

LAPORAN PRAKTIKUM
PEMROGRAMAN PERANGKAT BERGERAK

MODUL 3
PENGENALAN DART



Disusun Oleh :
SATRIA ARIQ ADELARD
DOMPAS/2211104033SE062

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

```
bin > guided_03.dart > main
Run | Debug | Codeium: Refactor | Explain | Generate Function Comment | X
1 void main() {
2     // DEKLARASI VARIABLE
3     String firstName = "Praktikum"; // Variable dengan Type Annotation
4     var lastName = "PBB"; // Variable dengan Var
5     final cuaca = "cerah"; // Variable dengan final
6
7     // MELIHAT OUTPUT
8     print("Hari ini $firstName $lastName");
}
```

Output

```
OUTPUT  DEBUG CONSOLE  PROBLEMS  2  TERMINAL  PORTS  COMMENTS
Connecting to VM Service at ws://127.0.0.1:49984/IOyCluvhHTo=/ws
Connected to the VM Service.
Hari ini Praktikum PBB

Exited.
```

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	Sekumpulan 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

Switchcase

```
// SWITCH CASE STATEMENT
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 DEBUG CONSOLE PROBLEMS 1 TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:50152/q070Q6_y2MA=/ws
Connected to the VM Service.
Nilai bagus

Exited.

Ternary

```
var open = 8;
var close = 17;
var now = 10;

if (now >= open && now < close) {
  print("Toko Buka");
} else if (now == 12) {
  print("Toko sedang istirahat");
} else {
  print("Toko Tutup");
}

// TERNARY OPRATOR
var toko = now > open ? "Toko Buka" : "Toko Tutup";
print(toko);
```

OUTPUT DEBUG CONSOLE PROBLEMS 1 TERMINAL PORTS COMMENTS

Connecting to VM Service at ws://127.0.0.1:50214/03Yn43rbLUI=/ws
Connected to the VM Service.
2 Toko Buka

Exited.

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".

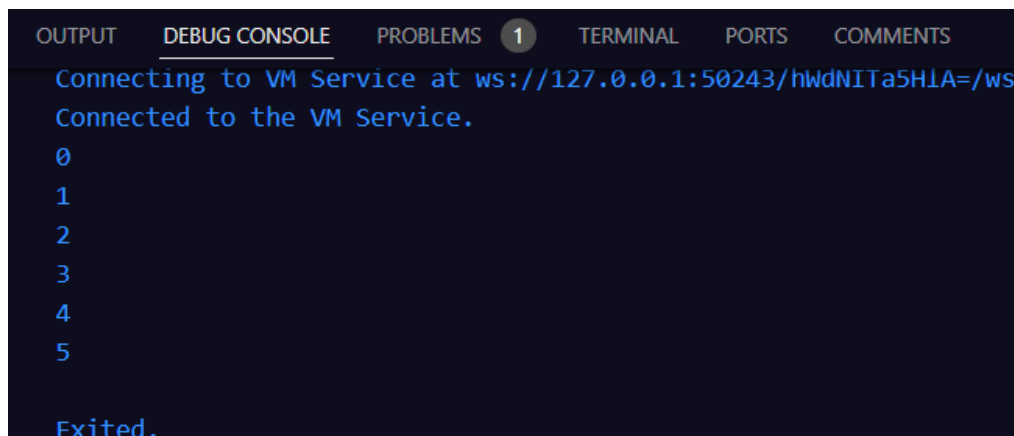
Deskripsi Ternary Operator

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

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



```
OUTPUT  DEBUG CONSOLE  PROBLEMS  1  TERMINAL  PORTS  COMMENTS
Connecting to VM Service at ws://127.0.0.1:50243/hwdNITa5H1A=ws
Connected to the VM Service.
0
1
2
3
4
5
Exited.
```

While Loop

```
// WHILE LOOP
var i = 0;
while (i <= 5) {
    print("print: $i");
    i++;
}
```

```
OUTPUT  DEBUG CONSOLE  PROBLEMS  1  TERMINAL  PORTS  COMMENTS
Connecting to VM Service at ws://127.0.0.1:50267/Co5PTru3rZs=/ws
Connected to the VM Service.
print: 0
print: 1
print: 2
print: 3
print: 4
print: 5

Exited.
```

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

```
// LIST
List<int> fixesList = List.filled(3, 0); // Fixed Length List
fixesList[0] = 10;
fixesList[1] = 20;
fixesList[2] = 30;
print("Fixed Length List: $fixesList");
fixesList.removeAt(30);
print("After removing element at index 30: $fixesList");
```

```
OUTPUT  DEBUG CONSOLE  PROBLEMS  1  TERMINAL  PORTS  COMMENTS
Connecting to VM Service at ws://127.0.0.1:50319/pojF5kTkNYc=/ws
Connected to the VM Service.
Fixed Length List: [10, 20, 30]
```

Growable List

```
List<int> growableList = []; // Growable List
growableList.add(10);
growableList.add(20);
growableList.add(30);
print("Growable List: $growableList");
growableList.add(50);
growableList.add(60);
growableList.add(70);
print(growableList);
```

OUTPUT DEBUG CONSOLE PROBLEMS 1 TERMINAL PORTS COMMENTS

```
Connecting to VM Service at ws://127.0.0.1:50502/p72JptF3xSI=/ws
Connected to the VM Service.
Growable List: [10, 20, 30]
[10, 20, 30, 50, 60, 70]

Exited.
```

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**

```
bin > fungsi.dart > main
1 // Fungsi tanpa nilai kembalian
  Codeium: Refactor | Explain | Generate Function Comment | X
2 void cetakPesan(String pesan) {
3   print(pesan); // Mencetak pesan
4 }
5
6 // Fungsi dengan nilai kembalian
  Codeium: Refactor | Explain | Generate Function Comment | X
7 int kaliAngka(int a, int b) {
8   return a * b; // Mengembalikan hasil perkalian
9 }
10
  Run | Debug | Codeium: Refactor | Explain | Generate Function Comment | X
11 void main() {
12   // Memanggil fungsi dan menyimpan hasilnya
13   int hasil = kaliAngka(4, 5);
14
15   // Menampilkan hasil
16   print('Hasil kali: $hasil');
17
18   // Memanggil fungsi tanpa nilai kembalian
19   cetakPesan('Halo, Dart!');
20
21   // Fungsi greet
22   greet('Alice', 25);
23 }
24
  Codeium: Refactor | Explain | Generate Function Comment | X
25 void greet(String name, int age) {
26   print('Hello $name, you are $age years old.');
```

Output

```
OUTPUT  DEBUG CONSOLE  PROBLEMS 1  TERMINAL  PORTS  COMMENTS
Connecting to VM Service at ws://127.0.0.1:50704/XVyvNgA4myI=/ws
Connected to the VM Service.
Hasil kali: 20
Halo, Dart!
Hello Alice, you are 25 years old.

Exited.
```

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

```
import 'dart:io';
```

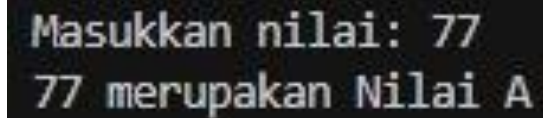
Codeium: Refactor | Explain | Generate Function Comment | ✕

```
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 "";  
  }  
}
```

Run | Debug | Codeium: Refactor | Explain | Generate Function Comment | ✕

```
void main() {  
  stdout.write('Masukkan nilai: ');  
  int nilai = int.parse(stdin.readLineSync()!);  
  String hasil = cekNilai(nilai);  
  if (hasil.isNotEmpty) {  
    print('$nilai merupakan $hasil');  
  } else {  
    print('Nilai tidak valid');  
  }  
}
```


Output



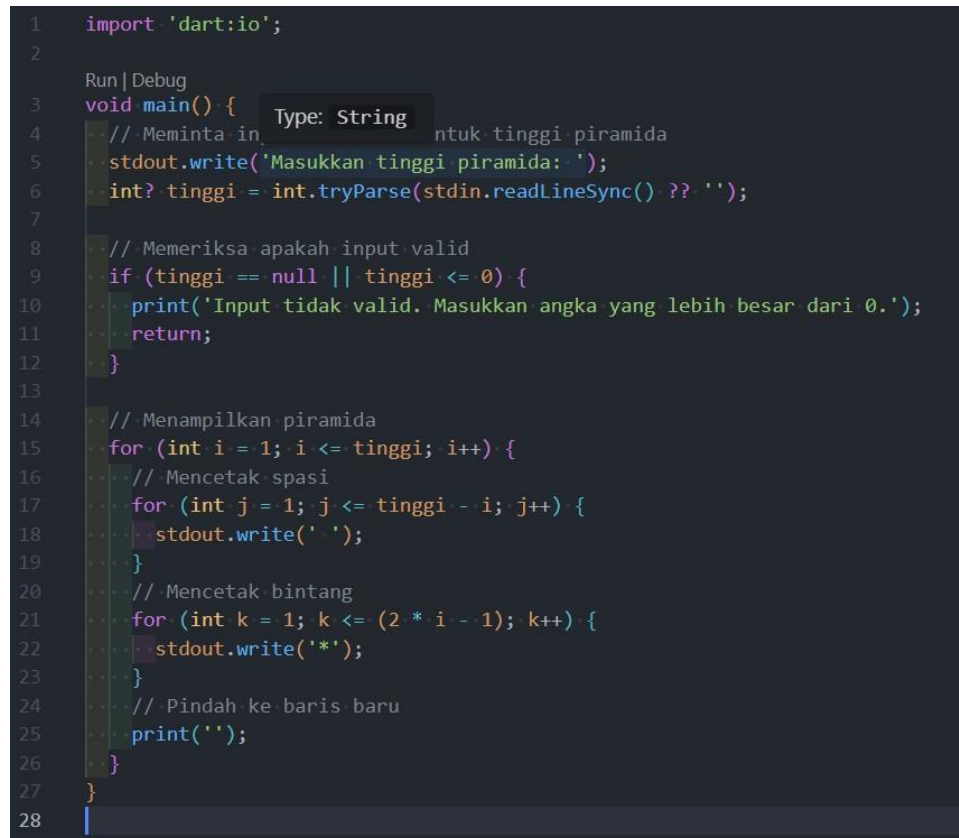
```
Masukkan nilai: 77
77 merupakan Nilai A
```

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. ');
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

```
Masukkan tinggi piramida: 8
      *
     ***
    *****
   *********
  ***********
 *****
*****
*****
*****
```

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

```
import 'dart:io';
```

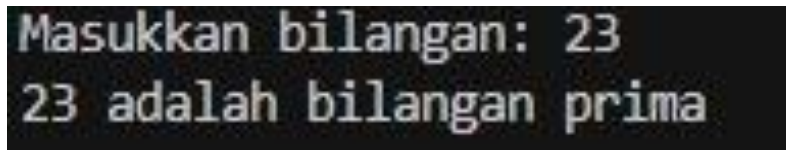
Codeium: Refactor | Explain | Generate Function Comment | ✕

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

Run | Debug | Codeium: Refactor | Explain | Generate Function Comment | ✕

```
void main() {
  stdout.write('Masukkan bilangan: ');
  int bilangan = int.parse(stdin.readLineSync());
  if (cekPrima(bilangan)) {
    print('$bilangan adalah bilangan prima');
  } else {
    print('$bilangan bukan bilangan prima');
  }
}
```

Output



```
Masukkan bilangan: 23
23 adalah bilangan prima
```

Deskripsi

Minta pengguna untuk memasukkan bilangan bulat positif.

Input dibaca dan diubah menjadi bilangan bulat menggunakan `int.TryParse()`.

Periksa apakah inputnya valid atau tidak.

Jika angkanya 0 atau kurang dari atau sama dengan 0, program akan mencetak pesan kesalahan dan berhenti.

Panggil fungsi `checkPrimeNumber` untuk memeriksa apakah bilangan yang dimasukkan adalah bilangan prima atau bukan dan simpan hasilnya dalam variabel `isPrime`.

Tampilkan hasil pengujian: jika `isPrime` benar, menunjukkan bahwa bilangan tersebut adalah bilangan prima; sebaliknya menunjukkan bahwa bilangan tersebut bukan bilangan prima.

Fungsi ini memeriksa apakah suatu bilangan prima atau bukan.

Jika angkanya kurang dari 2, fungsi akan langsung mengembalikan `false`.

Gunakan perulangan untuk memeriksa apakah suatu bilangan habis dibagi oleh bilangan lain antara 2 dan setengah dari bilangan tersebut (angka $\sim/2$).

Jika ada bilangan genap, fungsi mengembalikan nilai salah.

Jika tidak ada bilangan yang dapat membagi secara merata, fungsi tersebut akan mengembalikan nilai `true`.