

**LAPORAN PRAKTIKUM  
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL 3  
PENGENALAN DART**



**Disusun Oleh :**

**Dimas Cahyo Margono / 2211104060**

**SE-06-02**

**Asisten Praktikum :**

**Muhammad Faza Zulian Gesit Al Barru**

**Aisyah Hasna Aulia**

**Dosen Pengampu :**

**Yudha Islami Sulistya**

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

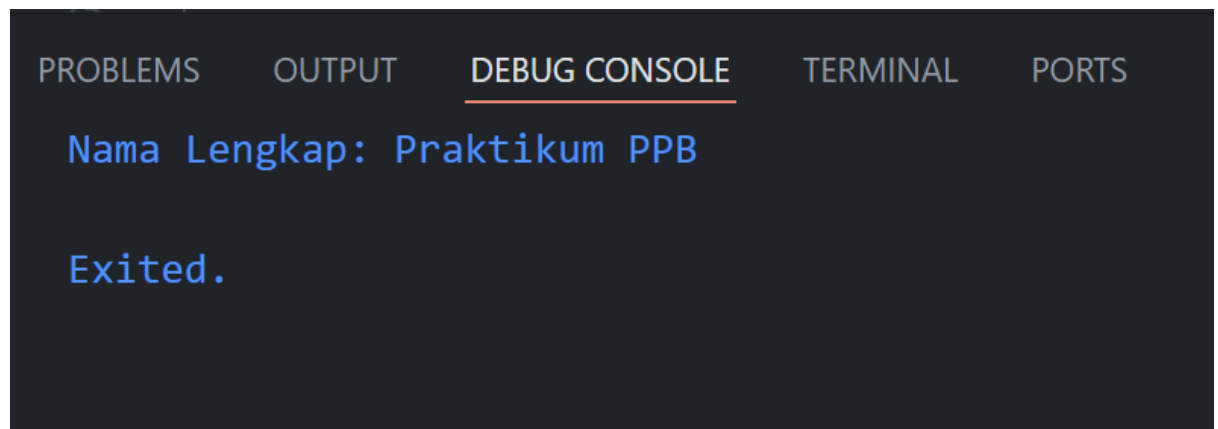
## GUIDED

### A. PENAMAAN VARIABEL DENGAN CAMELCASE

#### Sourcecode

```
void main() {  
    // 1. Penamaan variabel menggunakan camelCase  
    String firstName, lastName; // Tipe data String  
    firstName = "Praktikum";  
    lastName = "PPB";  
  
    print("Nama Lengkap: $firstName $lastName");  
}
```

#### Screenshoot Output



#### Deskripsi Program

Kode tersebut merupakan program sederhana dalam bahasa Dart yang dimulai dengan fungsi utama `void main()` sebagai titik awal eksekusi. Program ini mendeklarasikan dua variabel bertipe `String`, yaitu `firstName` dan `lastName`, yang masing-masing diinisialisasi dengan nilai "Praktikum" dan "PPB". Kemudian, program mencetak teks ke konsol menggunakan fungsi `print` dengan format `print("Nama Lengkap: \$firstName \$lastName");`, yang menggunakan sintaks interpolasi string (`\$`) untuk menyisipkan nilai dari variabel `firstName` dan `lastName`. Hasil akhir yang dicetak ke konsol adalah: `Nama Lengkap: Praktikum PPB`.

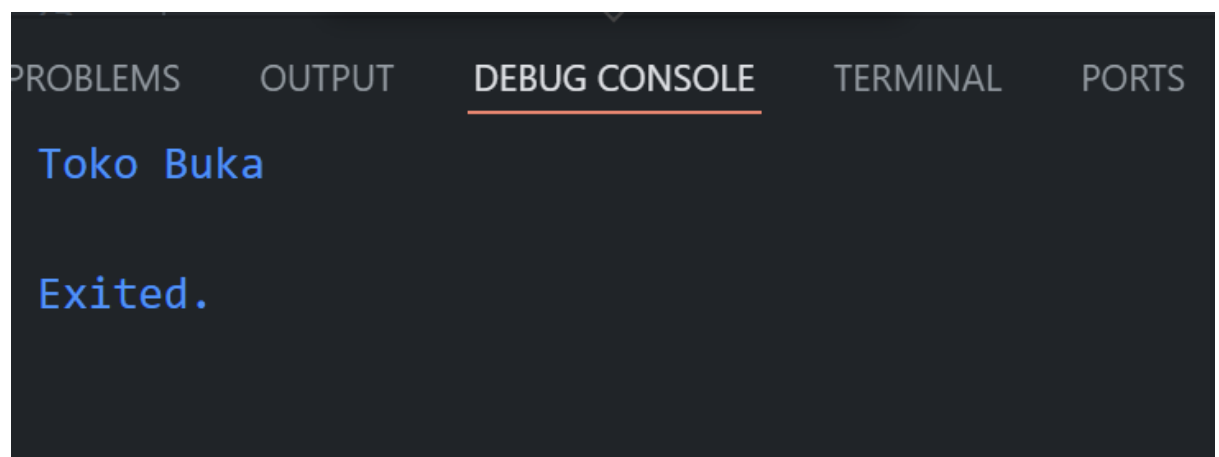
## B. STATEMENT IF ELSE

### Sourcecode

```
// 2. Statement If-Else
var open = 8;
var close = 17;
var now = 12;

if (now ≥ open && now < close) {
    print("Toko Buka");
} else if (now == 12) {
    print("Toko Sedang Istirahat");
} else {
    print("Toko Tutup");
}
```

### Screenshoot Output




## Deskripsi Program

Kode tersebut merupakan program Dart yang menggunakan struktur kontrol if-else\*untuk menentukan status toko berdasarkan variabel `open`, `close`, dan `now`. Variabel `open` dan `close` menyimpan jam buka (8) dan jam tutup (17) toko, sedangkan `now` mewakili waktu saat ini (12). Program mengecek kondisi `if (now >= open && now < close)` untuk menentukan apakah `now` berada dalam rentang waktu buka toko; jika benar, maka akan mencetak "Toko Buka". Jika kondisi tersebut tidak terpenuhi, maka program memeriksa kondisi `else if (now == 12)` untuk mengecek apakah `now` sama dengan 12 (jam istirahat), dan jika benar akan mencetak "Toko Sedang Istirahat". Jika kedua kondisi sebelumnya tidak terpenuhi, maka akan mengeksekusi blok `else` dan mencetak "Toko Tutup".

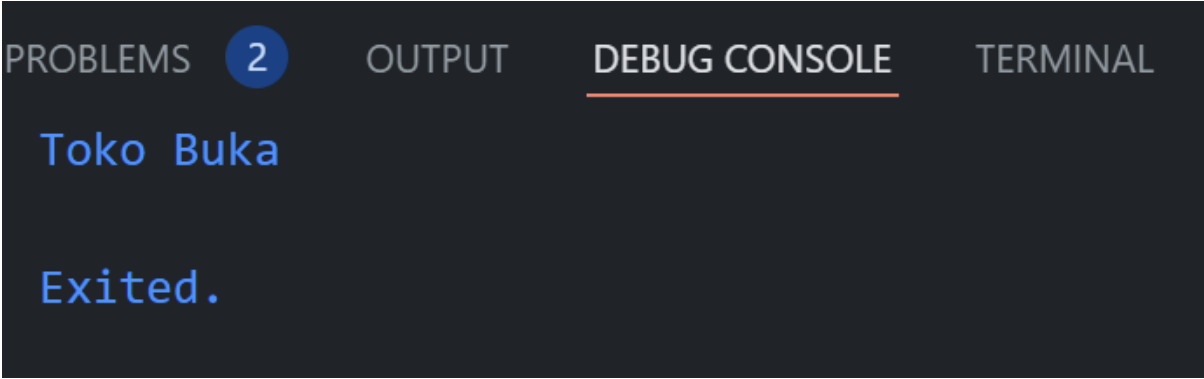
## C. CONDITION TRUE FALSE

### Sourcecode



```
// Condition ? true ; false
var toko = now > open ? "Toko Buka" :
"Toko Tutup";
print(toko);
```

### Screenshoot Output



PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL

Toko Buka

Exited.

### Deskripsi program

Kode tersebut menggunakan operator ternary untuk mengevaluasi apakah waktu saat ini (now) lebih besar dari waktu buka (open). Jika kondisi `now > open` bernilai benar, variabel `toko` akan diisi dengan string "Toko Buka", sedangkan jika kondisi tersebut salah, variabel `toko` akan diisi dengan "Toko Tutup". Setelah penentuan nilai tersebut, program mencetak hasilnya ke konsol menggunakan `print(toko)`, sehingga output yang dihasilkan adalah status "Toko Buka" jika `now` lebih besar dari `open`, atau "Toko Tutup" jika sebaliknya.

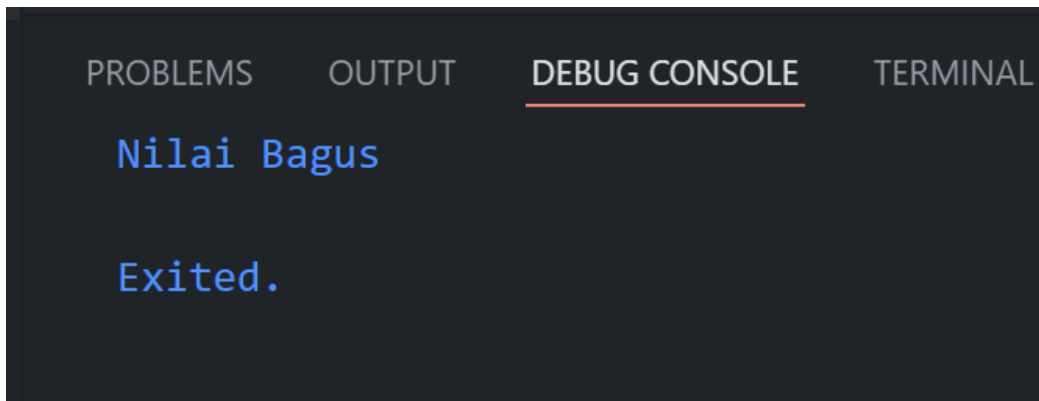
## D. SWITCH CASE

### Sourcecode

```
var grade = 'b';

switch (grade) {
  case 'a':
    print("Nilai Sangat Bagus");
    break;
  case 'b':
    print("Nilai Bagus");
  case 'c':
    print("Nilai Cukup");
    break;
  default:
    print("Nilai tidak tersedia");
}
}
```

### Screenshoot output



### Deskripsi program

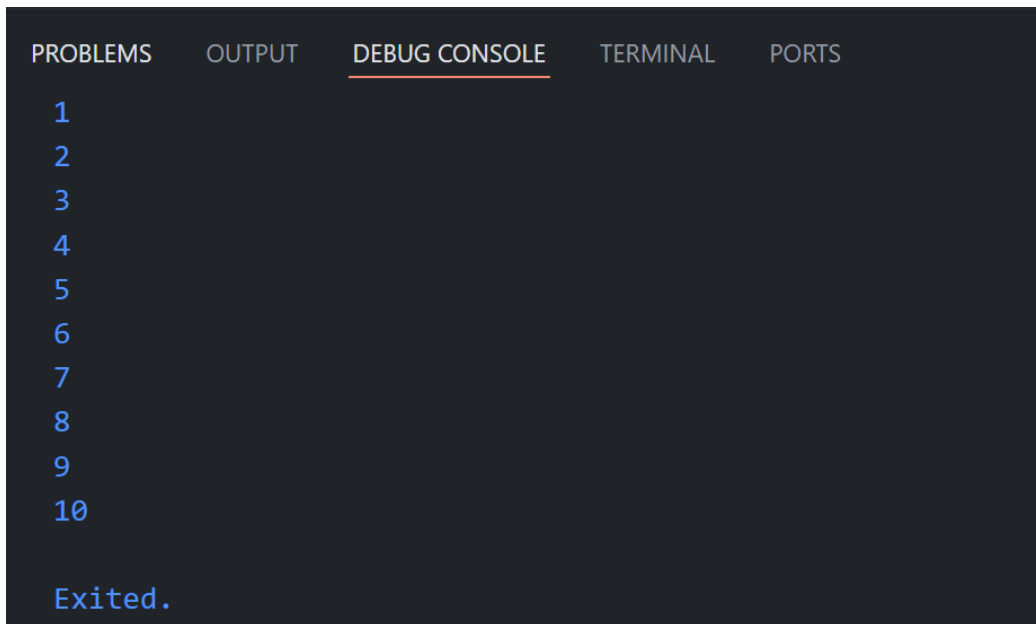
Kode tersebut adalah sebuah switch-case statement Dart yang digunakan untuk memeriksa nilai variabel `grade` yang diinisialisasi dengan `b`. Jika `grade` memiliki nilai `a`, maka program akan mencetak "Nilai Sangat Bagus" dan berhenti setelah kata kunci `break`. Jika `grade` bernilai `b`, program mencetak "Nilai Bagus", tetapi karena tidak ada `break`, eksekusi dilanjutkan ke `case 'c'`, sehingga program juga mencetak "Nilai Cukup" sebelum berhenti. Pada `case 'c'`, `break` digunakan untuk menghentikan eksekusi. Jika nilai `grade` tidak cocok dengan `case 'a'`, `case 'b'`, atau `case 'c'`, maka bagian `default` akan dijalankan yang mencetak "Nilai tidak tersedia". Untuk mencegah `case 'c'` dijalankan setelah `case 'b'`, tambahkan `break` setelah `print("Nilai Bagus")`.

## E. LOOPING FOR

### Sourcecode

```
void main() {  
    for (int i = 1; i ≤ 10; i++) {  
        print(i);  
    }  
}
```

## Screenshoot output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

1
2
3
4
5
6
7
8
9
10

Exited.
```

### Deskripsi program

Program Dart tersebut menggunakan sebuah for loop untuk mencetak angka dari 1 hingga 10. Loop dimulai dengan deklarasi variabel `i` yang diinisialisasi dengan nilai 1 (`int i = 1`), dan akan terus berjalan selama `i` kurang dari atau sama dengan 10 (`i <= 10`). Setelah setiap iterasi, nilai `i` akan bertambah 1 (`i++`). Pada setiap iterasi, pernyataan `print(i);` akan dijalankan untuk mencetak nilai `i` saat ini ke konsol. Program ini akan mencetak deretan angka 1, 2, 3, hingga 10 pada baris yang terpisah.

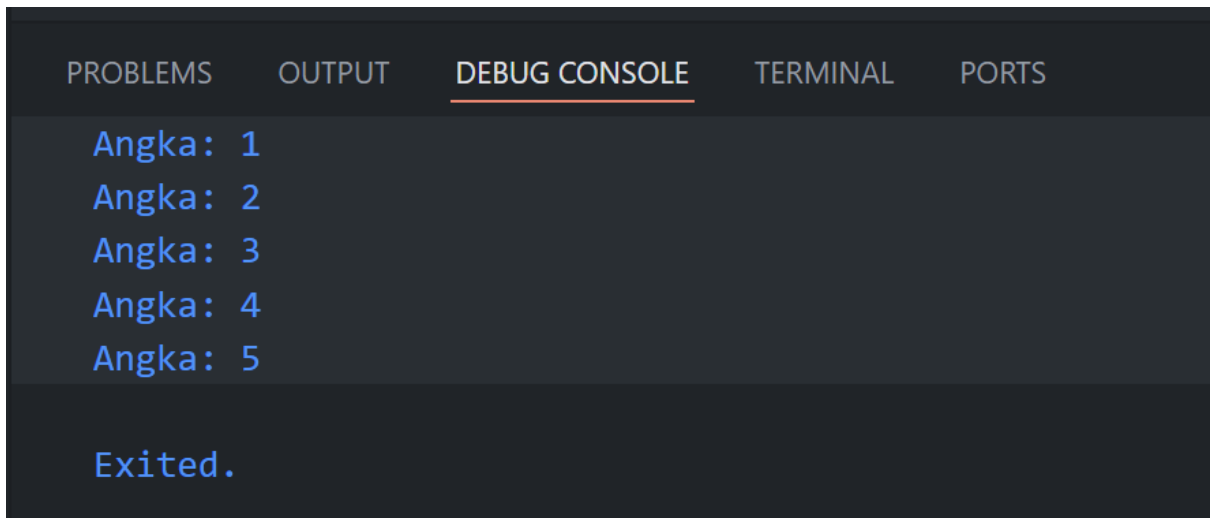
## F. LOOPING WHILE

### Sourcecode



```
void main() {
    int i = 1; // Inisialisasi
variabel
    // While loop untuk mencetak angka 1
sampai 5
    while (i <= 5) {
        print('Angka: $i');
        i++; // Increment untuk
menghindari loop tak berujung
    }
}
```

## Screenshoot output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Angka: 1
Angka: 2
Angka: 3
Angka: 4
Angka: 5

Exited.
```

## Deskripsi program

Kode Dart tersebut menggunakan while loop untuk mencetak angka dari 1 hingga 5. Program dimulai dengan menginisialisasi variabel `i` dengan nilai 1 (`int i = 1;`). \*While loop\* akan berjalan selama kondisi `i <= 5` terpenuhi. Di dalam loop, pernyataan `print('Angka: \$i');` digunakan untuk mencetak nilai `i` saat ini ke konsol, di mana `\$i` adalah interpolasi string yang menampilkan nilai `i`. Setelah mencetak, `i` akan ditambah 1 (`i++`) pada setiap iterasi untuk menghindari loop tak berujung. Ketika nilai `i` lebih dari 5, kondisi `i <= 5` akan menjadi salah, dan \*while loop\* berhenti, mengakhiri program setelah mencetak "Angka: 1" hingga "Angka: 5".

## G. FIXED LIST

### Sourcecode



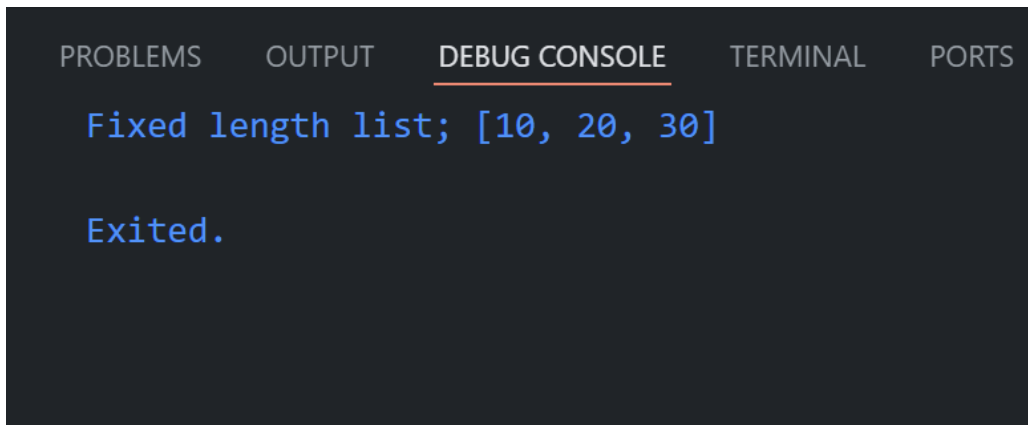
```
// Fixed List
void main() {
  List<int> fixedList = List.filled(3, 0);

  fixedList[0] = 10;
  fixedList[1] = 20;
  fixedList[2] = 30;

  print('Fixed length list; $fixedList');
}
```



## Screenshoot output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Fixed length list; [10, 20, 30]

Exited.
```

## Penjelasan program

Kode Dart tersebut membuat sebuah fixed-length list (daftar dengan panjang tetap) yang berisi elemen-elemen bertipe `int`. Program dimulai dengan mendeklarasikan variabel `fixedList` menggunakan `List.filled(3, 0)`, yang berarti membuat sebuah list dengan panjang 3 dan setiap elemennya diinisialisasi dengan nilai 0. Selanjutnya, elemen-elemen pada `fixedList` diubah nilainya secara langsung menggunakan indeks, di mana `fixedList[0]` diisi dengan nilai 10, `fixedList[1]` diisi dengan nilai 20, dan `fixedList[2]` diisi dengan nilai 30. Setelah semua elemen diubah, pernyataan `print('Fixed length list; \$fixedList');` digunakan untuk mencetak list tersebut ke konsol, sehingga output yang dihasilkan adalah `Fixed length list; [10, 20, 30]`.

## H. GROWABLE LIST

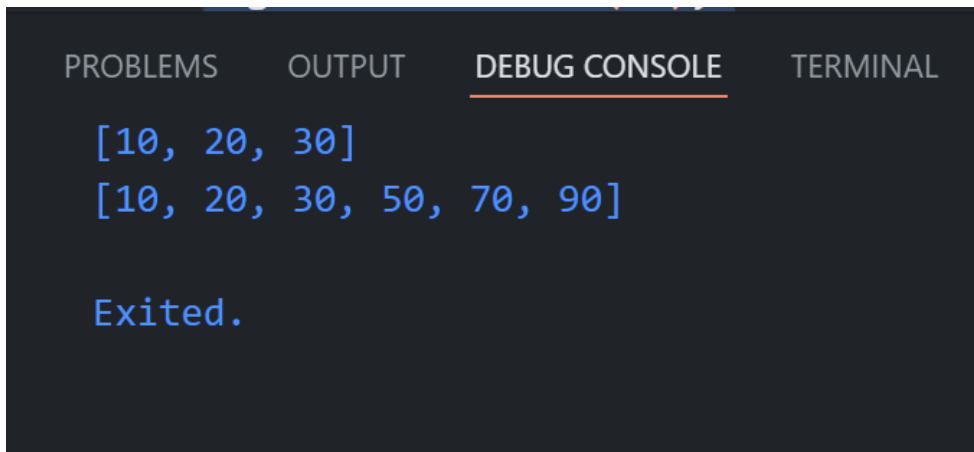
### Sourcecode



```
void main() {
  // Membuat growablelist (panjangnya bisa berubah)
  List<int> growableList = [];
  // Menambahkan elemen baru ke dalam list
  growableList.add(10);
  growableList.add(20);
  growableList.add(30);

  print(growableList);
  // Menambahkan lebih banyak elemen
  growableList.add(50);
  growableList.add(70);
  growableList.add(90);
  print(growableList); // Output: [10, 20, 30, 40, 50]
}
```

## Screenshoot output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

[10, 20, 30]
[10, 20, 30, 50, 70, 90]

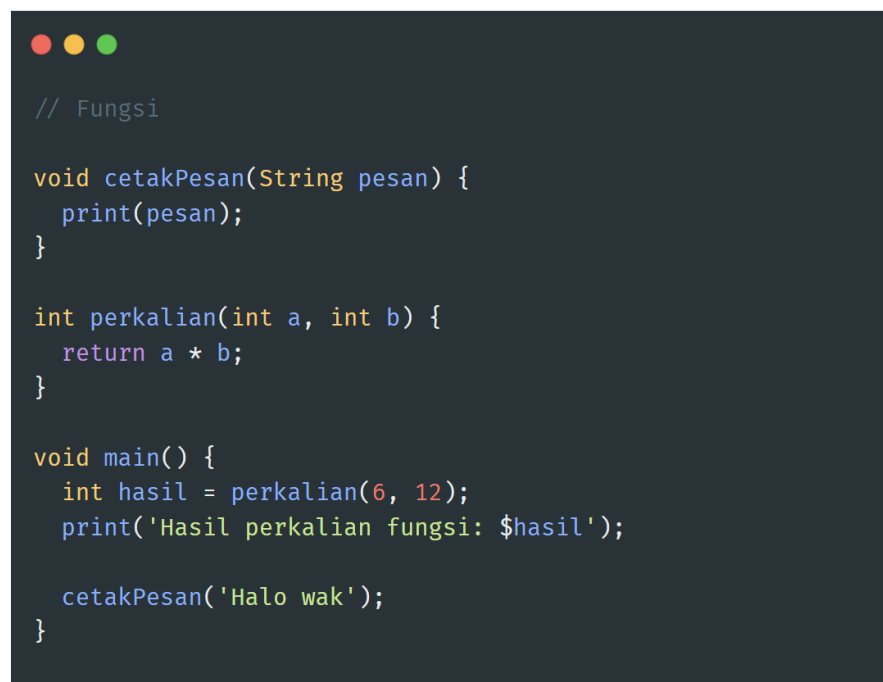
Exited.
```

## Deskripsi program

Kode Dart tersebut membuat sebuah growable list, yaitu daftar yang panjangnya bisa berubah secara dinamis, berisi elemen bertipe `int`. Pada awalnya, list `growableList` diinisialisasi sebagai list kosong (`[]`). Kemudian, beberapa elemen ditambahkan ke dalam list menggunakan metode `add()`, yaitu angka 10, 20, dan 30. Pernyataan `print(growableList);` akan mencetak isi list saat ini, yaitu `[10, 20, 30]`. Setelah itu, lebih banyak elemen ditambahkan lagi ke list dengan nilai 50, 70, dan 90, sehingga panjang list bertambah seiring penambahan elemen-elemen baru tersebut. Pernyataan `print(growableList);` kedua akan mencetak list yang sudah diperbarui, yaitu `[10, 20, 30, 50, 70, 90]`. Kode ini menunjukkan fleksibilitas list yang dapat berubah ukurannya di Dart. \*\*Catatan:\*\* Komentar di kode (`// Output: [10, 20, 30, 40, 50]`) kurang tepat, seharusnya output akhir adalah `[10, 20, 30, 50, 70, 90]`.

## I. FUNGSI

### Sourcecode



```
// Fungsi

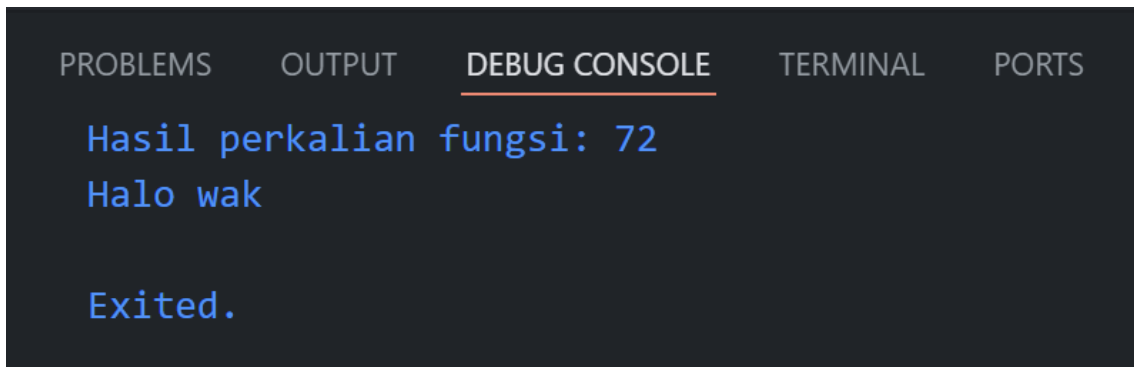
void cetakPesan(String pesan) {
    print(pesan);
}

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

void main() {
    int hasil = perkalian(6, 12);
    print('Hasil perkalian fungsi: $hasil');

    cetakPesan('Halo wak');
}
```

## Screenshoot output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Hasil perkalian fungsi: 72
Halo wak

Exited.
```

## Deskripsi program

Kode Dart tersebut mendefinisikan dua fungsi: `cetakPesan` dan `perkalian`. Fungsi `cetakPesan` menerima parameter bertipe `String` bernama `pesan` dan mencetak nilai dari parameter tersebut ke konsol menggunakan `print(pesan);`. Fungsi `perkalian` menerima dua parameter bertipe `int`, yaitu `a` dan `b`, dan mengembalikan hasil perkalian dari keduanya menggunakan `return a \* b;`. Pada fungsi utama (`main()`), `perkalian(6, 12)` dipanggil, sehingga hasil perkalian 6 dan 12 (yaitu 72) disimpan ke dalam variabel `hasil` dan dicetak dengan `print('Hasil perkalian fungsi: \$hasil');`. Setelah itu, fungsi `cetakPesan` dipanggil dengan argumen `Halo wak`, yang mencetak pesan tersebut ke konsol. Output program ini akan berupa: `Hasil perkalian fungsi: 72` diikuti oleh `Halo wak`.

## UNGUIDED

### 1. Tugas Percabangan (Branching )

#### Soal:

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

#### Deskripsi :

Jika nilai **lebih besar dari 70**, program harus mereturn **"Nilai A"**.

Jika nilai **lebih besar dari 40** tetapi **kurang atau sama dengan 70**, program harus mereturn **"Nilai B"**.

Jika nilai **lebih besar dari 0** tetapi **kurang atau sama dengan 40**, program harus mereturn **"Nilai C"**.

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 B

#### Sourcecode

```
import 'dart:io';

void main() {
  // Menggunakan stdout.write untuk menampilkan pesan input tanpa pindah baris
  stdout.write("Masukkan nilai (angka bulat): ");

  // Mengambil input dari pengguna
  String? input = stdin.readLineSync();

  // Memeriksa apakah input valid
  if (input == null || input.isEmpty || int.tryParse(input) == null) {
    print("Input tidak valid. Harap masukkan angka bulat.");
    return;
  }

  int nilai = int.parse(input);

  // Mengecek apakah nilai kurang dari atau sama dengan 40, jika ya maka nilai digandakan
  if (nilai <= 40 && nilai > 0) {
    nilai = nilai * 10; // Mengandakan nilai jika <= 40 dan > 0
  }

  // Memanggil fungsi untuk menentukan nilai setelah digandakan (jika <= 40)
  String hasil = tentukanNilai(nilai);

  if (hasil.isEmpty) {
    print("");
  } else {
    print("$nilai merupakan $hasil");
  }
}

String tentukanNilai(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 ""; // Kondisi else: Nilai tidak memenuhi kriteria A, B, atau C
  }
}
```

## Screenshoot output

```
PS D:\PPB_Dimas_Cahyo_Margono_2211104060\03_Pengenalan_Dart\UNGUIDED> dart percabangan.dart
Masukkan nilai (angka bulat): 80
80 merupakan Nilai A
PS D:\PPB_Dimas_Cahyo_Margono_2211104060\03_Pengenalan_Dart\UNGUIDED> dart percabangan.dart
Masukkan nilai (angka bulat): 4
40 merupakan Nilai C
PS D:\PPB_Dimas_Cahyo_Margono_2211104060\03_Pengenalan_Dart\UNGUIDED> █
```

## Deskripsi program

Program di atas merupakan program Dart yang meminta input berupa angka bulat dari pengguna, memvalidasi input, dan memberikan output yang sesuai berdasarkan kondisi tertentu. Program dimulai dengan meminta input dari pengguna melalui ``stdin.readLineSync()`` dan kemudian memeriksa apakah input valid, yaitu tidak kosong, bukan ``null``, dan dapat dikonversi menjadi angka bulat. Jika input tidak valid, program akan menampilkan pesan kesalahan dan berhenti. Jika input valid, nilai tersebut akan dikonversi menjadi tipe data ``int``. Selanjutnya, program memeriksa apakah nilai input lebih kecil atau sama dengan 40 dan lebih besar dari 0, jika ya maka nilai tersebut digandakan (dikalikan 10). Setelah itu, nilai akan dikirim ke fungsi ``tentukanNilai(int nilai)`` untuk dikategorikan sebagai "Nilai A" jika lebih besar dari 70, "Nilai B" jika di antara 41 hingga 70, atau "Nilai C" jika di antara 1 hingga 40. Jika nilai tidak memenuhi salah satu dari kondisi tersebut (misalnya, negatif atau nol), fungsi akan mengembalikan string kosong. Program kemudian memeriksa apakah hasilnya adalah string kosong, dan jika ya, tidak ada output yang dicetak; jika tidak, program menampilkan hasil dalam format seperti "50 merupakan Nilai B".

## 2. Tugas Looping (Perulangan)

### Soal:

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

### Contoh Output:



## Sourcecode

```
import 'dart:io';

void main() {
  // Meminta input tinggi piramida dari user dengan validasi
  print("Masukkan tinggi piramida (angka bulat positif): ");
  String? input = stdin.readLineSync();

  // Memeriksa apakah input valid
  if (input == null || input.isEmpty || int.tryParse(input) == null) {
    print("Input tidak valid. Harap masukkan angka bulat positif.");
    return;
  }

  int tinggi = int.parse(input);

  if (tinggi <= 0) {
    print(
      "Input tidak valid. Harap masukkan angka bulat positif yang lebih besar dari nol.");
    return;
  }

  // Membuat piramida bintang
  buatPiramidaBintang(tinggi);
}

void buatPiramidaBintang(int tinggi) {
  for (int i = 1; i <= tinggi; i++) {
    // Menambahkan spasi
    stdout.write(" " * (tinggi - i));
    // Menambahkan bintang
    stdout.write("*" * (2 * i - 1));
    // Pindah ke baris berikutnya
    print("");
  }
}
```

## Screenshoot output

```
PS D:\PPB_Dimas_Cahyo_Margono_2211104060\03_Pengenalan_Dart\UNGUIDED> dart looping.dart
Masukkan tinggi piramida (angka bulat positif):
6
  *
 ***
*****
*****
*****
*****
*****
PS D:\PPB_Dimas_Cahyo_Margono_2211104060\03_Pengenalan_Dart\UNGUIDED> |
```

## Deskripsi program

Kode di atas adalah program Dart yang membuat pola piramida bintang berdasarkan input tinggi dari pengguna. Program dimulai dengan meminta input tinggi piramida dari pengguna menggunakan `stdin.readLineSync()`. Kemudian program memvalidasi input tersebut, memastikan bahwa input tidak kosong, dapat dikonversi menjadi bilangan bulat, dan merupakan angka positif lebih besar dari nol. Jika input tidak memenuhi kriteria, program akan menampilkan pesan error dan berhenti. Setelah validasi berhasil, nilai input dikonversi menjadi tipe `int` dan diteruskan ke fungsi `buatPiramidaBintang()`. Fungsi `buatPiramidaBintang(int tinggi)` bertugas membuat piramida bintang menggunakan perulangan `for` yang mencetak pola bintang pada setiap baris. Pada setiap iterasi, program mencetak sejumlah spasi (" ") untuk membuat pola piramida yang rata di tengah, diikuti dengan sejumlah bintang ("\*") yang bertambah sebanyak 2 pada setiap barisnya. Setelah bintang dicetak, program memindahkan kursor ke baris baru menggunakan `print("")` sehingga piramida bintang terbentuk secara visual.

## 3. Tugas Function

### Soal:

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

### Sourcecode

```
import 'dart:io';

// Fungsi untuk mengecek apakah bilangan adalah bilangan prima
bool cekBilanganPrima(int angka) {
  if (angka <= 1) {
    return false; // Bilangan kurang dari atau sama dengan 1 bukan bilangan prima
  }
  for (int i = 2; i <= angka ~/ 2; i++) {
    if (angka % i == 0) {
      return false; // Jika ada pembagi selain 1 dan angka itu sendiri, bukan bilangan prima
    }
  }
  return true; // Jika tidak ditemukan pembagi, maka bilangan prima
}

void main() {
  // Meminta input dari user dengan validasi
  print("Masukkan bilangan bulat: ");
  String? input = stdin.readLineSync();

  // Memeriksa apakah input valid
  if (input == null || input.isEmpty || int.tryParse(input) == null) {
    print("Input tidak valid. Harap masukkan bilangan bulat.");
    return;
  }

  int angka = int.parse(input);

  // Memanggil fungsi untuk mengecek apakah bilangan prima
  if (cekBilanganPrima(angka)) {
    print("$angka adalah bilangan prima");
  } else {
    print("$angka bukan bilangan prima");
  }
}
```

## Screenshoot output

```
PS D:\PPB_Dimas_Cahyo_Margono_2211104060\03_Pengenalan_Dart\UNGUIDED> dart function.dart
Masukkan bilangan bulat:
1
1 bukan bilangan prima
PS D:\PPB_Dimas_Cahyo_Margono_2211104060\03_Pengenalan_Dart\UNGUIDED> dart function.dart
Masukkan bilangan bulat:
2
2 adalah bilangan prima
PS D:\PPB_Dimas_Cahyo_Margono_2211104060\03_Pengenalan_Dart\UNGUIDED> 
```

## Deskripsi program

Kode di atas merupakan program Dart yang digunakan untuk mengecek apakah sebuah bilangan merupakan bilangan prima atau bukan. Program dimulai dengan meminta input bilangan bulat dari pengguna menggunakan `stdin.readLineSync()`. Setelah menerima input, program memeriksa apakah input valid, yaitu input tidak boleh kosong, harus dapat dikonversi menjadi bilangan bulat, dan bukan `null`. Jika input tidak valid, program akan menampilkan pesan error dan berhenti. Setelah input dinyatakan valid, nilai input diubah menjadi tipe `int` dan disimpan dalam variabel `angka`. Kemudian, fungsi `cekBilanganPrima(int angka)` dipanggil untuk mengecek apakah bilangan tersebut merupakan bilangan prima. Fungsi ini bekerja dengan memeriksa apakah bilangan `angka` lebih besar dari 1 dan tidak memiliki pembagi selain 1 dan dirinya sendiri. Jika ditemukan pembagi selain 1 dan `angka`, maka fungsi akan mengembalikan `false`, menunjukkan bahwa `angka` bukan bilangan prima; sebaliknya, jika tidak ada pembagi yang ditemukan, fungsi akan mengembalikan `true`, menunjukkan bahwa `angka` adalah bilangan prima. Program kemudian menampilkan hasilnya di layar dengan pesan yang sesuai: `"$angka adalah bilangan prima"` jika hasilnya `true` atau `"$angka bukan bilangan prima"` jika hasilnya `false`.