023';
22
23 print(list);
24 }
25
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

[null, Serli Putri Maharani, 2141720023, null, null]

Exited.

## PRAKTIKUM 2

### Langkah 1

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

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

```
bin > praktikum2.dart > main
1 import 'package:praktikum2/praktikum2.dart' as praktikum2;
2
3 void main(List<String> arguments) {
4   var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'}
5   print(halogens);
6 }
7
```

### Langkah 2

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

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
{fluorine, chlorine, bromine, iodine, astatine}
Exited.
```

### 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.

```
bin > praktikum2.dart > main
1 import 'package:praktikum2/praktikum2.dart' as praktikum2;
2
Run | Debug
3 void main(List<String> arguments) {}
4   var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'}
5   print(halogens);
6
7   var names1 = <String>{};
8   Set<String> names2 = {}; // This works, too.
9   var names3 = {}; // Creates a map, not a set.
10  print(names1);
11  print(names2);
12  print(names3);
13 }
```

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
{fluorine, chlorine, bromine, iodine, astatine}
{}
{}
{}
Exited.
```

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.

Dokumentasikan code dan hasil di console, lalu buat laporannya.

```
praktikum2.dart 1 X
in > praktikum2.dart > main
1 import 'package:praktikum2/praktikum2.dart' as praktikum2;
2
Run | Debug
3 void main(List<String> arguments) {
4   // var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine', 'astatine'};
5   // print(halogens);
6
7   var names1 = <String>{};
8   Set<String> names2 = {}; // This works, too.
9
10
11   // menggunakan .add()
12   names1.add('Serli Putri Maharani');
13   names1.add('2141720023');
14
15   // menggunakan .addAll()
16   names2.addAll({'Serli Putri', '2141720023'});
17
18   print(names1);
19   print(names2);
20 }
```

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
{Serli Putri Maharani, 2141720023}
{Serli Putri, 2141720023}

Exited.
```

## PRAKTIKUM 3

### 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);

```

```

bin > praktikum3.dart > main
1  import 'package:praktikum3/praktikum3.dart' as praktikum3;
2
Run | Debug
3  void main(List<String> arguments) {}
4      var gifts = {
5          // Key:    Value
6          'first': 'partridge',
7          'second': 'turtledoves',
8          'fifth': 1
9      };
10
11     var nobleGases = {
12         2: 'helium',
13         10: 'neon',
14         18: 2,
15     };
16
17     print(gifts);
18     print(nobleGases);
19 }

```

## Langkah 2

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

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
{first: partridge, second: turtledoves, fifth: 1}
{2: helium, 10: neon, 18: 2}

Exited.

```

### 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';
```



The screenshot shows an IDE window titled 'praktikum3.dart 3'. The code is as follows:

```
in > praktikum3.dart > main
Run | Debug
3 void main(List<String> arguments) {
4   var gifts = {
5     // Key:    Value
6     'first': 'partridge',
7     'second': 'turtledoves',
8     'fifth': 1
9   };
10
11   var nobleGases = {
12     2: 'helium',
13     10: 'neon',
14     18: 2,
15   };
16
17   print(gifts);
18   print(nobleGases);
19
20   var mhs1 = Map<String, String>();
21   gifts['first'] = 'partridge';
22   gifts['second'] = 'turtledoves';
23   gifts['fifth'] = 'golden rings';
24
25   var mhs2 = Map<int, String>();
26   nobleGases[2] = 'helium';
27   nobleGases[10] = 'neon';
28   nobleGases[18] = 'argon';
29 }
```

```
DEBUG CONSOLE  ...  Filter (e.g. text, !exclude)
{first: partridge, second: turtledoves, fifth: 1}
{2: helium, 10: neon, 18: 2}

Exited.
```

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!



```
praktikum3.dart 3 X
bin > praktikum3.dart > main
2
Run | Debug
3 void main(List<String> arguments) {
4   var gifts = {
5     // Key: Value
6     'nama': 'Serli Putri Maharani',
7     'NIM': '2141720023',
8     'first': 'partridge',
9     'second': 'turtledoves',
10    'fifth': 1
11  };
12
13  var nobleGases = [
14    0: 'Serli Putri M',
15    1: '2141720023',
16    2: 'helium',
17    10: 'neon',
18    18: 2,
19  ];
```

```
21 print(gifts);
22 print(nobleGases);
23
24 var mhs1 = Map<String, String>();
25 gifts['nama'] = 'Serli Putri Maharani';
26 gifts['NIM'] = '2141720023';
27 gifts['first'] = 'partridge';
28 gifts['second'] = 'turtledoves';
29 gifts['fifth'] = 'golden rings';
30
31 var mhs2 = Map<int, String>();
32 nobleGases[0] = 'Serli Putri M';
33 nobleGases[1] = '2141720023';
34 nobleGases[2] = 'helium';
35 nobleGases[10] = 'neon';
36 nobleGases[18] = 'argon';
37 }
38
```

```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS
{nama: Serli Putri Maharani, NIM: 2141720023, first: partridge, second: turtledoves, fifth: 1}
{0: Serli Putri M, 1: 2141720023, 2: helium, 10: neon, 18: 2}

Exited.
```

## PRAKTIKUM 4

### Langkah 1

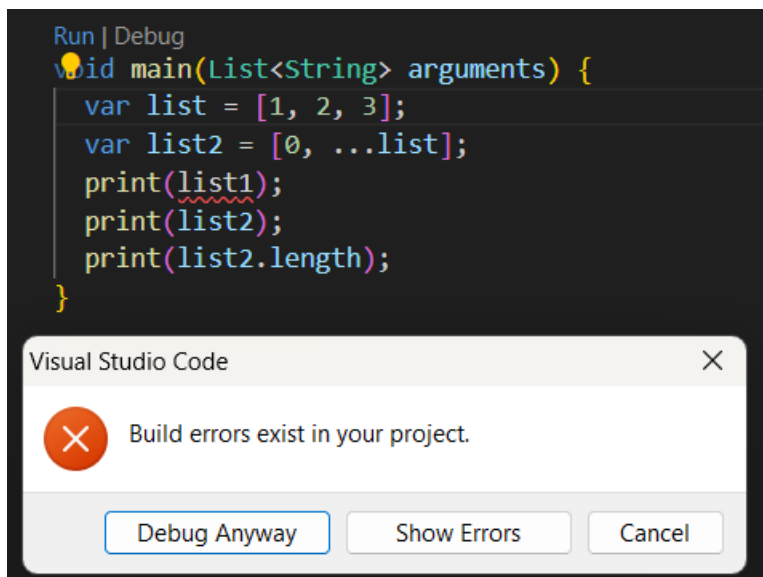
Ketik atau salin kode program berikut ke dalam fungsi `main()`.

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

```
bin > praktikum4.dart > main
1 import 'package:praktikum4/praktikum4.dart' as praktikum4;
2
Run | Debug
3 void main(List<String> arguments) {}
4   var list = [1, 2, 3];
5   var list2 = [0, ...list];
6   print(list1);
7   print(list2);
8   print(list2.length);
9 }
```

## Langkah 2

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



```
praktikum4.dart 1 X
bin > praktikum4.dart > main
1 import 'package:praktikum4/praktikum4.dart' as praktikum4;
2
Run | Debug
3 void main(List<String> arguments) {
4   var list1 = [1, 2, 3];
5   var list2 = [0, ...list1];
6   print(list1);
7   print(list2);
8   print(list2.length);
9 }
```

```
PROBLEMS 1 OUTPUT
[1, 2, 3]
[0, 1, 2, 3]
4
Exited.
```

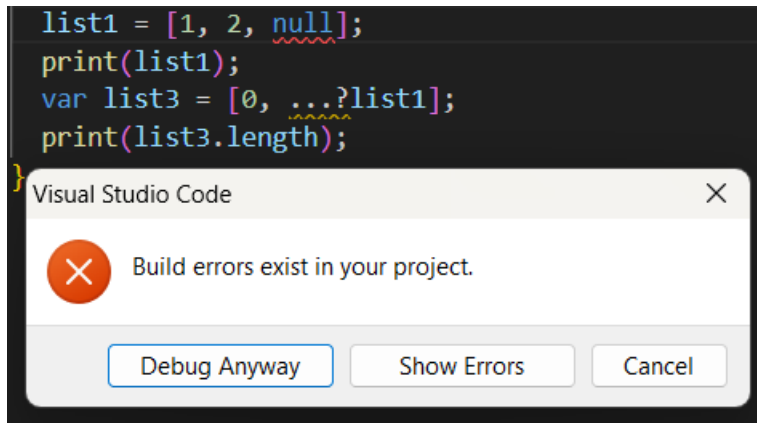
### Langkah 3

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

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

```
bin > praktikum4.dart > main
1 import 'package:praktikum4/praktikum4.dart' as praktikum4;
2
Run | Debug
3 void main(List<String> arguments) {
4   var list1 = [1, 2, 3];
5   var list2 = [0, ...list1];
6   print(list1);
7   print(list2);
8   print(list2.length);
9
10  list1 = [1, 2, null];
11  print(list1);
12  var list3 = [0, ...?list1];
13  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!

#### 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`.

#### 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.

#### 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

### Langkah 1

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

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

```
bin > praktikum5.dart > main
1  import 'package:praktikum5/praktikum5.dart' as praktikum5;
2
   Run | Debug
3  void main(List<String> arguments) {
4      var record = ('first', a: 2, b: true, 'last');
5      print(record);
6  }
```

### Langkah 2

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

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE
(first, last, a: 2, b: true)
Exited.
```

### 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.

```
bin > praktikum5.dart > main > tukar
1  import 'package:praktikum5/praktikum5.dart' as praktikum5;
2
Run | Debug
3  void main(List<String> arguments) {
4    // LANGKAH 1
5    var record = ('first', a: 2, b: true, 'last');
6    print(record);
7
8    // LANGKAH 3
9    (int, int) tukar((int, int) record) {
10   var (a, b) = record;
11   return (b, a);
12 }
13 }
```

DEBUG CONSOLE ... Filter (e.g. text, !exclude)

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

Exited.
```

#### 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!

```
14 // LANGKAH 4
15 // Record type annotation in a variable declaration:
16 (String, int) mahasiswa;
17 mahasiswa = ('Serli Putri Maharani', 2141720023);
18 print(mahasiswa);
19 }
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (

(first, last, a: 2, b: true)  
(Serli Putri Maharani, 2141720023)

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!

```
20 // LANGKAH 5
21 var mahasiswa2 = ('Serli Putri M', a: 2141720023, b: true, 'last');
22
23 print(mahasiswa2.$1); // Prints 'Serli Putri M'
24 print(mahasiswa2.a); // Prints 2141720023
25 print(mahasiswa2.b); // Prints true
26 print(mahasiswa2.$2); // Prints 'last'
27 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, !exclude)

Serli Putri M  
2141720023  
true  
last

Exited.

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

➔ Functions dalam Bahasa Dart adalah blok bangunan dari kode yang dapat dibaca, dipelihara, dan dapat digunakan Kembali. Fungsi adalah sekumpulan pernyataan untuk melakukan tugas tertentu dan mengatur program ke dalam blok kode logis. Setelah didefinisikan, fungsi dapat dipanggil untuk mengakses kode dan membuat kode dapat digunakan Kembali. Deklarasi fungsi berisi nama fungsi, tipe nilai kembalian fungsi, dan parameter-parameter yang dibutuhkan oleh fungsi tersebut.

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

➔ Parameter posisi: parameter yang diberikan berdasarkan urutan posisi saat memanggil fungsi.

Contoh:

```
void ucapan(String nama, String ucap) {  
    print('$ucap, $nama!');  
}
```

```
ucapan('Serli', 'Hello'); // Output: Hello, Serli!
```

➔ Parameter opsional: parameter yang tidak wajib diisi saat memanggil fungsi.

Contoh:

```
void ucapan(String nama, [String ucap]) {  
    if (ucap == null) {  
        ucap = 'Hello';  
    }  
    print('$ucap, $nama!');  
}
```

```
ucapan(Serli); // Output: Hello, Serli!
```

```
ucapan('Serli', 'Hi'); // Output: Hi, Serli!
```

➔ Parameter Bernama: parameter yang diberikan dengan menyebutkan namanya saat memanggil fungsi.

Contoh:

```
void ucapan({String nama, String ucapan}) {  
    if (ucapan == null) {  
        ucapan = 'Hello';  
    }  
}
```



```
    print('$ucapan, $nama!');  
}
```

```
ucap(nama: 'Serli'); // Output: Hello, Serli!
```

```
ucap(nama: 'Serli', ucapan: 'Hi'); // Output: Hi, Serli!
```

- ➔ Parameter nilai default: parameter yang memiliki nilai default jika tidak diisi saat memanggil fungsi.

Contoh:

```
void ucap(String nama, {String ucapan = 'Hello'}) {  
    print('$ucapan, $nama!');  
}
```

```
ucap('Serli'); // Output: Hello, Serli!
```

```
ucap('Serli', ucapan: 'Hi'); // Output: Hi, Serli!
```

- ➔ Parameter rest: parameter yang dapat menerima jumlah argument yang tidak terbatas saat memanggil fungsi.

Contoh:

```
void sum(int a, int b, [int c, int d, int e]) {  
    int result = a + b;  
    if (c != null) {  
        result += c;  
    }  
    if (d != null) {  
        result += d;  
    }  
    if (e != null) {  
        result += e;  
    }  
    print(result);  
}
```

```
sum(1, 2); // Output: 3
```

```
sum(1, 2, 3); // Output: 6
```

```
sum(1, 2, 3, 4); // Output: 10
```

```
sum(1, 2, 3, 4, 5); // Output: 15
```

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

- ➔ Functions sebagai first-class objects: fungsi yang diperlakukan seperti objek lainnya dalam Bahasa pemrograman. Dalam konsep ini, fungsi dapat disimpan dalam variabel, dijadikan parameter pada fungsi lain, dan dijadikan nilai kembalian dari fungsi lain. Konsep ini memungkinkan penggunaan fungsi secara fleksibel dan dapat meningkatkan efisiensi kode.

Contoh:

```
int add(int n1, int n2) {  
    return n1 + n2;  
}
```

```
// fungsi sebagai variabel
```

```
var calc1 = add;
```

```
// fungsi sebagai parameter
```

```
int calculate(int Function(int, int) operation, int n1, int n2) {  
    return operation(n1, n2);  
}
```

```
var result = calculate(add, 4, 2);
```

```
print(result); // Output: 6
```

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

- ➔ Anonymous functions atau lambda functions adalah fungsi yang tidak memiliki nama dan biasanya digunakan untuk melakukan operasi sederhana yang tidak memerlukan definisi fungsi yang lengkap. Fungsi ini dapat didefinisikan secara langsung di dalam kode atau sebagai parameter pada fungsi lain.

Contoh:

```
void main() {  
    var add = (int a, int b) => a + b;  
    var subtract = (int a, int b) => a - b;
```

```
var result1 = add(4, 2);
```

```
var result2 = subtract(4, 2);
```

```
print(result1); // Output: 6
```

```
print(result2); // Output: 2
```

}

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

➔ Lexical scope:

- Menentukan bagian program mana yang dapat menggunakan dan mengubah bagian lainnya
- Membuat sumber daya yang dideklarasikan (variabel, fungsi, dan objek) hanya tersedia untuk fungsi di mana mereka dideklarasikan untuk setiap fungsi anak yang bersarang di dalamnya
- Sumber daya yang dideklarasikan dalam fungsi anak yang bersarang biasanya tidak tersedia untuk induknya
- Lexical scope ditentukan pada saat menulis kode

Contoh:

```
void main() {  
  
    String outerVariable = "Saya di luar!";  
  
    void innerFunction() {  
  
        String innerVariable = "Saya di dalam!";  
  
        print(outerVariable); // mencetak "Saya di luar!"  
  
    }  
  
    // print(innerVariable); // ReferenceError: innerVariable is not defined  
  
    innerFunction();  
  
}
```

➔ Lexical closures:

- Adalah fungsi yang memiliki akses ke variabel dalam konteks di mana mereka di mana mereka didefinisikan, bukan di mana mereka di panggil
- Dibuat ketika sebuah fungsi di definisikan di dalam fungsi lain, dan fungsi dalam memiliki akses ke variabel luar fungsi dalam
- Memungkinkan untuk privasi data dan enkapsulasi dalam Dart
- Konsep yang lebih lanjut dan sering menjadi topik wawancara teknis

Contoh:

```
void outerFunction() {  
  
    String outerVariable = "Saya di luar!";  
  
    void innerFunction() {  
  
        String innerVariable = "Saya di dalam!";  
  
        print(outerVariable); // mencetak "Saya di luar!"  
  
    }  
  
    return innerFunction;  
  
}
```

```
void main() {  
  
    Function closure = outerFunction();  
  
    closure(); // mencetak "Saya di luar!"  
  
}
```

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

➔ Menggunakan list: dengan mengembalikan sebuah list yang berisi beberapa nilai

Contoh:

```
List<int> getMinMax(List<int> numbers) {  
    int min = numbers[0];  
    int max = numbers[0];  
    for (int i = 1; i < numbers.length; i++) {  
        if (numbers[i] < min) {  
            min = numbers[i];  
        }  
        if (numbers[i] > max) {  
            max = numbers[i];  
        }  
    }  
}
```

```

    }
}
return [min, max];
}

```

```

void main() {
    List<int> numbers = [5, 2, 8, 1, 9, 3];
    List<int> result = getMinMax(numbers);
    print('Minimum value: ${result[0]}');
    print('Maximum value: ${result[1]}');
}

```

➔ Menggunakan map: mengembangkan sebuah map yang berisi beberapa nilai

Contoh:

```

Map<String, dynamic> getPerson() {
    String name = 'John Doe';
    int age = 30;
    String address = '123 Main St';
    return {'name': name, 'age': age, 'address': address};
}

```

```

void main() {
    Map<String, dynamic> person = getPerson();
    print('Name: ${person['name']}');
    print('Age: ${person['age']}');
    print('Address: ${person['address']}');
}

```

➔ Menggunakan record: mengembalikan sebuah record yang berisi beberapa nilai

Contoh:

```

record Person(String name, int age, String address) {}

```

```

Person getPerson() {
    String name = 'John Doe';
    int age = 30;
    String address = '123 Main St';
    return Person(name, age, address);
}

```

```
}
```

```
void main() {  
    Person person = getPerson();  
    print('Name: ${person.name}');  
    print('Age: ${person.age}');  
    print('Address: ${person.address}');  
}
```

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