

**PRAKTIKUM PEMROGRAMAN PERANGKAT BERGERAK  
TUGAS GUIDED & UNGUIDED**

**MODUL III  
PENGENALAN DART**



**Disusun Oleh :**  
**Rizky Hanifa Afania / 2211104017**  
**SE-06-01**

**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**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## GUIDED

### Pengertian Dart

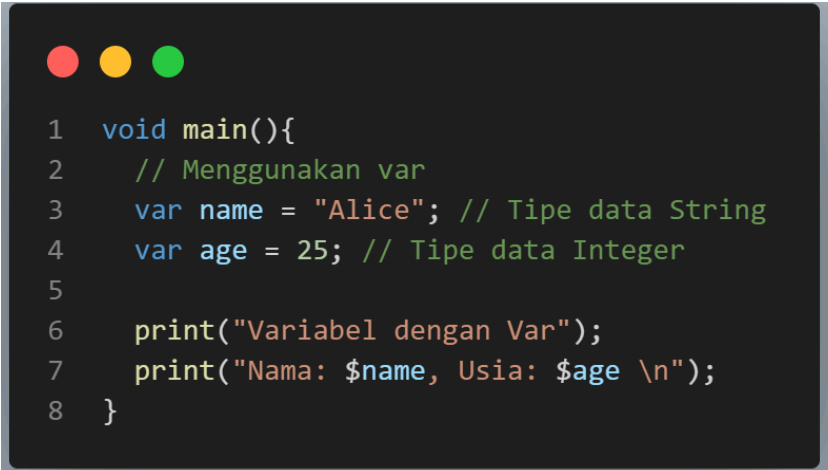
Dart adalah bahasa pemrograman open source yang dirancang untuk berbagai keperluan umum. Dikembangkan oleh Google, Dart ditujukan untuk membangun aplikasi lintas platform seperti mobile, desktop, dan web. Bahasa ini modern dan kaya fitur, dengan banyak kesamaan dengan bahasa populer lainnya seperti Java, C#, JavaScript, Swift, dan Kotlin.

#### a. Variabel

##### - Variabel dengan var

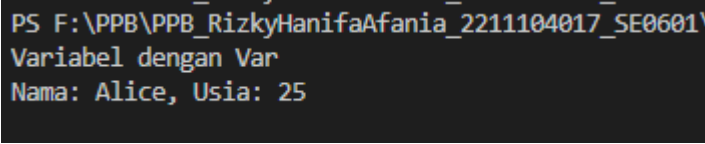
Var adalah cara untuk mendeklarasikan variabel tanpa menentukan tipe datanya secara eksplisit. Dart akan secara otomatis menentukan tipe data berdasarkan nilai yang diberikan.

##### Contoh Kode:

A screenshot of a code editor with a dark background and three colored window control buttons (red, yellow, green) at the top left. The code is written in Dart and uses the 'var' keyword for variable declaration. It includes comments in Indonesian explaining the data types. The code is as follows:

```
1 void main(){
2     // Menggunakan var
3     var name = "Alice"; // Tipe data String
4     var age = 25; // Tipe data Integer
5
6     print("Variabel dengan Var");
7     print("Nama: $name, Usia: $age \n");
8 }
```

##### Hasil:

A screenshot of a terminal window showing the output of the Dart program. The text is as follows:

```
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\
Variabel dengan Var
Nama: Alice, Usia: 25
```

##### - Type Annotation

Type annotation adalah cara untuk mendeklarasikan variabel dengan menentukan tipe datanya secara eksplisit.

### Contoh Kode:

```
1 void main(){
2     // Type annotation
3     String nama = "Bob"; // Tipe data String
4     int umur = 30; // Tipe data Integer
5
6     print("Type annotation");
7     print("Nama: $nama, \nUsia: $umur \n");
8 }
```

### Hasil:

```
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\03_1
Type annotation
Nama: Bob,
Usia: 30
```

### - Multiple Variable

Dart memungkinkan untuk mendeklarasikan beberapa variabel sekaligus dengan tipe yang sama.

### Contoh Kode:

```
1 void main(){
2     // Multiple variable
3     String firstName, lastName; // Tipe data String
4     firstName = "Charlie";
5     lastName = "Brown";
6
7     print("Multiple variable");
8     print("Nama Lengkap: $firstName $lastName \n");
9
10 }
```

### Hasil:

```
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\
Multiple variable
Nama Lengkap: Charlie Brown
```

## b. Statement Control

### - IF-ELSE Statement

Untuk memproses suatu kondisi dalam program, kita dapat menggunakan ekspresi if. Jika hasil dari ekspresi tersebut bernilai true (benar), maka blok kode di dalam if akan dijalankan. Namun, jika kondisi tersebut tidak terpenuhi atau bernilai false (salah), kita dapat menggunakan ekspresi else untuk mengeksekusi blok kode alternatif.

#### Contoh Kode:

```
1 void main(){
2     //Bentuk singkat dari if else
3     var openHours = 8;
4     var closedHours = 21;
5     var now = 8;
6
7     // condition true
8     var shopstatus = now >= openHours && now < closedHours? "Hello, we're open" : "Sorry, we've close";
9     print("If ELse");
10    print(shopstatus );
11
12 }
```


#### Hasil:

```
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\
If ELse
Hello, we're open
```

### - 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.

### Contoh Kode:



```
1  void main(){
2      // Switch-Case Statement
3      print("\nSwitch-Case Statement");
4      var day = 3; // Misalkan 1 = Senin, 2 = Selasa, dst.
5      switch (day) {
6          case 1:
7              print("Senin");
8              break;
9          case 2:
10             print("Selasa");
11             break;
12          case 3:
13             print("Rabu");
14             break;
15          case 4:
16             print("Kamis");
17             break;
18          case 5:
19             print("Jumat");
20             break;
21          case 6:
22             print("Sabtu");
23             break;
24          case 7:
25             print("Minggu");
26             break;
27          default:
28             print("Hari tidak valid");
29      }
30 }
```

### Hasil:

```
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\03
Switch-Case Statement
Rabu
```

### c. Looping

#### - For Loops

Gunakan for loop saat kondisinya tau persis seberapa banyak looping akan dilakukan, contohnya melakukan perulangan sebanyak 10 kali dengan iterasi sebanyak 1 tingkat atau 1 kali.

##### Contoh Kode:

```
1 void main() {
2     // LOOPONG
3     // 1. FOR LOOP : saat mengetahui kapan perulangan berhenti
4     print("\nFor Loops");
5
6     for (int i = 1; i <= 5; i++) {
7         print(i);
8     }
9 }
```

##### Hasil:

```
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\
For Loops
1
2
3
4
5
```

#### - While Loops

Gunakan while loop saat kondisinya tidak tahu kapan perulangan akan berhenti, contohnya sediakan input angka hingga user menginput tanda "-".

##### Contoh Kode:

```
1 void main() {
2     // LOOPONG
3     // 2. WHILE LOOP
4     print("\nWhile Loops");
5
6     int i = 1; // Deklarasi variabel
7     while (i <= 5) {
8         print(i);
9         i++; // Tambahkan 1 ke i setelah setiap iterasi
10    }
11 }
```

**Hasil:**

```
PS F:\PPB\PPB_RizkyHanifaAfanfa_2211104017_SE0601\
While Loops
1
2
3
4
5
```

#### d. List

List adalah tipe data yang digunakan untuk merepresentasikan koleksi nilai yang terurut. List dapat berisi elemen-elemen dengan tipe data yang sama atau berbeda. List dapat diinisialisasi dengan nilai-nilai yang diberikan, atau dapat dibuat kosong dan kemudian diisi kemudian

##### - Fixed Length List

**Contoh Kode:**

```
1 void main() {
2     // LIST
3     // 1. Fixed Length List
4     // Membuat fixed-length list dengan panjang 3
5     List<int> fixedList = List.filled(3, 0); // List dengan 3 elemen, diisi dengan 0
6
7     // Mengubah elemen dalam list
8     fixedList[0] = 10;
9     fixedList[1] = 20;
10    fixedList[2] = 30;
11    print("\nFixed List\n");
12    print(fixedList); // Output: [10, 20, 30]
13 }
```

**Hasil:**

```
PS F:\PPB\PPB_RizkyHanifaAfanfa_2211104017_SE0601\
Fixed List
[10, 20, 30]
```

##### - Growable List

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

### Contoh Kode:

```
1 void main() {
2     // 2. Growable List
3     // Membuat growable list (panjangnya bisa berubah)
4     List<int> growableList = [];
5     // Menambahkan elemen ke dalam list
6     growableList.add(10);
7     growableList.add(20);
8     growableList.add(30);
9     print(growableList); // Output: [10, 20, 30]
10    // Menambahkan lebih banyak elemen
11    growableList.add(40);
12    growableList.add(50);
13    print(growableList); // Output: [10, 20, 30, 40, 50]
14    // Menghapus elemen dari list
15    growableList.remove(20);
16    print(growableList); // Output: [10, 30, 40, 50]
17 }
```

### Hasil:

```
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\
[10, 20, 30]
[10, 20, 30, 40, 50]
[10, 30, 40, 50]
```

### e. Fungsi

Pada bahasa pemrograman yang mendukung Object Oriented Programming, fungsi atau prosedur memiliki peranan yang sangat penting.

### Contoh Kode:

```
1 // FUNGSI
2 // Mendefenisikan Fungsi
3 String sapaan(String nama) {
4     return "Halo, $nama!";
5 }
6
7 void main() {
8     // Mengembalikan nilai
9     String pesan = sapaan("Teman-Teman");
10    print(pesan);
11
12 }
```



**Hasil:**

```
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\  
Halo, Teman-Teman!
```

Tambahkan return apabila anda mendefinisikan sebuah fungsi, contohnya ada pada codingan diatas pada fungsi diatas, number merupakan parameter. Variable diluar fungsi yang dibuat agar dapat digunakan didalam fungsi. Fungsi memiliki scope yang terbatas, tentunya fungsi butuh input dari luar agar program didalamnya bisa memproses tugasnya.

## UNGUIDED

### 1. Tugas Percabangan (Branching)

Buatlah sebuah fungsi dalam Dart yang menerima sebuah nilai dari user, lalu melakukan percabangan untuk memberikan output berdasarkan kondisi berikut:

Deskripsi :

- a. Jika nilai lebih besar dari 70, program harus mereturn "Nilai A".
  - b. Jika nilai lebih besar dari 40 tetapi kurang atau sama dengan 70, program harus mereturn "Nilai B".
  - c. Jika nilai lebih besar dari 0 tetapi kurang atau sama dengan 40, program harus mereturn "Nilai C".
  - d. Jika nilai tidak memenuhi semua kondisi di atas, program harus mereturn teks kosong.
- Sampel Input: 80
  - Sampel Output: 80 merupakan Nilai A
  - Sampel Input: 5
  - Sampel Output: 50 merupakan Nilai

**Kode:**

```
1  import 'dart:io';
2
3  void main() {
4      //Memberi inputan kepada user
5      stdout.write('Masukkan nilai: ');
6      int? nilai = int.parse(stdin.readLineSync()!);
7
8      String hasil(int nilai) {
9          if (nilai > 70 ) {
10             return 'Nilai A';
11          } else if (nilai > 40 && nilai <= 70) {
12             return 'Nilai B';
13          } else if (nilai > 0 && nilai <= 40) {
14             return 'Nilai C';
15          } else {
16             return '';
17          }
18      }
19
20      // Fungsi untuk memanggil hasil nilai
21      String kategoriNilai = hasil(nilai);
22
23      //Mencetak output
24      if (kategoriNilai.isNotEmpty) {
25          print('$nilai merupakan $kategoriNilai');
26      } else {
27          print('Nilai tidak valid');
28      }
29  }
```

### Hasil:

```
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\
Masukkan nilai: 80
80 merupakan Nilai A
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\
Masukkan nilai: 50
50 merupakan Nilai B
```

### Penjelasan:

Program akan meminta input dari pengguna berupa nilai yang akan ditampilkan pada terminal. Nilai yang diinputkan kemudian diubah menjadi tipe data integer. Jika inputnya bukan angka maka program akan memberikan nilai default seperti -1. Setelah nilai berhasil diinput dan dikonversi, nilai tersebut akan diolah menggunakan fungsi. Jika nilai lebih dari 70, fungsi akan mengembalikan 'Nilai A', jika antara 41 dan 70 maka hasilnya 'Nilai B', dan jika antara 1 dan 40 maka hasilnya adalah 'Nilai C'. Jika nilai tidak sesuai dengan kriteria ini, maka akan mengembalikan string kosong. Jikaka tidak valid maka akan muncul pesan bahwa “Nilai tidak valid”.

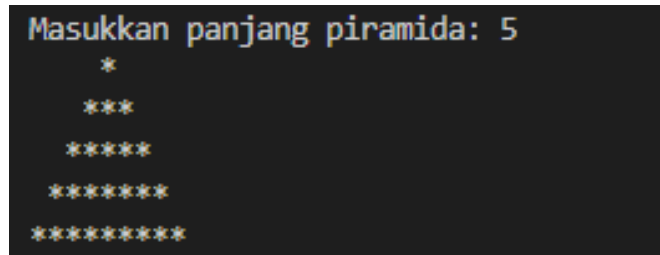
## 2. Tugas Looping (Perulangan)

Buatlah sebuah program dalam Dart yang menampilkan piramida bintang dengan menggunakan for loop. Panjang piramida ditentukan oleh input dari user.

### Kode:

```
1 import 'dart:io';
2
3 void main() {
4     //Meminta inputan kepada user
5     stdout.write('Masukkan panjang piramida: ');
6     int? panjang = int.parse(stdin.readLineSync()!);
7
8     // Perulangan For
9     for (int i = 1; i <= panjang; i++) {
10         // Mencetak spasi
11         stdout.write(' ' * (panjang - i));
12
13         // Mencetak bintang
14         stdout.write('*' * (2 * i - 1));
15
16         // Pindah ke baris berikutnya
17         print('');
18     }
19 }
```

## Hasil:



```
Masukkan panjang piramida: 5
*
***
*****
*****
*****
```

## Penjelasan:

Program meminta input panjang piramida, yaitu jumlah baris yang ingin ditampilkan. Lalu input tersebut diubah menjadi integer. Melalui perulangan for, program mencetak setiap baris piramida. Pada setiap iterasi, baris akan dimulai dengan mencetak spasi yang berkurang sesuai dengan tinggi piramida, diikuti oleh bintang (\*) yang jumlahnya meningkat di setiap baris untuk membentuk piramida. Setelah setiap baris selesai, program akan mencetak baris baru hingga piramida lengkap.

## 3. Tugas Function

Buatlah program Dart yang meminta input berupa sebuah bilangan bulat dari user, kemudian program akan mengecek apakah bilangan tersebut merupakan bilangan prima atau bukan.

- Sampel Input: 23
- Sampel Output: bilangan prima
- Sampel Input: 12
- Sampel Output: bukan bilangan prima

### Kode:

```
1 import 'dart:io';
2
3 void main() {
4     //Meminta inputan kepada user
5     stdout.write('Masukkan Bilangan Bulat: ');
6     int? bilangan = int.parse(stdin.readLineSync()!);
7
8     // Fungsi untuk mengecek apakah bilangan yang diinput user
9     bool cekPrima(int bilangan) {
10     if (bilangan <= 1) {
11         return false; // Bilangan kurang dari atau sama dengan 1 = bukan bilangan prima
12     }
13
14     for (int i = 2; i <= bilangan ~/ 2; i++) {
15         if (bilangan % i == 0) {
16             return false; // Jika ada pembagi selain 1 dan dirinya sendiri = bukan prima
17         }
18     }
19     return true; // Jika tidak ada pembagi selain 1 dan dirinya sendiri = bilangan prima
20 }
21
22 // Cek termasuk bilangan prima atau bukan
23 if (cekPrima(bilangan)) {
24     print('$bilangan adalah bilangan prima');
25 } else {
26     print('$bilangan bukan bilangan prima');
27 }
28 }
```

### Hasil:

```
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\
Masukkan Bilangan Bulat: 23
23 adalah bilangan prima
PS F:\PPB\PPB_RizkyHanifaAfania_2211104017_SE0601\
Masukkan Bilangan Bulat: 12
12 bukan bilangan prima
```

### Penjelasan:

Program meminta input berupa bilangan bulat dari pengguna, yang akan diubah menjadi tipe integer. Fungsi cekPrima() digunakan untuk memeriksa apakah bilangan tersebut prima. Jika bilangan kurang dari atau sama dengan 1, fungsi mengembalikan false karena bilangan tersebut bukan prima. Selanjutnya, perulangan for digunakan untuk mencari pembagi bilangan dari 2 hingga setengah dari bilangan yang diinput.

Jika ditemukan bilangan pembagi selain 1 dan bilangan itu sendiri, fungsi juga mengembalikan false. Jika tidak ada pembagi, fungsi mengembalikan true, yang berarti bilangan tersebut adalah prima. Kemudian program mencetak hasil apakah bilangan tersebut merupakan bilangan prima atau bukan berdasarkan nilai yang dikembalikan oleh

fungsi cekPrima().