

**TUGAS PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK**

MODUL III

03_Pengenalan_Dart



Disusun Oleh :

Aditya Prabu Mukti / 2211104037

SE06B

Asisten Praktikum :

Muhammad Faza Zulian Gesit Al Barru

Aisyah Hasna Aulia

Dosen Pengampu :

Yudha Islami Sulistya, S.Kom., M.Cs.

**PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA**

TUGAS GUIDED

A. penamaan variable menggunakan camelCase

- **CamelCase** adalah gaya penulisan dalam pemrograman yang digunakan untuk menulis nama variabel, fungsi, atau identifier lainnya dengan cara menggabungkan beberapa kata tanpa spasi.

Source code

```
void main() {  
    //penamaan variable menggunakan camelCase  
  
    String firstName = 'praktum';  
    var lastName = 'PPB';  
    print('hari ini $firstName $lastName');
```

Output:

```
Connecting to VM Service at ws://127.0.0.1:58435/5Pz3FyZueXA=/ws  
Connected to the VM Service.  
hari ini praktikum PPB  
Exited.
```

B. Statement Control

- **IF-ELSE Statement** adalah salah satu struktur kontrol dalam pemrograman yang digunakan untuk membuat keputusan berdasarkan kondisi tertentu.

Source Code:

```
var open = 8;
var close = 17;
var now = 12;

if (now >= open && now < close) {
| print("Toko buka");
} else if (now == 12) {
| print("Tokko Sedang Istirahhat");
} else {
| print("toko tutup");
}
```

Output:

```
Connecting to VM Service at ws://127.0.0.1:58710/axKeCSWRoMo=/ws
Connected to the VM Service.
Toko buka

Exited.
```

- **Switch-Case Statement**

Switch-case digunakan untuk memilih salah satu dari banyak blok kode untuk dieksekusi berdasarkan nilai suatu ekspresi. Dengan switch-case, program dapat memeriksa variabel dan menjalankan kode yang sesuai dengan nilai yang diberikan. Jika tidak ada case yang cocok, blok default dapat digunakan untuk menangani nilai yang tidak terduga.

Source Code:

```
//Switch Case
var nilai = 'b';

switch (nilai) {
  case 'a':
    print('Nilai Sangat Bagus');
    break;
  case 'b':
    print('Nilai Bagus');
    break;
  case 'c':
    print('Nilai Cukup');
    break;
  default:
    print('Nilai Tidak Tersedia');
}
```

Output:

```
Connecting to VM Service at ws://127.0.0.1:59227/59Q_wyOssTA=/ws
Connected to the VM Service.
Nilai Bagus

Exited.
```

C. Looping terdapat dua cara untuk melakukan *looping* di dart, yaitu menggunakan *for loop* dan *while loop*.

- **For Loop**

For loop digunakan untuk mengulangi suatu blok kode dengan jumlah perulangan yang sudah ditentukan.

Source code

```
// For loop untuk mencetak angka 1 sampai 5
for (int i = 1; i <= 5; i++) {
    print(i);
}
```

Output:

```
Connecting to VM Service at ws://127.0.0.1:59331/kXtIxDnGihI=/ws
Connected to the VM Service.
1
2
3
4
5

Exited.
```

- **While Loops**

While loop digunakan ketika jumlah iterasi belum pasti, dan perulangan akan berlangsung selama kondisi yang diberikan bernilai true.

Source Code

```
// While loop untuk mencetak angka 1 sampai 5

// Inisialisasi variabel
int i = 1;

while (i <= 5) {
    print('Angka: $i');
    i++; // Increment untuk menghindari loop tak berujung
}
```

Output:

```
Connecting to VM Service at ws://127.0.0.1:59362/brLM3XrG9SQ=/ws
Connected to the VM Service.
Angka: 1
Angka: 2
Angka: 3
Angka: 4
Angka: 5

Exited.
```

D. List

List adalah tipe data yang digunakan untuk merepresentasikan sekumpulan nilai yang terurut. Elemen-elemen di dalamnya bisa memiliki tipe data yang sama atau berbeda. List bisa diinisialisasi dengan elemen-elemen yang sudah ada, atau bisa dibuat kosong dan diisi belakangan.

- **Fixed Length List**

- Tidak dapat diubah ukurannya setelah dibuat.
- Elemen-elemen dalam list masih bisa dimodifikasi, tetapi kita tidak bisa menambahkan atau menghapus elemen karena ukurannya sudah ditentukan sejak awal pembuatan..

Source Code

```
// Membuat fixed-length list dengan panjang 3
List<int> fixedList =
| | List.filled(3, 0); // List dengan 3 elemen, semua diinisialisasi ke 0
|
// Mengubah elemen dalam list
fixedList[0] = 10;
fixedList[1] = 20;
fixedList[2] = 30;
print('Fixed Length List: $fixedList');

// Menambah atau menghapus elemen tidak diperbolehkan
// fixedList.add(30); // Ini akan menimbulkan error
// fixedList.removeAt(0); // Ini juga akan menimbulkan error
```

Output:

```
Connecting to VM Service at ws://127.0.0.1:59438/JhwzCFBqovw=/ws
Connected to the VM Service.
Fixed Length List: [10, 20, 30]

Exited.
```

- **Growable List**

Gunakan growable list apabila memiliki banyak object yang tidak menentu atau banyaknya object yang terus bertambah.

Source Code:

```
// Membuat growable list (panjangnya bisa berubah)
List<int> growableList = [];
// Menambahkan elemen baru ke dalam list
growableList.add(10);
growableList.add(20);
growableList.add(30);
print('Growable List setelah menambah elemen: $growableList');
// Menambahkan lebih banyak elemen
growableList.add(40);
growableList.add(50);
print(growableList);
// Menghapus elemen dari list
growableList.remove(20);
print('Growable List setelah menghapus elemen: $growableList');
```

Output:

```
Connecting to VM Service at ws://127.0.0.1:59478/TMI4rdCr2QI=/ws
Connected to the VM Service.
Growable List setelah menambah elemen: [10, 20, 30]
[10, 20, 30, 40, 50]
Growable List setelah menghapus elemen: [10, 30, 40, 50]

Exited.
```

Note: berbeda dari fixed length list, growable list dapat diubah ukurannya secara dinamis, kita bisa menambah atau menghapus elemen kapan saja.

E. Fungsi

Fungsi adalah blok kode yang dirancang untuk menjalankan tugas tertentu dan, jika diperlukan, dapat mengembalikan hasil. Dalam Dart, fungsi juga bisa menerima parameter atau argumen sebagai masukan. Di bahasa pemrograman yang mendukung Object Oriented Programming (OOP), fungsi atau prosedur memiliki peran yang sangat penting. Untuk menciptakan kode yang berkualitas, programmer dapat menerapkan beberapa prinsip pemrograman seperti SOLID, KISS, YAGNI, dan lainnya. Prinsip-prinsip ini menekankan pada separation of concern, yang berarti setiap bagian kode memiliki tanggung jawab spesifik dan berupaya meminimalisir boilerplate code sebanyak mungkin.

Source Code:


```
//Memanggil Fungsi

void cetakPesan(String pesan) {
    | print(pesan);
    | }

    int perkalian(int a, int b) {
    | return a * b;
    | }

Run | Debug
void main() {
    | int hasil = perkalian(6, 12);
    | print('Hasil perkalian fungsi: $hasil');
    |
    | cetakPesan('Halo Deck');
    | }
}
```

Output:

```
Connecting to VM Service at ws://127.0.0.1:59567/sgXC8ZP5muk=/ws
Connected to the VM Service.
Hasil perkalian fungsi: 72
Halo Deck

Exited.
```

TUGAS UNGUIDED

1. Tugas Percabangan (Branching)

Source code

```
import 'dart:io';

String cekNilai(int nilai) {
  if (nilai > 70) {
    return "Nilai A";
  } else if (nilai > 40 && nilai <= 70) {
    return "Nilai B";
  } else if (nilai > 0 && nilai <= 40) {
    return "Nilai C";
  } else {
    return "";
  }
}

void main() {

  print("Masukkan nilai: ");
  int? inputNilai = int.parse(stdin.readLineSync());

  String hasil = cekNilai(inputNilai);

  if (hasil.isNotEmpty) {
    print("$inputNilai merupakan $hasil");
  } else {
    print("Nilai tidak valid");
  }
}
```

OUTPUT:

```
PS D:\SEMESTER 5\PROJECT PPB\PPB_ADITYA_PRABU_MUKTI_2211104037\03_Pengenalan_Dart\UNGUIDED> dart branching.dart
Masukkan nilai:
80
80 merupakan Nilai A
```

PENJELASAN:

Fungsi ini menggunakan percabangan if-else if-else untuk memeriksa apakah nilai tersebut termasuk dalam kategori "Nilai A", "Nilai B", atau "Nilai C" berdasarkan rentang nilainya.

Jika nilai di atas 70, maka akan dikategorikan sebagai "Nilai A"; jika nilai di antara 41 hingga 70, maka dikategorikan sebagai "Nilai B"; jika nilai antara 1 hingga 40, maka termasuk "Nilai C";

Jika nilai tidak memenuhi semua kondisi tersebut, program mengembalikan teks kosong. Hasil dari fungsi ini ditampilkan ke layar dengan pesan yang menunjukkan nilai tersebut termasuk kategori mana.

2. Tugas Looping (Perulangan)

Source code:

```
import 'dart:io';

void main() {

  stdout.write("Masukkan panjang piramida: ");
  int? panjang = int.parse(stdin.readLineSync());

  for (int i = 1; i <= panjang; i++) {

    for (int j = panjang - i; j > 0; j--) {
      stdout.write(" ");
    }

    for (int k = 1; k <= 2 * i - 1; k++) {
      stdout.write("*");
    }

    print("");
  }
}
```

OUTPUT:

```
PS D:\SEMESTER 5\PROJECT PPB\PPB_ADITYA_PRABU_MUKTI_2211104037\03_Pengenalan_Dart\UNGUIDED> dart perulangan.dart
Masukkan panjang piramida: 6
  *
 ***
*****
*****
*****
*****
*****
```

Penjelasan:

Program meminta pengguna memasukkan panjang piramida yang diinginkan. Menggunakan loop for, program mengatur jumlah baris piramida yang akan ditampilkan.

Dalam setiap iterasi, loop kedua mencetak spasi untuk membentuk sisi kiri piramida, sedangkan loop ketiga mencetak bintang yang membentuk bagian atas piramida.

Dengan cara ini, jumlah bintang pada setiap baris meningkat, menciptakan pola piramida yang simetris.

3. Tugas Function

Source code:

```
import 'dart:io';

void main() {

  stdout.write("Masukkan sebuah bilangan bulat: ");
  int? bilangan = int.parse(stdin.readLineSync()!);

  if (isPrima(bilangan)) {
    print("$bilangan adalah bilangan prima");
  } else {
    print("$bilangan bukan bilangan prima");
  }
}

bool isPrima(int n) {
  if (n <= 1) {
    return false;
  }
  for (int i = 2; i <= n ~/ 2; i++) {
    if (n % i == 0) {
      return false;
    }
  }
  return true;
}
```

OUTPUT:

```
PS D:\SEMESTER 5\PROJECT PPB\PPB_ADITYA_PRABU_MUKTI_2211104037\03_Pengenalan_Dart\UNGUIDED> dart function.dart
Masukkan sebuah bilangan bulat: 0
0 bukan bilangan prima
PS D:\SEMESTER 5\PROJECT PPB\PPB_ADITYA_PRABU_MUKTI_2211104037\03_Pengenalan_Dart\UNGUIDED> dart function.dart
Masukkan sebuah bilangan bulat: 5
5 adalah bilangan prima
```

Penjelasan:

Setelah meminta input dari pengguna, program menggunakan fungsi `isPrima` untuk melakukan pengecekan. Fungsi ini akan mengembalikan nilai `false` jika bilangan tersebut kurang dari atau sama dengan 1, dan melakukan pemeriksaan lebih lanjut untuk bilangan di atas 1 dengan mencoba membagi bilangan tersebut dengan angka dari 2 hingga setengah dari bilangan itu sendiri.