

**LAPORAN TUGAS GUIDED
PEMROGRAMAN PERANGKAT BERGERAK**

**MODUL 3
PENGENALAN DART**



Disusun Oleh :
Fikri Khairul Fajri / 2211104052
SE-06-02

Asisten Praktikum :
Muhammad Faza Zulian Gesit Al Barru
Aisyah Hasna Aulia

Dosen Pengampu :
Yudha Islami Sulistya

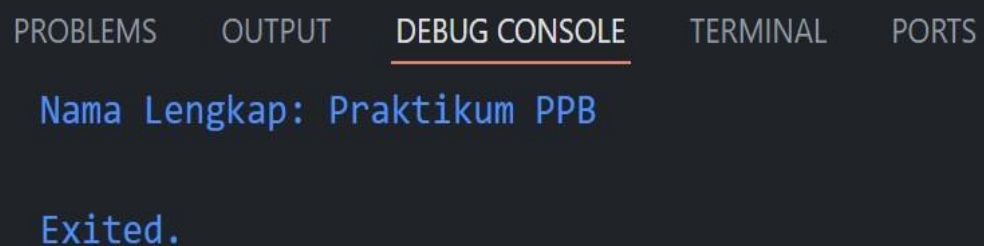
PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024
GUIDED DAN UNGUIDED

A. PENAMAAN VARIABEL DENGAN CAMELCASE

- Sourcecode

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

- Screenshoot Output



The screenshot shows the 'DEBUG CONSOLE' tab in the IDE. The output text is 'Nama Lengkap: Praktikum PPB' followed by 'Exited.' on a new line.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS  
  
Nama Lengkap: Praktikum PPB  
  
Exited.
```

- Deskripsi Program

Program sederhana dalam bahasa pemrograman Dart, yang digunakan untuk mendemonstrasikan cara penamaan variabel menggunakan konvensi **camelCase** serta menampilkan data menggunakan interpolasi string.

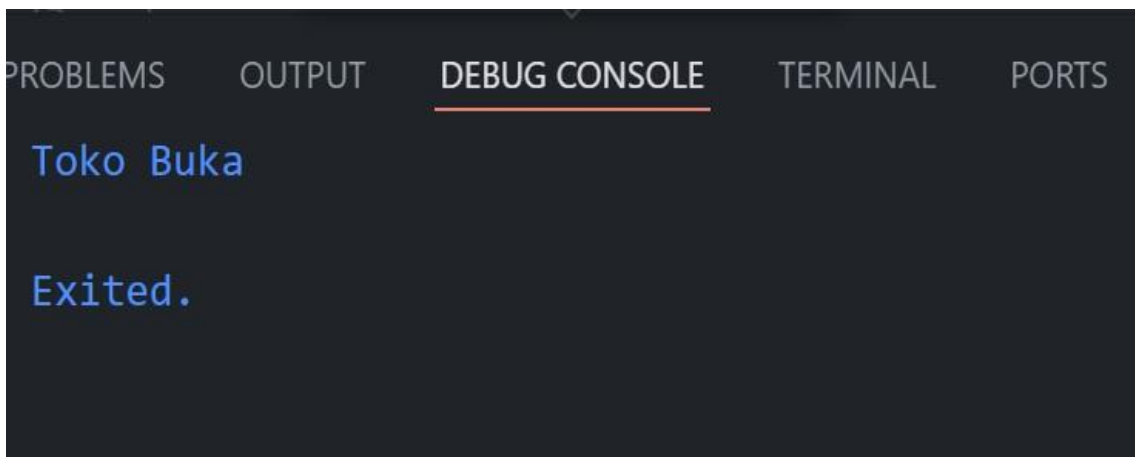
B. STATEMENT IF ELSE

Sourcecode

```
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



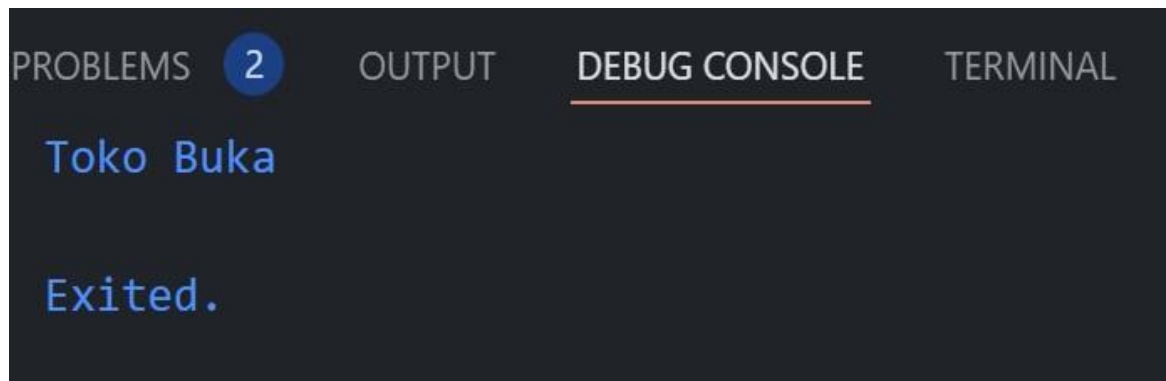
Deskripsi Program

Penggunaan **statement if-else** dan **operator ternary** dalam bahasa pemrograman Dart untuk menentukan status buka atau tutupnya sebuah toko berdasarkan waktu.

C. CONDITION TRUE FALSE Sourcecode

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

Screenshoot Output

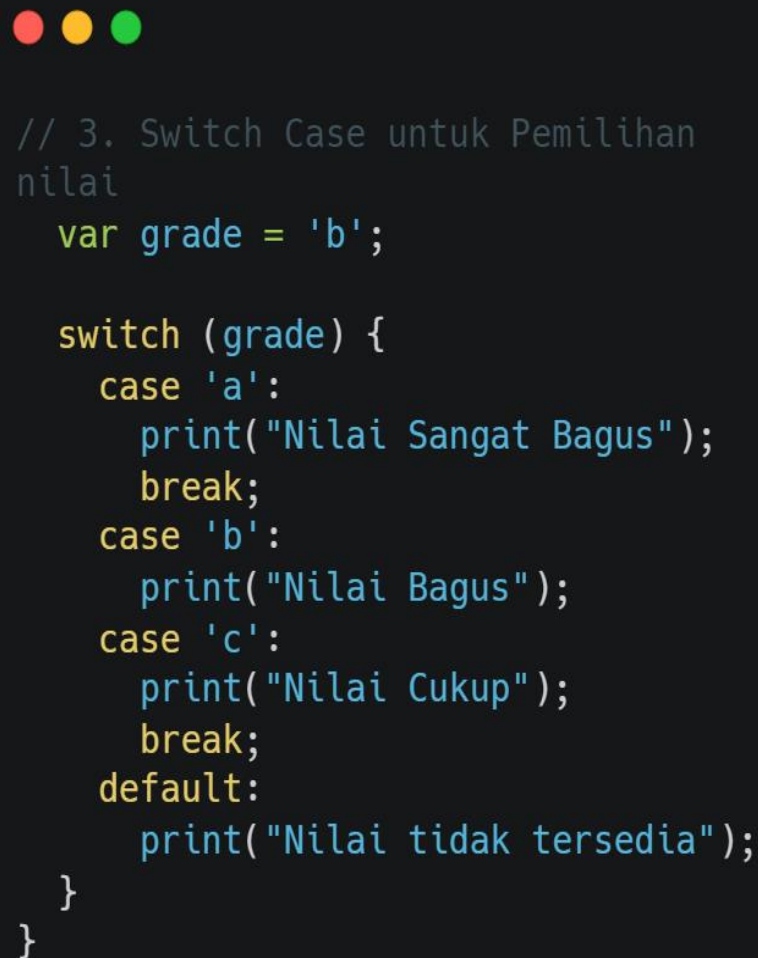


Deskripsi program

Penggunaan operator ternary dalam bahasa pemrograman Dart untuk menentukan status buka atau tutupnya sebuah toko berdasarkan waktu. Operator ternary adalah cara singkat untuk menulis pernyataan if-else sederhana.

D. SWITCH CASE

Sourcecode



```
// 3. Switch Case untuk Pemilihan
nilai
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

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

Nilai Bagus

Exited.

Deskripsi program

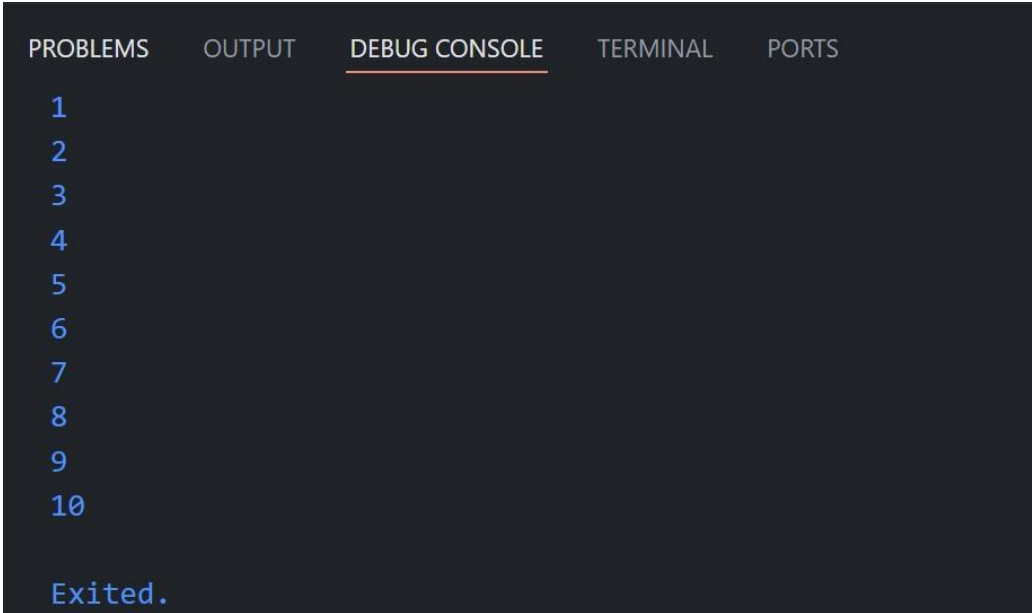
Penggunaan **switch-case statement** dalam bahasa pemrograman Dart untuk menampilkan nilai evaluasi berdasarkan variabel grade. Pernyataan switch-case digunakan untuk melakukan pemilihan berdasarkan beberapa kemungkinan nilai dari sebuah variabel.

E. LOOPING FOR Sourcecode

A screenshot of a code editor with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The code is written in Dart and uses syntax highlighting: keywords are in blue, integers are in red, and string literals are in green.

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

Screenshoot output

A screenshot of the output console in an IDE. The console has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE (which is selected and underlined), TERMINAL, and PORTS. The output shows the numbers 1 through 10 on separate lines, followed by the text "Exited." on the last line.

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
  
Exited.
```

Deskripsi program

penggunaan **looping for** dalam bahasa pemrograman Dart untuk mencetak angka 1 hingga 10.

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  
        beujung  
    }  
}
```

Screenshoot output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
  
Angka: 1  
Angka: 2  
Angka: 3  
Angka: 4  
Angka: 5  
  
Exited.
```

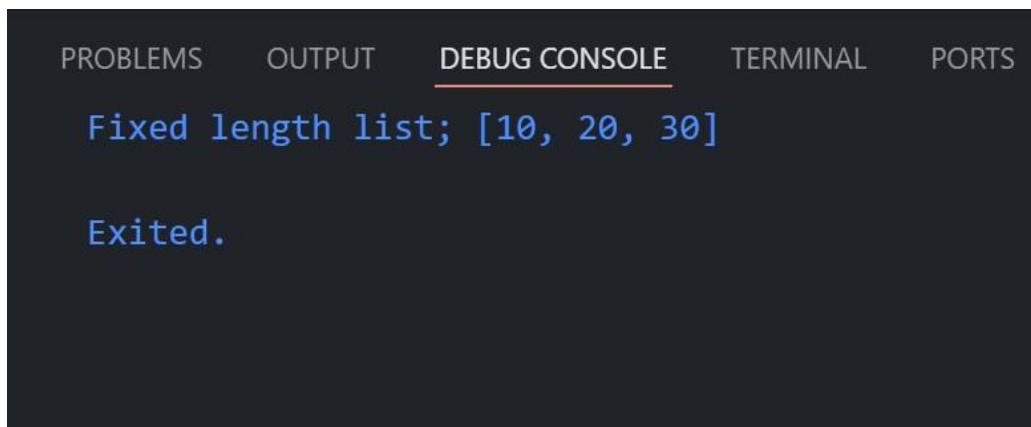
Deskripsi program

Penggunaan **looping while** dalam bahasa pemrograman Dart untuk mencetak angka dari 1 hingga 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

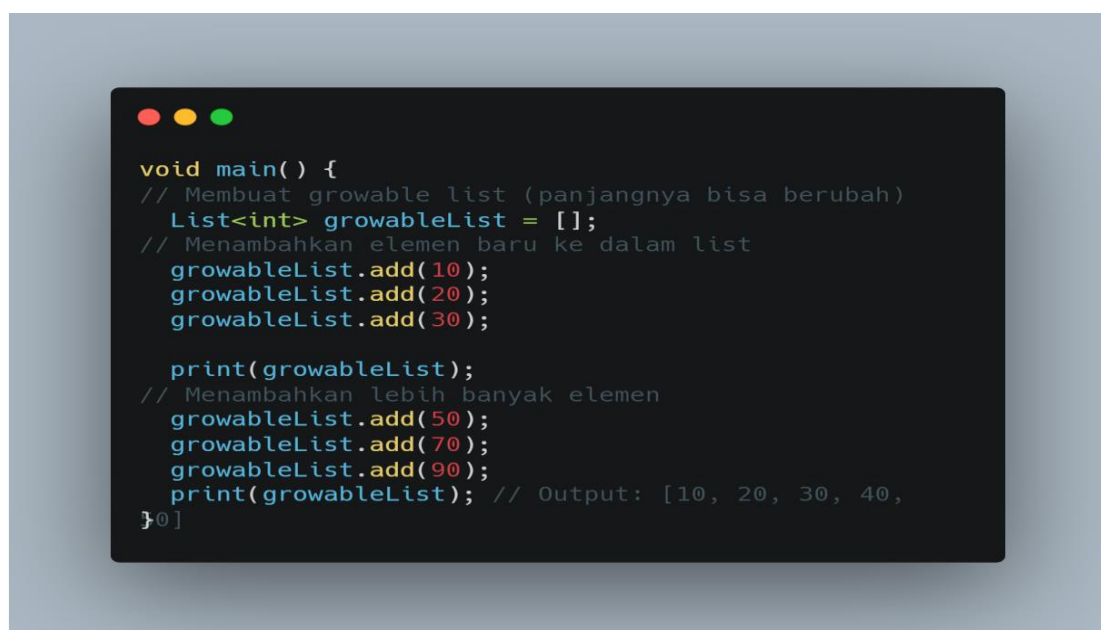
Fixed length list; [10, 20, 30]

Exited.
```

Penjelasan program

Program ini mendeklarasikan dan menggunakan **Fixed List** (List dengan panjang tetap) di Dart menggunakan `List.filled()`. Setelah list dibuat, elemen-elemen di dalamnya dapat diubah nilainya, tetapi panjang list tersebut tidak bisa ditambah atau dikurangi. Output dari program ini adalah tampilan list yang sudah diisi dengan nilai baru.

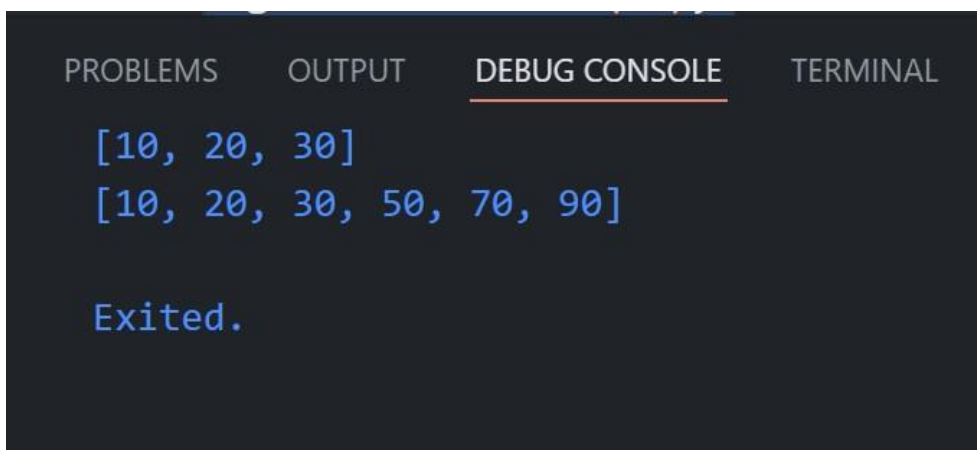
H. GROWABLE LIST



```
void main() {
  // 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(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

Program ini menggunakan **growable list** untuk menyimpan elemen-elemen bertipe integer (int). Metode add() digunakan untuk menambahkan elemen baru ke dalam list. Growable list dapat diubah ukurannya (dapat menambah atau menghapus elemen) saat program berjalan. Program mencetak isi list dua kali: pertama setelah menambahkan tiga elemen (10, 20, 30), dan kedua setelah menambahkan lebih banyak elemen (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

Deklarasi Fungsi:

- cetakPesan adalah fungsi yang mencetak pesan ke layar.
- perkalian adalah fungsi yang mengembalikan hasil perkalian dua bilangan.

Pemanggilan Fungsi di main():

- perkalian dipanggil dengan argumen 6 dan 12, menghasilkan 72.
- cetakPesan dipanggil dengan argumen "Halo wak", mencetak pesan tersebut ke layar.

Output Program:

- Program mencetak hasil perkalian dan pesan yang diberikan sebagai argumen fungsi.

B. UNGUIDED

a. Tugas Looping (Perulangan)

```
import 'dart:io';

void main() {
  // Meminta input dari user
  stdout.write("Masukkan tinggi piramida: ");
  int? n = int.tryParse(stdin.readLineSync()!);

  if (n != null && n > 0) {
    // Memastikan input valid dan tinggi piramida lebih dari 0
    // Membuat piramida
    for (int i = 1; i <= n; i++) {
      // Cetak spasi untuk membuat piramida terpusat
      for (int j = 1; j <= n - i; j++) {
        stdout.write(" ");
      }
      // Cetak bintang pada setiap baris
      for (int k = 1; k <= 2 * i - 1; k++) {
        stdout.write("*");
      }
      // Pindah ke baris berikutnya
      print("");
    }
  } else {
    print("Input tidak valid, masukkan angka lebih dari 0.");
  }
}
```

Output :

```
Masukkan tinggi piramida: 7
```

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

```
PS C:\Users\Fikri Khairul Fajri\unguided1\lib> |
```

- Penjelasan :

- a. Program menggunakan input pengguna untuk menentukan tinggi piramida.
- b. Memeriksa validitas input untuk memastikan bahwa pengguna memasukkan angka positif lebih dari 0.
- c. Menggunakan loop bersarang untuk mencetak spasi dan bintang, sehingga menghasilkan tampilan piramida yang rapi.
- d. Jika input tidak valid, program memberikan pesan error yang sesuai.

Program sederhana namun efektif dalam menunjukkan konsep penggunaan loop, input, dan output di bahasa Dart, serta bagaimana mengatur tata letak karakter pada konsol.

b. Tugas Percabangan (Branching)

```
import 'dart:io';

// Fungsi untuk menentukan nilai berdasarkan kondisi yang diberikan
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 ""; // Jika nilai tidak memenuhi semua kondisi di atas, return teks
               // kosong
  }
}

void main() {
  // Menerima input dari user
  stdout.write("Masukkan nilai: ");
  int? nilai = int.tryParse(stdin.readLineSync(!));

  // Memastikan input adalah angka dan tidak null
  if (nilai != null) {
    // Memanggil fungsi cekNilai untuk mendapatkan hasil
    String hasil = cekNilai(nilai);

    // Memeriksa apakah hasil tidak kosong
    if (hasil.isNotEmpty) {
      print("$nilai merupakan $hasil");
    } else {
      print("Nilai tidak valid.");
    }
  } else {
    print("Input tidak valid, harap masukkan angka.");
  }
}
```

Output :

```
PS C:\Users\Fikri Khairul Fajri\unguided1\lib> dart Percabangan.dart
Masukkan nilai: 85
85 merupakan Nilai A
PS C:\Users\Fikri Khairul Fajri\unguided1\lib> 
```

Penjelasan :

Program ini digunakan untuk mengevaluasi nilai yang dimasukkan pengguna dan mengelompokkan hasilnya ke dalam kategori "Nilai A", "Nilai B", atau "Nilai C" berdasarkan rentang nilai yang diberikan. Program ini juga menangani kondisi kesalahan seperti input tidak valid (bukan angka) dan menampilkan pesan kesalahan yang sesuai jika kondisi tersebut terjadi.

Menunjukkan pemahaman tentang konsep fungsi, pengkondisian if-else, validasi input, serta cara membaca dan menampilkan data di konsol menggunakan bahasa Dart.

c. Tugas Function

```
import 'dart:io';

// Fungsi untuk mengecek apakah suatu bilangan adalah bilangan prima
bool cekPrima(int n) {
  if (n <= 1) {
    return false; // Bilangan <= 1 bukan bilangan prima
  }

  for (int i = 2; i <= n ~/ 2; i++) {
    if (n % i == 0) {
      return false; // Jika ada pembagi selain 1 dan n, maka bukan bilangan prima
    }
  }
  return true; // Jika tidak ada pembagi, maka bilangan prima
}

void main() {
  // Menerima input dari user
  stdout.write("Masukkan bilangan bulat: ");
  int? angka = int.tryParse(stdin.readLineSync()!);

  // Memeriksa apakah input valid
  if (angka != null) {
    // Memanggil fungsi cekPrima untuk menentukan apakah bilangan prima atau bukan
    if (cekPrima(angka)) {
      print("$angka merupakan bilangan prima");
    } else {
      print("$angka bukan bilangan prima");
    }
  } else {
    print("Input tidak valid, harap masukkan bilangan bulat.");
  }
}
```

Output :

```
PS C:\Users\Fikri Khairul Fajri\unguided1\lib> dart Fungsi.dart  
Masukkan bilangan bulat: 5  
5 merupakan bilangan prima  
PS C:\Users\Fikri Khairul Fajri\unguided1\lib> |
```

Penjelasan :

Program ini berfungsi untuk menentukan apakah sebuah bilangan bulat yang dimasukkan pengguna adalah bilangan prima atau bukan. Program memanfaatkan fungsi cekPrima untuk mengecek sifat bilangan prima menggunakan loop dan operasi modulus. Selain itu, program juga memvalidasi input pengguna agar hanya menerima bilangan bulat yang valid, serta menampilkan pesan kesalahan jika input tidak memenuhi kriteria.

Program ini menunjukkan pemahaman dasar tentang konsep fungsi, perulangan (for loop), dan penggunaan if-else untuk validasi kondisi serta pengelolaan input/output di konsol menggunakan bahasa Dart.