

Laporan UAS

Statistik Komputasi

NAMA : Ersa Oktavian Ramadan

NIM : 2241720208

Kelas : TI - 3C

ABSEN : 09

Praktikum 1: Eksperimen Tipe Data List

- **Langkah 1**

```
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]);  
}
```

Output :

```
● PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum1.dart  
3  
2  
1
```

- **Langkah 2**

Program diatas mendemonstrasikan cara mengakses panjang list, membaca elemen list, dan mengubah elemen list

- Langkah 3

```
void main() {  
    final List<String?> list = List.filled(5, null);  
  
    list[1] = "Ersa Oktavian Ramadan"; // Nama  
    list[2] = "2241720208"; // NIM  
  
    print("Isi list:");  
    for (int i = 0; i < list.length; i++) {  
        print("Index $i: ${list[i]}");  
    }  
}
```

Output :

```
PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum1.dart  
Isi list:  
Index 0: null  
Index 1: Ersa Oktavian Ramadan  
Index 2: 2241720208  
● Index 3: null  
Index 4: null
```

Isi list:

Index 0: null

Index 1: Ersa Oktavian Ramadan

Index 2: 2241720208

Index 3: null

Index 4: null

Praktikum 2: Eksperimen Tipe Data Set

- Langkah 1

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

Output :

```
PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum2.dart  
{fluorine, chlorine, bromine, iodine, astatine}  
PS D:\Belajar\Mobile Programming\Week 4> █
```

{fluorine, chlorine, bromine, iodine, astatine}

- Langkah 2

Pada sintaks `{}` digunakan untuk membuat Set secara literal lalu digunakan untuk menyimpan kumpulan nilai unik tanpa urutan tertentu.

- Langkah 3

```
void main() {  
    var halogens = {'fluorine', 'chlorine', 'bromine', 'iodine',  
                    'astatine'};  
    print(halogens);  
  
    var names1 = <String>{}; // Set kosong dengan tipe String  
    Set<String> names2 = {}; // Ini juga Set kosong dengan tipe  
String  
  
    // Menambahkan elemen nama dan NIM ke dalam Set  
    names1.add('Ersa Oktavian Ramadan'); // Menggunakan .add()  
    names1.add('2241720208'); // Menggunakan .add()
```

```

    names2.addAll({'Ersa Oktavian Ramadan', '2241720208'}); //
Menggunakan .addAll()

    print(names1); // Menampilkan names1
    print(names2); // Menampilkan names2
}

```

`names1.add()` digunakan untuk menambahkan elemen satu per satu.
`names2.addAll()` digunakan untuk menambahkan beberapa elemen sekaligus.

Output :

```

PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum2.dart
{fluorine, chlorine, bromine, iodine, astatine}
{Ersa Oktavian Ramadan, 2241720208}
{Ersa Oktavian Ramadan, 2241720208}

```

```

{fluorine, chlorine, bromine, iodine, astatine}
{Ersa Oktavian Ramadan, 2241720208}
{Ersa Oktavian Ramadan, 2241720208}

```

Praktikum 3: Eksperimen Tipe Data Maps

- 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);  
}
```

Output :

```
PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum3.dart  
{first: partridge, second: turtledoves, fifth: 1}  
{2: helium, 10: neon, 18: 2}
```

{first: partridge, second: turtledoves, fifth: 1}

{2: helium, 10: neon, 18: 2}

- Langkah 2

Penjelasan kode diatas adalah

Map **gifts**: Menyimpan data dengan key berupa **String** dan value berupa String atau integer, lalu untuk Map **nobleGases** menyimpan key berupa **Integer** dan value berupa String atau integer

- **Langkah 3**

Penjelasan :

Kedua map gifts dan nobleGases menyimpan data nama dan NIM di dalamnya

gifts: Map dengan key berupa **string** dan value campuran menyimpan hadiah, data nama dan , NIM.

nobleGases: Map dengan key **integer** yang menyimpan data unsur gas mulia, serta nama dan NIM.

mhs1: Map dengan key **string** menyimpan nama dan NIM.

mhs2: Map dengan key **integer** menyimpan nama dan NIM.

```
void main() {  
    var gifts = {  
        'first': 'partridge',  
        'second': 'turtledoves',  
        'fifth': 'golden rings',  
        'name': 'Ersa Oktavian Ramadan',  
        'nim': '2241720208',  
    };  
  
    var nobleGases = {  
        2: 'helium',  
        10: 'neon',  
        18: 'argon',  
        99: 'Ersa Oktavian Ramadan',  
        100: '2241720208',  
    };  
  
    var mhs1 = {'name': 'Ersa Oktavian Ramadan', 'nim':  
'2241720208'};  
    var mhs2 = {1: 'Ersa Oktavian Ramadan', 2: '2241720208'};  
  
    print(gifts);  
    print(nobleGases);  
    print(mhs1);  
    print(mhs2);  
}
```

Output :

```
PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum3.dart
{first: partridge, second: turtledoves, fifth: golden rings, name: Ersa Oktavian Ramadan, nim: 2241720208}
{2: helium, 10: neon, 18: argon, 99: Ersa Oktavian Ramadan, 100: 2241720208}
{name: Ersa Oktavian Ramadan, nim: 2241720208}
{1: Ersa Oktavian Ramadan, 2: 2241720208}
```

{first: partridge, second: turtledoves, fifth: golden rings, name: Ersa Oktavian Ramadan, nim: 2241720208}

{2: helium, 10: neon, 18: argon, 99: Ersa Oktavian Ramadan, 100: 2241720208}

{name: Ersa Oktavian Ramadan, nim: 2241720208}

{1: Ersa Oktavian Ramadan, 2: 2241720208}

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

- Langkah 1

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

Output :

Praktikum4.dart:4:7: Error: Undefined name 'list1'.
print(list1);
 ^^^^^

- Langkah 2

Penjelasan :

`list1` tidak didefinisikan dalam awal kode maka error, lalu diganti menjadi `print(list)` untuk mencetak variabel `list` yang sudah didefinisikan.

Perbaikan :

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

Output :

```
• PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum4.dart
[1, 2, 3]
[0, 1, 2, 3]
4
```

[1, 2, 3]

[0, 1, 2, 3]

4

- **Langkah 3**

```
void main() {
    var list = [1, 2, 3];
    var list1 = [1, 2, null];
    var nim = ['2241720208'];

    print(list);
    print([0, ...list]);
    print([0, ...list].length);

    print(list1);
    print([0, ...?list1]);
    print([0, ...?list1].length);

    print([0, ...list, ...nim]);
    print([0, ...list, ...nim].length);
}
```

Output :

```

PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum4.dart
[1, 2, 3]
[0, 1, 2, 3]
⊗ 4
[1, 2, null]
[0, 1, 2, null]
4
[0, 1, 2, 3, 2241720208]
5

```

- Langkah 4

- promoActive = true

```

void main() {
  var list = [1, 2, 3];
  var list1 = [1, 2, null];
  var nim = ['2241720208'];

  print(list);
  print([0, ...list]);
  print([0, ...list].length);

  print(list1);
  print([0, ...?list1]);
  print([0, ...?list1].length);

  print([0, ...list, ...nim]);
  print([0, ...list, ...nim].length);

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

```

- promoActive = false

```

void main() {
  var list = [1, 2, 3];
  var list1 = [1, 2, null];

```

```
var nim = ['2241720208'];

print(list);
print([0, ...list]);
print([0, ...list].length);

print(list1);
print([0, ...?list1]);
print([0, ...?list1].length);

print([0, ...list, ...nim]);
print([0, ...list, ...nim].length);

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

Output :

- TRUE

```

PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum4.dart
[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2, null]
[0, 1, 2, null]
4
[0, 1, 2, 3, 2241720208]
5
[Home, Furniture, Plants, Outlet]

```

- FALSE

```

PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum4.dart
[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2, null]
[0, 1, 2, null]
4
[0, 1, 2, 3, 2241720208]
5
[Home, Furniture, Plants]

```

```

[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2, null]
[0, 1, 2, null]
4
[0, 1, 2, 3, 2241720208]
5
[Home, Furniture, Plants, Outlet]

```

Penjelasan :

- Mengganti var promoActive menjadi **true** untuk mendapatkan 'Outlet' pada list nav
Maka hasilnya nav promoActive adalah ['Home', 'Furniture', 'Plants', 'Outlet']
- Mengganti var promoActive menjadi **false** untuk mendapatkan 'Outlet' pada list nav
Maka hasilnya nav promoActive adalah ['Home', 'Furniture', 'Plants']

• Langkah 5

- Menggunakan login 'Manager'

```

void main() {
    var list = [1, 2, 3];
    var list1 = [1, 2, null];
    var nim = ['2241720208'];

    print(list);
    print([0, ...list]);
    print([0, ...list].length);

    print(list1);
    print([0, ...?list1]);
    print([0, ...?list1].length);

    print([0, ...list, ...nim]);
    print([0, ...list, ...nim].length);

    // Tambahan kode
    var login = 'Manager';
    var nav2 = ['Home', 'Furniture', 'Plants', if (login ==
'Manager') 'Inventory'];
    print(nav2);
}

```

- Disini saya menggunakan kondisi lain yaitu 'CEO'

```

void main() {

```

```
var list = [1, 2, 3];
var list1 = [1, 2, null];
var nim = ['2241720208'];

print(list);
print([0, ...list]);
print([0, ...list].length);

print(list1);
print([0, ...?list1]);
print([0, ...?list1].length);

print([0, ...list, ...nim]);
print([0, ...list, ...nim].length);

// Tambahan kode
var login = 'CEO';
var nav2 = ['Home', 'Furniture', 'Plants', if (login ==
'Manager') 'Inventory'];
print(nav2);
}
```

Output :

- Login 'Manager'

```

PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum4.dart
[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2, null]
[0, 1, 2, null]
4
[0, 1, 2, 3, 2241720208]
5
[Home, Furniture, Plants, Inventory]

```

[1, 2, 3]

[0, 1, 2, 3]

4

[1, 2, null]

[0, 1, 2, null]

4

[0, 1, 2, 3, 2241720208]

5

[Home, Furniture, Plants, Inventory]

- Login 'CEO'

```

PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum4.dart
[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2, null]
[0, 1, 2, null]
4
[0, 1, 2, 3, 2241720208]
5
[Home, Furniture, Plants]

```

[1, 2, 3]

[0, 1, 2, 3]

4

[1, 2, null]

[0, 1, 2, null]

4

[0, 1, 2, 3, 2241720208]

5

[Home, Furniture, Plants]

Penjelasan :

Jika `login` adalah `'Manager'`, `'Inventory'` akan ditambahkan ke list `nav2`.

Jika `login` nilai lain, `'Inventory'` tidak akan ditambahkan.

Pada case diatas saya menggunakan nilai lain yaitu `'CEO'` maka `'Inventory'` tidak ditambahkan pada `nav2`

- **Langkah 6**

```
void main() {  
    var list = [1, 2, 3];  
    var list1 = [1, 2, null];  
    var nim = ['2241720208'];  
  
    print(list);  
    print([0, ...list]);  
    print([0, ...list].length);  
  
    print(list1);  
    print([0, ...?list1]);  
    print([0, ...?list1].length);  
  
    print([0, ...list, ...nim]);  
    print([0, ...list, ...nim].length);  
  
    // Tambahan kode  
    var login = 'Manager';  
    var nav2 = ['Home', 'Furniture', 'Plants', if (login ==  
'Manager') 'Inventory'];  
    print(nav2);  
  
    // Kode tambahan  
    var listOfInts = [1, 2, 3];  
    var listOfStrings = ['#0', for (var i in listOfInts) '#$i'];  
    assert(listOfStrings[1] == '#1');  
    print(listOfStrings);  
}
```

Output :

```
PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum4.dart
[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2, null]
[0, 1, 2, null]
4
[0, 1, 2, 3, 2241720208]
5
[Home, Furniture, Plants, Inventory]
[#0, #1, #2, #3]
```

```
[1, 2, 3]
[0, 1, 2, 3]
4
[1, 2, null]
[0, 1, 2, null]
4
[0, 1, 2, 3, 2241720208]
5
[Home, Furniture, Plants, Inventory]
[#0, #1, #2, #3]
```

Penjelasan :

`for (var i in listOfInts) '#$i'` digunakan untuk menghasilkan elemen baru berdasarkan iterasi, memudahkan pula pembuatan list baru dari list yang ada dengan transformasi di setiap elemen.

Praktikum 5: Eksperimen Tipe Data Records

- Langkah 1

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

Output :

```
PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum5.dart  
(first, last, a: 2, b: true)
```

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

- Langkah 2

Error pada akhir print(record) yang kekurangan dengan (;) titik koma

- Langkah 3

```
void main() {  
    var record = ('first', a: 2, b: 3, 'last');  
    print('Original record: $record');  
  
    var swapped = tukar((record.a ?? 0, record.b ?? 0));  
    print('Swapped values: $swapped');  
}  
  
(int, int) tukar((int, int) record) {  
    var (a, b) = record;  
    return (b, a);  
}
```

Output :

```
PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum5.dart
Original record: (first, last, a: 2, b: 3)
Swapped values: (3, 2)
```

Original record: (first, last, a: 2, b: 3)

Swapped values: (3, 2)

Penjelasan ;

record: Menampilkan tuple dengan format (String, int, int, String).

tukar(): Menukar nilai dari tuple (2, 3) menjadi (3, 2).

- **Langkah 4**

```
void main() {
    var record = ('first', a: 2, b: 3, 'last');
    print('Original record: $record');

    var swapped = tukar((record.a ?? 0, record.b ?? 0));
    print('Swapped values: $swapped');

    (String, int) mahasiswa = ('Ersa Oktavian Ramadan', 2241720208);
    print('Mahasiswa record: $mahasiswa');
}

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

Output :

```
PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum5.dart
Original record: (first, last, a: 2, b: 3)
Swapped values: (3, 2)
Mahasiswa record: (Ersa Oktavian Ramadan, 2241720208)
```

Original record: (first, last, a: 2, b: 3)

Swapped values: (3, 2)

Mahasiswa record: (Ersa Oktavian Ramadan, 2241720208)

Penjelasan :

Penggunaan tuple untuk menyimpan data yang terstruktur. Variabel `record` dan `mahasiswa` masing-masing memiliki tipe dan nilai yang berbeda, dengan `mahasiswa` diinisialisasi untuk menyimpan nama dan NIM dengan tipe `(String, int)`.

Maka hasil yang tersimpan adalah **Mahasiswa record: (Ersa Oktavian Ramadan, 2241720208)**

- Langkah 5

```
void main() {  
    var record = ('first', a: 2, b: 3, 'last');  
    print('Original record: $record');  
  
    var swapped = tukar((record.a ?? 0, record.b ?? 0));  
    print('Swapped values: $swapped');  
  
    (String, int) mahasiswa = ('Ersa Oktavian Ramadan', 2241720208);  
    print('Mahasiswa record: $mahasiswa');  
  
    var mahasiswa2 = ('first', a: 2, b: true, 'last');  
  
    print(mahasiswa2.$1);  
    print(mahasiswa2.a);  
    print(mahasiswa2.b);  
    print(mahasiswa2.$2);  
  
    var mahasiswa3 = ('Ersa Oktavian Ramadan', a: 2241720208, b:  
true, 'last');  
    print(mahasiswa3.$1);  
    print(mahasiswa3.a);  
    print(mahasiswa3.b);  
    print(mahasiswa3.$2);  
}  
  
(int, int) tukar((int, int) record) {  
    var (a, b) = record;  
    return (b, a);  
}
```

Output :

```
● PS D:\Belajar\Mobile Programming\Week 4> dart run Praktikum5.dart
⊗ Original record: (first, last, a: 2, b: 3)
  Swapped values: (3, 2)
  Mahasiswa record: (Ersa Oktavian Ramadan, 2241720208)
  first
  2
  true
  last
  Ersa Oktavian Ramadan
  2241720208
  true
  last
```

Swapped values: (3, 2)

Mahasiswa record: (Ersa Oktavian Ramadan, 2241720208)

first

2

true

last

Ersa Oktavian Ramadan

2241720208

true

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!
3. Jelaskan jenis-jenis parameter di Functions beserta contoh sintaksnya!
4. Jelaskan maksud Functions sebagai first-class objects beserta contoh sintaknya!
5. Apa itu Anonymous Functions? Jelaskan dan berikan contohnya!
6. Jelaskan perbedaan Lexical scope dan Lexical closures! Berikan contohnya!
7. Jelaskan dengan contoh cara membuat return multiple value di Functions!
8. Kumpulkan berupa link commit repo GitHub pada tautan yang telah disediakan di grup Telegram!

1. Sudah saya lampirkan diatas
2. Functions dalam Dart adalah cara untuk mengelompokkan serangkaian instruksi menjadi satu unit yang bisa digunakan berulang kali, membuat kode Anda lebih efisien dan mudah dikelola.
3. - **Parameter Wajib (Required Parameters)** : Ini seperti bahan utama dalam resep yang harus ada

Contoh sintaks :

```
void masakNasi(String jenisNasi, int jumlahGelas) {  
    // kode untuk memasak nasi  
}
```

- **Parameter Opsional (Optional Parameters)** : Ada dua jenis parameter opsional:

a. Parameter Opsional Posisi (Optional Positional Parameters) Ini seperti bumbu tambahan yang bisa ditambahkan atau tidak.

Contoh sintaks :

```
void buatKopi(String jenisKopi, [String? susu, String? gula]) {  
    // kode untuk membuat kopi  
}
```

b. Parameter Opsional Bernama (Optional Named Parameters) Ini seperti menu tambahan yang bisa kita pilih saat memesan makanan.

Contoh sintaks :

```
void pesanPizza({String topping = 'keju', String? saus, required int ukuran}) {  
    // kode untuk memesan pizza  
}
```

- Parameter Default (Default Parameters) : Ini seperti resep standar yang bisa diubah jika kita mau

Contoh sintaks :

```
void buatTeh((String jenisTeh = 'hitam', int gula = 1)) {  
    // kode untuk membuat teh  
}
```

4. - Menyimpan function dalam variabel:

```
int tambah(int a, int b) {  
    return a + b;  
}
```

```
void main() {  
    var operasiMatematika = tambah;  
    print(operasiMatematika(5, 3)); // Output: 8  
}
```

- Melewatkan function sebagai argumen:

```
void lakukanOperasi(int a, int b, Function operasi) {  
    print(operasi(a, b));  
}
```

```
int kali(int x, int y) {  
    return x * y;  
}
```

```
void main() {  
    lakukanOperasi(4, 5, kali); // Output: 20  
}
```

- Mengembalikan function dari function lain:

```
Function buatPengganda(int pengali) {  
    return (int angka) => angka * pengali;  
}
```

```
void main() {  
    var kaliDua = buatPengganda(2);  
    print(kaliDua(5)); // Output: 10
```

```
    var kaliTiga = buatPengganda(3);  
    print(kaliTiga(5)); // Output: 15  
}
```


5. Anonymous Functions, atau bisa kita sebut "fungsi tanpa nama", adalah fungsi yang kita buat tanpa memberinya nama khusus, contoh penerapan

- Penggunaan dasar:

```
void main() {  
    var tambahLima = (int angka) {  
        return angka + 5;  
    };  
  
    print(tambahLima(10)); // Output: 15  
}
```

- Penggunaan dengan list :

```
void main() {  
    var buah = ['apel', 'pisang', 'ceri'];  
  
    buah.forEach((item) {  
        print('Saya suka makan ${item}');  
    });  
}
```

- Penggunaan singkat (arrow function):

```
void main() {  
    var angka = [1, 2, 3, 4, 5];  
  
    var hasilKaliDua = angka.map((n) => n * 2);  
  
    print(hasilKaliDua); // Output: (2, 4, 6, 8, 10)  
}
```

6. **Lexical scope** adalah tentang aturan akses variabel berdasarkan struktur kode.

Contoh :

```
var global = 'Saya global';  
  
void fungsiLuar() {  
    var luar = 'Saya di luar';  
  
    void fungsiBersarang() {  
        var dalam = 'Saya di dalam';  
        print(global); // Bisa akses  
        print(luar); // Bisa akses  
        print(dalam); // Bisa akses  
    }  
  
    fungsiBersarang();  
    print(global); // Bisa akses  
    print(luar); // Bisa akses
```

```

    // print(dalam); // Tidak bisa akses, akan error
}

void main() {
    fungsiLuar();
    print(global); // Bisa akses
    // print(luar); // Tidak bisa akses, akan error
    // print(dalam); // Tidak bisa akses, akan error
}

```

Lexical closures adalah kemampuan fungsi untuk "mengingat" dan mengakses variabel dari lingkup luarnya, bahkan setelah fungsi luar selesai.

Contoh :

```

Function buatPenghitung() {
    var hitungan = 0;

    return () {
        hitungan++;
        print('Hitungan: $hitungan');
    };
}

void main() {
    var penghitung = buatPenghitung();
    penghitung(); // Output: Hitungan: 1
    penghitung(); // Output: Hitungan: 2
    penghitung(); // Output: Hitungan: 3
}

```

7. - Menggunakan List

```

List<dynamic> hitungStatistik(List<int> angka) {
    int jumlah = angka.reduce((a, b) => a + b);
    double rataRata = jumlah / angka.length;
    int terkecil = angka.reduce((a, b) => a < b ? a : b);
    int terbesar = angka.reduce((a, b) => a > b ? a : b);

    return [jumlah, rataRata, terkecil, terbesar];
}

void main() {
    var hasil = hitungStatistik([1, 2, 3, 4, 5]);
    print('Jumlah: ${hasil[0]}');
    print('Rata-rata: ${hasil[1]}');
    print('Terkecil: ${hasil[2]}');
    print('Terbesar: ${hasil[3]}');
}

```

- Menggunakan Map

```
Map<String, dynamic> hitungStatistik(List<int> angka) {  
    int jumlah = angka.reduce((a, b) => a + b);  
    double rataRata = jumlah / angka.length;  
    int terkecil = angka.reduce((a, b) => a < b ? a : b);  
    int terbesar = angka.reduce((a, b) => a > b ? a : b);  
  
    return {  
        'jumlah': jumlah,  
        'rataRata': rataRata,  
        'terkecil': terkecil,  
        'terbesar': terbesar  
    };  
}  
  
void main() {  
    var hasil = hitungStatistik([1, 2, 3, 4, 5]);  
    print('Jumlah: ${hasil['jumlah']}');  
    print('Rata-rata: ${hasil['rataRata']}');  
    print('Terkecil: ${hasil['terkecil']}');  
    print('Terbesar: ${hasil['terbesar']}');  
}
```

- Menggunakan Record (Dart 3.0+)

```
(int jumlah, double rataRata, int terkecil, int terbesar) hitungStatistik(List<int> angka) {  
    int jumlah = angka.reduce((a, b) => a + b);  
    double rataRata = jumlah / angka.length;  
    int terkecil = angka.reduce((a, b) => a < b ? a : b);  
    int terbesar = angka.reduce((a, b) => a > b ? a : b);  
  
    return (jumlah, rataRata, terkecil, terbesar);  
}  
  
void main() {  
    var hasil = hitungStatistik([1, 2, 3, 4, 5]);  
    print('Jumlah: ${hasil.$1}');  
    print('Rata-rata: ${hasil.$2}');  
    print('Terkecil: ${hasil.$3}');  
    print('Terbesar: ${hasil.$4}');  
}
```

8. Github: https://github.com/Ersaoktaviannn/MobileProgramming_2024-2025.git

—