

LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL 3
PENGENALAN DART



Disusun Oleh :
YOGI HAFIDH MAULANA/2211104061
SE062

Asisten Praktikum :
Ayu Susilowati
Noviana Rizki Anisa Putri

Dosen Pengampu :
Yudha Islami Sulistya

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2024

A. GUIDED

- **Deklarasi Variabel**

Code

```
// Deklarasi Variable
String firstName = "Praktikum"; // Variable dengan keyword tipe data
var lastName = "PBB"; // Variable dengan Var
final cuaca = "cerah"; // The value of the local variable 'cuaca' is

// Melihat Output
print("Hari ini $firstName $lastName");
```

Output

```
[Running] dart "d:\PROJECT\Flutter Project\Praktikum\praktikum3\guided_03.dart"
Hari ini Praktikum PBB
```

Deskripsi Program

1. var adalah cara untuk mendeklarasikan variabel tanpa menentukan tipe datanya secara eksplisit. Dart akan secara otomatis menentukan tipe data berdasarkan nilai yang diberikan. Dalam contoh diatas adalah
2. Type annotation adalah cara untuk mendeklarasikan variabel dengan menentukan tipe datanya secara eksplisit. Pada contoh diatas yaitu String adalah tipe data yang secara eksplisit menyatakan bahwa variabel akan menyimpan teks.

Tipe	Deskripsi	Contoh
int	Integer (bilangan bulat)	1, -2, 0
double	Bilangan desimal	3.14, 10.53, -23.23
num	Bilangan bulat dan bilangan desimal	7, 3.14, -12.00
bool	Boolean	True, False
String	Text yang terdiri dari 0 atau beberapa karakter	'Faza', 'Hasna'
List	Sekuumpulan nilai	[0,1,2,3,4], ['a', 'b', 'c']
Map	Pasangan key-value	{ "a": "aku", "b": 12 }
dynamic	Tipe apapun	

3. final: Menyatakan bahwa nilai variabel tidak dapat diubah setelah inisialisasi.

- **Statement Control**

Code

If else

```
10 // IF-ELSE Statement
11 var open = 8;
12 var close = 17;
13 var now = 10;
14
15 if (now >= open && now < close) {
16     print("Toko Buka");
17 } else if (now == 12) {
18     print("Toko sedang istirahat");
19 } else {
20     print("Toko Tutup");
21 }
```

```
[Running] dart "d:\PROJECT\Flutter Project\Praktikum\praktikum3\guided_03.dart"
Toko Buka
```

Switchcase

```
23 // SWITCH CASE Statement
24 var nilai = "b";
25 switch (nilai) {
26     case "a":
27         print("Nilai sangat bagus");
28         break;
29     case "b":
30         print("Nilai bagus");
31         break;
32     case "c":
33         print("Nilai cukup");
34         break;
35     default:
36         print("Nilai tidak tersedia");
37 }
```

```
[Running] dart "d:\PROJECT\Flutter Project\Praktikum\praktikum3\guided_03.dart"
Nilai bagus
```

Ternary

```
23 // TERNARY Oprator
24 var toko = now > open ? "Toko Buka" : "Toko Tutup";
25 print(toko);
```

```
[Running] dart "d:\PROJECT\Flutter Project\Praktikum\praktikum3\guided_03.dart"
Toko Buka
```

Deskripsi Program If else

Kode tersebut mengecek nilai variabel now yang menyatakan waktu saat ini.

1. Pertama, kode mengecek apakah now (waktu saat ini) berada di antara nilai variabel open (jam buka) dan close (jam tutup) menggunakan kondisi if (now >= open && now < close).
2. Jika kondisi tersebut benar, maka akan mencetak "Toko Buka".
3. Jika tidak, kode akan mengecek apakah now sama dengan 12 menggunakan kondisi else if (now == 12).
4. Jika now sama dengan 12, maka akan mencetak "Toko sedang istirahat".
5. Jika tidak ada kondisi di atas yang terpenuhi, maka akan mencetak "Toko Tutup".

Deskripsi Program Switch Case

Kode ini menggunakan switch case statement untuk memeriksa nilai variabel nilai. Switch statement akan membandingkan nilai dari variabel nilai dengan setiap case. Jika nilai sama dengan "a", maka akan mencetak "Nilai sangat bagus". Jika nilai sama dengan "b", maka akan mencetak "Nilai bagus". Jika nilai sama dengan "c", maka akan mencetak "Nilai cukup". Jika tidak ada case yang cocok, maka bagian default akan dieksekusi dan mencetak "Nilai tidak tersedia".

Deskripsi Ternary Operator

Kode diatas menggunakan ternary operator untuk menetapkan nilai variabel toko. Jika kondisi now > open (waktu saat ini lebih besar dari jam buka) benar, maka toko akan diisi dengan string "Toko Buka". Jika kondisi tersebut salah, toko akan diisi dengan "Toko Tutup". Setelah itu, kode mencetak nilai dari variabel toko.

- **Looping**

Code

For Loop

```
43 // FOR LOOP
44 for (var i = 0; i <= 5; i++) {
45     print(i);
46 }
```

[Running] dart "d:\PROJECT\Flutter Project\Praktikum\praktikum3\guided_03.dart"

0
1
2
3
4
5

While Loop

```
48 // WHILE LOOP
49 var i = 0;
50 while (i <= 5) {
51     print("print: $i");
52     i++;
53 }
```

[Running] dart "d:\PROJECT\Flutter Project\Praktikum\praktikum3\guided_03.dart"

print: 0
print: 1
print: 2
print: 3
print: 4
print: 5

Deskripsi for loop

Cara Kerja: Menginisialisasi variabel i dengan 0. Memeriksa apakah i kurang dari atau sama dengan 5. Jika benar, mencetak nilai i dan menambah i sebesar 1.

Proses ini diulang hingga i lebih besar dari 5. Output: Angka 0 hingga 5 dicetak satu per satu.

Deskripsi While loop

Cara Kerja: Menginisialisasi i dengan 0. Memeriksa apakah i kurang dari atau sama dengan 5. Jika benar, mencetak "print: \$i" dan menambah i sebesar 1. Proses ini berlanjut hingga i lebih besar dari 5. Output: Mencetak "print: 0" hingga "print: 5".

- **List**
Code

Fixed Length List

```
55 // LIST
56 List<int> fixesList = List.filled(3, 0);
57 fixesList[0] = 10;
58 fixesList[1] = 20;
59 fixesList[2] = 30;
60
61 print("Fixed Length List: $fixesList");
```

[Running] dart "d:\PROJECT\Flutter Project\Praktikum\praktikum3\guided_03.dart"

Fixed Length List: [10, 20, 30]

Growable List

```
66 List<int> growableList = [];  
67  
68 growableList.add(10);  
69 growableList.add(20);  
70 growableList.add(30);  
71 print("Growable List: $growableList");  
72  
73 growableList.add(50);  
74 growableList.add(60);  
75 growableList.add(70);  
76 print(growableList);  
[Running] dart "d:\PROJECT\Flutter Project\Praktikum\praktikum3\guided_03.dart"  
Growable List: [10, 20, 30]  
[10, 20, 30, 50, 60, 70]
```

Deskripsi List Fixed Length List

Inisialisasi: Membuat list `fixesList` dengan panjang tetap 3 dan diisi dengan nilai 0.

Pengisian Nilai: Mengisi elemen-elemen list dengan nilai 10, 20, dan 30.

Output: Mencetak isi dari `fixesList`, yang sekarang adalah [10, 20, 30].

Penghapusan Elemen: Mencoba menghapus elemen pada indeks 30. Masalah: Karena `fixesList` hanya memiliki 3 elemen (indeks 0, 1, dan 2), mencoba menghapus elemen pada indeks 30 akan menyebabkan error (index out of range).

Fixed Length List: List dengan panjang tetap yang tidak bisa diubah setelah inisialisasi. Error Handling: Kode ini akan menghasilkan error saat mencoba menghapus elemen di luar batas list. Sebaiknya menggunakan indeks yang valid untuk menghindari error.

Deskripsi Growable List

Inisialisasi: Membuat list `growableList` yang dapat berkembang (growable) dengan awal kosong.

Menambahkan Elemen: Menambahkan elemen 10, 20, dan 30 ke dalam list.

Output: Mencetak isi dari `growableList`, yang sekarang adalah [10, 20, 30].

Menambahkan Elemen Tambahan: Menambahkan elemen 50, 60, dan 70 ke dalam list.

Output: Mencetak isi terbaru dari `growableList`, yang sekarang adalah [10, 20, 30, 50, 60, 70].

Growable List: List yang dapat bertambah panjang saat elemen baru ditambahkan. Fleksibilitas: Memungkinkan untuk menambah elemen kapan saja tanpa batasan ukuran awal.

- **Fungsi**
Code

```
75 // FUNGSI
76 // Mendefenisikan Fungsi
77 void cetakPesan(String pesan) {
78   ..print(pesan);
79 }
80
81 // Mengembalikan Nilai
82 int perkalian(int a, int b) {
83   ..return a * b;
84 }
85
86 // Menambahkan Parameter
87 void greet(String name, int age) {
88   ..print('Hello $name, you are $age years old. ');
89 }
90
91 // Memanggil Fungsi
92 void mainfunction() {
93   ..int hasil = perkalian(5, 4);
94   ..print("Hasil perkalian fungsi: $hasil");
95
96   ..cetakPesan("Halo, Guys!");
97   ..greet('Alice', 25);
98 }
99
```

Output

```
[Running] dart "d:\PROJECT\Flutter Project\Praktikum\praktikum3\guided_03.dart"
Hasil perkalian fungsi: 20
Halo, Guys!
```

Deskripsi

1. Fungsi cetakPesan: Menerima parameter bertipe String dan mencetak pesan yang diterima.
2. Fungsi perkalian: Menerima dua parameter bertipe int, mengalikan keduanya, dan mengembalikan hasilnya.
3. Fungsi greet: Menerima dua parameter, name bertipe String dan age bertipe int, lalu mencetak sapaan yang menyertakan nama dan usia.
4. Fungsi mainfunction: Memanggil fungsi-fungsi yang telah didefinisikan sebelumnya. Menghitung hasil perkalian 5 dan 4 dengan fungsi perkalian. Mencetak pesan menggunakan fungsi cetakPesan. Menggreet pengguna dengan nama 'Alice' dan usia 25 menggunakan fungsi greet.

B. Unguided

1. Soal 1

Code

```
1  import 'dart:io';
2
3  // Fungsi untuk mengecek nilai dan mengembalikan grade
4  String checkNilai(int nilai) {
5    if (nilai > 70) {
6      return 'Nilai A';
7    } else if (nilai > 40 && nilai <= 70) {
8      return 'Nilai B';
9    } else if (nilai > 0 && nilai <= 40) {
10     return 'Nilai C';
11   } else {
12     return '';
13   }
14 }
15
16 Run | Debug
17 void main() {
18   // Meminta input dari user
19   stdout.write('Masukkan nilai: ');
20   int? nilai = int.tryParse(stdin.readLineSync() ?? '');
21
22   // Memeriksa apakah input valid
23   if (nilai != null) {
24     // Memanggil fungsi checkNilai dengan nilai input dari user
25     String hasil = checkNilai(nilai);
26     // Menampilkan hasil
27     if (hasil.isNotEmpty) {
28       print('$nilai merupakan $hasil');
29     } else {
30       print('Nilai tidak valid.');
```


Output

```
D:\PROJECT\Flutter Project\Praktikum\praktikum3>dart run Soal1.dart
Masukkan nilai: 80
80 merupakan Nilai A

D:\PROJECT\Flutter Project\Praktikum\praktikum3>dart run Soal1.dart
Masukkan nilai: 50
50 merupakan Nilai B
```

Penjelasan

Mengimpor library `dart:io` untuk memungkinkan penggunaan fungsi input dan output. Meminta pengguna untuk memasukkan tinggi piramida. Menggunakan `stdin.readLineSync()` untuk membaca input, kemudian mengonversinya menjadi integer dengan `int.tryParse()`. Memeriksa apakah input valid. Jika nilai tinggi null atau kurang dari atau sama dengan 0, program mencetak pesan kesalahan dan berhenti. Menggunakan nested loops untuk mencetak piramida: Loop pertama (`for (int i = 1; i <= tinggi; i++)`) mengatur tingkat piramida. Loop kedua (`for (int j = 1; j <= tinggi - i; j++)`) mencetak spasi untuk perataan. Loop ketiga (`for (int k = 1; k <= (2 * i - 1); k++)`) mencetak bintang berdasarkan tingkat.

2. Soal 2

Code

```
1  import 'dart:io';
2
3  void main() {
4      // Meminta input untuk tinggi piramida
5      stdout.write('Masukkan tinggi piramida: ');
6      int? tinggi = int.tryParse(stdin.readLineSync() ?? '');
7
8      // Memeriksa apakah input valid
9      if (tinggi == null || tinggi <= 0) {
10         print('Input tidak valid. Masukkan angka yang lebih besar dari 0.');
```

Type: String

```
11     return;
12 }
13
14 // Menampilkan piramida
15 for (int i = 1; i <= tinggi; i++) {
16     // Mencetak spasi
17     for (int j = 1; j <= tinggi - i; j++) {
18         stdout.write(' ');
19     }
20     // Mencetak bintang
21     for (int k = 1; k <= (2 * i - 1); k++) {
22         stdout.write('*');
23     }
24     // Pindah ke baris baru
25     print('');
26 }
27 }
28
```

Output

```
D:\PROJECT\Flutter Project\Praktikum\praktikum3>dart run Soal2.dart
Masukkan tinggi piramida: 10

      *
     ***
    *****
   ********
  **********
 **********
 **********
 **********
 **********
 **********
 **********
```

Deskripsi

Mengimpor library `dart:io` untuk operasi input dan output. Meminta pengguna untuk memasukkan tinggi piramida. Input dibaca dan diubah menjadi integer. Memeriksa apakah input valid. Jika tidak, program mencetak pesan kesalahan dan berhenti. Menggunakan nested loops untuk mencetak piramida: Loop pertama mengontrol tingkat piramida. Loop kedua mencetak spasi untuk penataan. Loop ketiga mencetak bintang sesuai dengan tingkat.

3. Soal 3

Code

```

1 import 'dart:io';
2
3 Run | Debug
4 void main() {
5     // Meminta input dari user
6     stdout.write('Masukkan sebuah bilangan bulat: ');
7     int? angka = int.tryParse(stdin.readLineSync() ?? '');
8
9     // Memeriksa apakah input valid
10    if (angka == null || angka <= 0) {
11        print('Input tidak valid. Masukkan bilangan bulat positif.');
```

```

12    }
13
14    // Memanggil fungsi untuk memeriksa apakah bilangan prima atau bukan
15    bool isPrima = cekBilanganPrima(angka);
16
17    // Menampilkan hasil
18    if (isPrima) {
19        print('$angka adalah bilangan prima.');
```

```

20    } else {
21        print('$angka bukan bilangan prima.');
```

```

22    }
23 }
24
25 // Fungsi untuk memeriksa apakah bilangan prima
26 bool cekBilanganPrima(int angka) {
27     if (angka < 2) return false;
28     for (int i = 2; i <= angka ~/ 2; i++) {
29         if (angka % i == 0) {
30             return false;
31         }
32     }
33     return true;
34 }
```

Output

```
D:\PROJECT\Flutter Project\Praktikum\praktikum3>dart run Soal3.dart
Masukkan sebuah bilangan bulat: 23
23 adalah bilangan prima.

D:\PROJECT\Flutter Project\Praktikum\praktikum3>dart run Soal3.dart
Masukkan sebuah bilangan bulat: 12
12 bukan bilangan prima.
```

Deskripsi

Meminta pengguna untuk memasukkan bilangan bulat positif. Input dibaca dan diubah menjadi integer menggunakan `int.tryParse()`. Memeriksa apakah input valid. Jika angka null atau kurang dari atau sama dengan 0, program mencetak pesan kesalahan dan berhenti. Memanggil fungsi `cekBilanganPrima` untuk memeriksa apakah bilangan yang dimasukkan adalah bilangan prima, dan menyimpan hasilnya dalam variabel `isPrima`. Menampilkan hasil pemeriksaan: jika `isPrima` bernilai `true`, mencetak bahwa bilangan tersebut adalah bilangan prima; jika tidak, mencetak bahwa bilangan tersebut bukan bilangan prima. Fungsi ini memeriksa apakah angka adalah bilangan prima. Jika angka kurang dari 2, fungsi langsung mengembalikan `false`. Menggunakan loop untuk memeriksa apakah angka dapat dibagi oleh angka lain dari 2 hingga setengah dari angka ($\text{angka} \sim / 2$). Jika ada angka yang dapat membagi habis, fungsi mengembalikan `false`. Jika tidak ada angka yang membagi habis, fungsi mengembalikan `true`.