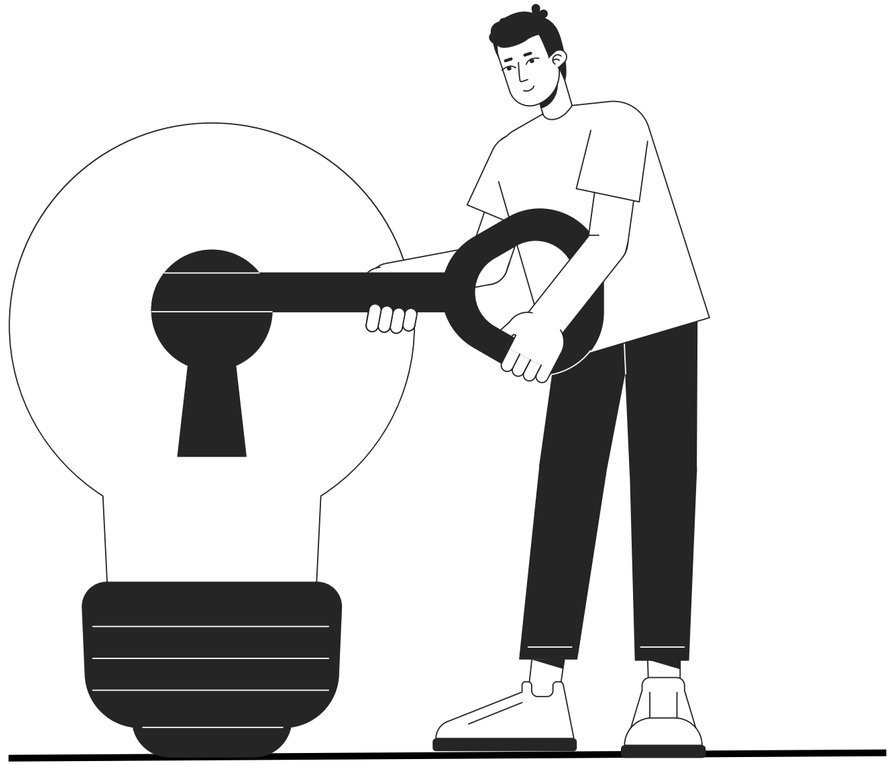
DART FUNDAMENTAL

dibuat oleh **Andrea Surya Habibie**

Tujuan Belajar

1. Memahami Konsep Dasar Dart
2. Menggunakan Komentar dalam Kode
3. Mengelola Variabel dan Konstanta
4. Memahami dan Menggunakan Tipe Data
5. Mengolah Nilai Angka dan String
6. Menggunakan Operator dengan Tepat
7. Menangani Kesalahan (Exception Handling)
8. Menerapkan Fungsi dalam Pemrograman
9. Menerapkan Null Safety



Comments

Comment adalah catatan atau penjelasan dalam kode yang tidak dijalankan oleh program.

Menjelaskan maksud baris kode
Menonaktifkan kode sementara (debugging)
Membantu tim memahami logika

- Komentar digunakan untuk dokumentasi kode.
- Single-line: // komentar
- Multi-line: /* komentar */
- Penting untuk kolaborasi tim dan penjelasan kode kompleks.

Comments

Contoh:

```
void main() {
  // Menyimpan nama pengguna
  String nama = "Yajid";

  // Menyimpan usia
  int usia = 25;

  // Mencetak biodata ke konsol
  print("Nama: $nama");
  print("Usia: $usia tahun");

  // TODO: Tambahkan kolom hobi nanti
}
```

Comments

Contoh:

```
void main() {
  // Cetak kalimat ke konsol
  print("Halo Dunia!");

  /*
    Ini adalah komentar multiline.
    Cocok untuk penjelasan panjang atau
    untuk mematikan blok kode sementara.
  */

  // TODO: Tambahkan validasi nama pengguna nanti
}
```

Variabel dalam Dart

Variabel adalah wadah untuk menyimpan data seperti angka, teks, atau boolean agar bisa digunakan berulang kali di program.

```
void main() {
  // Deklarasi variabel dengan tipe data
  String nama = "Dart";
  int usia = 10;
  double tinggi = 1.75;
  bool aktif = true;

  // Menampilkan nilai variabel
  print("Nama: $nama");
  print("Usia: $usia tahun");
  print("Tinggi: $tinggi meter");
  print("Status aktif: $aktif");
}
```

Menyimpan dan mengelola nilai
Mengubah data tanpa mengedit seluruh kode
Membuat program dinamis

- Variabel menyimpan data.
- Penulisan: var, dynamic, atau langsung tipe data seperti int, String
- Contoh: var name = "Ali";

Constants & Final

final dan const adalah variabel tetap (tidak bisa diubah) setelah diberi nilai.

```
void main() {
  final nama = "Aisyah";  // Bisa dari hasil kalkulasi runtime
  const umur = 25;        // Harus sudah diketahui sebelumnya

  print("Nama: $nama");
  print("Umur: $umur");

  // final dengan DateTime
  final waktuSekarang = DateTime.now();
  print("Waktu: $waktuSekarang");

  // const tidak bisa pakai nilai runtime
  // const waktuError = DateTime.now(); ❌ Ini akan error
}
```

Menjaga data tetap konsisten
Meningkatkan performa
Mencegah perubahan yang tidak disengaja

- Gunakan final jika nilai diketahui saat runtime
- Gunakan const jika nilai diketahui saat compile time

Constants & Final

final dan const adalah variabel tetap (tidak bisa diubah) setelah diberi nilai.

```
void main() {
  final nama = "Aisyah";  // Bisa dari hasil kalkulasi runtime
  const umur = 25;        // Harus sudah diketahui sebelumnya

  print("Nama: $nama");
  print("Umur: $umur");

  // final dengan DateTime
  final waktuSekarang = DateTime.now();
  print("Waktu: $waktuSekarang");

  // const tidak bisa pakai nilai runtime
  // const waktuError = DateTime.now(); ❌ Ini akan error
}
```

Menjaga data tetap konsisten
Meningkatkan performa
Mencegah perubahan yang tidak disengaja

- Gunakan final jika nilai diketahui saat runtime
- Gunakan const jika nilai diketahui saat compile time

Tipe data

Tipe data adalah jenis nilai yang bisa disimpan dalam variabel.

Tipe data membantu komputer memahami:
Bagaimana data disimpan
Apa yang bisa dilakukan dengan data itu (misalnya dijumlahkan atau dipotong)

- Menyimpan nilai ke variabel
- Melakukan operasi (penjumlahan, penggabungan string, dll)
- Validasi input pengguna

Semua programmer Dart selalu menggunakan tipe data, baik eksplisit maupun implisit.

STRING

- String digunakan untuk menyimpan teks.
- Dapat dideklarasikan dengan tanda kutip tunggal (' ') atau ganda (" ")

Contoh:

```
String pesan = "Hello, Dart!";
print(pesan); // Output: Hello, Dart!
```

INTEGER

- Integer digunakan untuk menyimpan bilangan bulat (tanpa koma).
- Integer sering digunakan untuk perhitungan matematika dan perulangan.

Contoh:

```
int umur = 26;
print(umur); // Output: ......

atau

int umur = 25;
print("Umur saya: $umur tahun");
```

DOUBLE

- Double digunakan untuk menyimpan bilangan desimal.
- Double sering digunakan untuk perhitungan yang melibatkan pecahan.

Contoh:

```
double pi = 3.14;
print(pi); // Output: 3.14

atau

double tinggi = 1.75;
print("Tinggi saya: $tinggi meter");
```

BOOLEAN

- Boolean hanya memiliki dua nilai: `true` atau `false`
- Digunakan dalam pengambilan keputusan (if-else).

Contoh:

```
bool isPPKDJakpus = true;
print(isPPKDJakpus); // Output: true

atau

bool isAktif = true;
print("Status aktif: $isAktif");
```

STUDY CASE

```
void main() {
 String name = "Ali";
 int age = 25;
 double height = 175.5;
 bool isStudent = false;
 print("Nama: $name, Umur: $age, Tinggi: $height, Mahasiswa: $isStudent");
}
```

JAWABAN :

LIST

- Tipe data untuk menyimpan kumpulan nilai berurutan (seperti array).
- Bisa juga List dari int, double, atau campuran dengan dynamic.

Contoh:

```
List<String> hobi = ["Baca", "Ngoding", "Menulis"];
print("Hobi saya: ${hobi[0]}");
```

Map

- Tipe data untuk menyimpan pasangan key-value
- Bisa juga List dari int, double, atau campuran dengan dynamic.

Contoh:

```
Map<String, dynamic> biodata = {
  "nama": "Budi",
  "umur": 30,
  "aktif": true,
};

print("Nama: ${biodata['nama']}");
print("Umur: ${biodata['umur']}");
```

Dynamic

- Tipe data fleksibel yang bisa menyimpan nilai apa saja, dan bisa berubah tipe saat runtime.

Contoh:

```
dynamic data = "Halo";
print(data);

data = 123;  // Sekarang jadi int
print(data);
```

var (Inference)

- Bukan tipe data, tapi cara untuk biarkan Dart menebak tipe data secara otomatis dari nilai pertama.

Contoh:

```
var kota = "Jakarta";  // dianggap sebagai String
var angka = 10;        // dianggap sebagai int
```

Ringkasan Tipe Data

Tipe Data	Contoh Nilai	Kegunaan
int	5, -10, 1000	Bilangan bulat
double	3.14, -0.5	Bilangan desimal
String	"Halo"	Teks atau karakter
bool	true, false	Logika benar atau salah
List	[1, 2, 3]	Kumpulan data terurut
Map	{'nama': 'Ali'}	Kumpulan key-value
dynamic	"Hello", 123	Nilai fleksibel, bisa berubah tipe
var	Otomatis dari nilai	Biarkan Dart menebak tipe

Study Case

Sebuah klub buku digital sedang mengembangkan sistem sederhana untuk menyimpan biodata anggotanya.Anda diminta untuk membuat program berbasis Dart yang menyimpan dan menampilkan data anggota, lengkap dengan dokumentasi menggunakan comment.

Instruksi Tugas:

1. Buat variabel untuk menyimpan informasi anggota berikut:

Nama (String)
Umur (int)
Tinggi badan (double)
Status aktif (bool)
Daftar buku favorit (List)

https://bit.ly/tugas1dart

2. Informasi tambahan disimpan dalam bentuk Map (contoh: alamat, profesi)

Gunakan:

// untuk komentar satu baris
/* ... */ untuk penjelasan multi-baris
/// untuk komentar dokumentasi di atas fungsi

Buat fungsi tampilkanBiodata() yang menampilkan seluruh informasi anggota.

Numbers & Arithmetic Operations

- Dart mendukung operasi matematika dasar
  Penjumlahan (+), Pengurangan (-), Perkalian (*), Pembagian (/), Modulus (%)

Operator	**Nama**	**Contoh (**a = 10, b = 3**)**
+	Penjumlahan	a + b = 13
-	Pengurangan	a - b = 7
*	Perkalian	a * b = 30
/	Pembagian	a / b = 3.33
~/	Bagi bulat	a ~/ b = 3
%	Sisa bagi	a % b = 1

Ringkasan Aritmatika

Contoh :

- int a = 10;
- int b = 5;
- print(a + b); // Output: 15
-

atau

```
void main() {
  int a = 10, b = 3;
  print(a + b);  // 13
  print(a * b);  // 30
  print(a % b);  // 1
}
```

Relational / Comparison Operators

Relational ini Membandingkan dua nilai.

Operator	Fungsi	Contoh
‘==	Sama dengan	5 == 5
!=	Tidak sama	5 != 4
>	Lebih besar	6 > 3
<	Lebih kecil	2 < 5
>=	Lebih besar/sama	5 >= 5
<=	Lebih kecil/sama	4 <= 6

Logical Operators (Operator Logika)

Relational ini Digunakan untuk ekspresi kondisi.

Operator	Nama	Contoh
&&	AND	true && false → false
||	OR	
!	NOT	!true → false

&& (AND): Hanya true jika kedua kondisi bernilai true

|| (OR): true jika salah satu kondisi bernilai true

! (NOT): Membalik kondisi, true jadi false, dan sebaliknya

Functions

Function (fungsi) adalah blok kode yang bisa kita beri nama dan panggil kapan pun kita butuh, tanpa perlu menulis ulang kode yang sama.

Fungsi sangat membantu untuk:

Menghindari pengulangan kode (DRY: Don't Repeat Yourself)

Membuat kode lebih rapi dan mudah dibaca

Memecah program besar menjadi bagian-bagian kecil

Struktur Dasar Fungsi di Dart

```
void namaFungsi() {
  // isi kode di sini
}
```

Contoh :

```
void sapaPengguna() {
  print("Halo! Selamat datang.");
}
```

```
void main() {
  sapaPengguna(); // Memanggil fungsi
}
```

Functions

Fungsi dalam Parameter

```
void sapa(String nama) {
  print("Halo, $nama!");
}

void main() {
  sapa("Aisyah");
  sapa("Budi");
}
```

Functions

Fungsi dengan Return Value

```
int tambah(int a, int b) {
  return a + b;
}

void main() {
  int hasil = tambah(5, 3);
  print("Hasil: $hasil");
}
```

Functions

```dart
int tambah(int a, int b) {
  return a + b;
}

int kurang(int a, int b) {
  return a - b;
}

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

double bagi(int a, int b) {
  return a / b;
}

void main() {
  int x = 10;
  int y = 2;

  print("Tambah: ${tambah(x, y)}");
  print("Kurang: ${kurang(x, y)}");
  print("Kali: ${kali(x, y)}");
  print("Bagi: ${bagi(x, y)}");
}
```

```
int tambah(int a, int b) {
  return a + b;
}

int kurang(int a, int b) {
  return a - b;
}

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

double bagi(int a, int b) {
  return a / b;
}

void main() {
  int x = 10;
  int y = 2;

print("Tambah: ${tambah(x, y)}");
  print("Kurang: ${kurang(x, y)}");
  print("Kali: ${kali(x, y)}");
  print("Bagi: ${bagi(x, y)}");
}
```

Tambah: 12

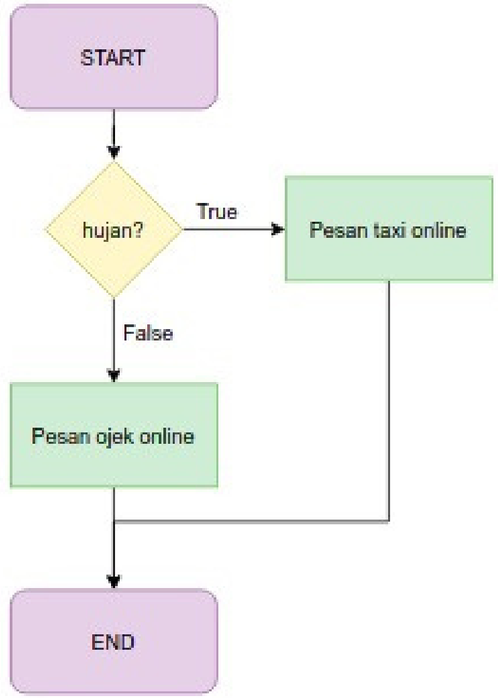
Kurang: 8

Kali: 20

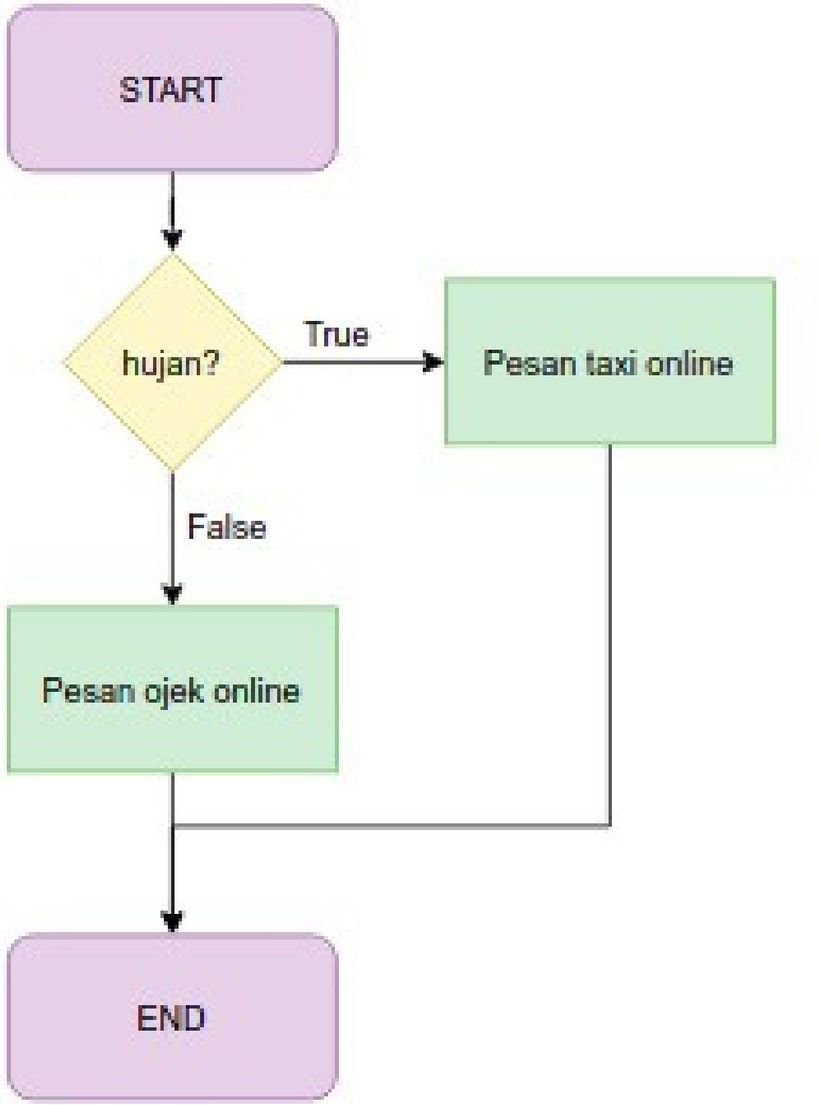
Bagi: 5

Control Flow

- Setiap hari kita melakukan perhitungan dan perbandingan guna membuat keputusan, apapun itu. Contohnya apakah perlu mencuci kendaraan ketika cuaca sedang cerah? Apa saja transportasi online yang bisa dipesan ketika hujan untuk sampai di tempat tujuan?



Control Flow



If-else (Conditional)

- if dan else adalah cara untuk membuat keputusan dalam program. Komputer akan menjalankan kode tertentu jika kondisi yang kita buat bernilai benar (true), dan melewati atau menjalankan yang lain jika tidak terpenuhi.

Contoh :

```
bool hujan = true;

if (hujan) {
  print("Pakai jas hujan.");
} else {
  print("Pakai jaket biasa.");
}
```

If-else (Conditional)

- Struktur sederhana dari if else

```
if (kondisi) {
  // kode jika kondisi benar
} else {
  // kode jika kondisi salah
}
```

If-else (Conditional)

- Contoh

```
void main() {
  int usia = 35;

  if (usia < 12) {
    print("Kategori: Anak-anak");
  } else if (usia < 18) {
    print("Kategori: Remaja");
  } else if (usia < 60) {
    print("Kategori: Dewasa");
  } else {
    print("Kategori: Lansia");
  }
}
```

Exceptions

- Error dapat ditangani menggunakan try-catch.

Contoh :

```
 try {
int result = 10 ~/ 0;
} catch (e) {
print("Terjadi kesalahan: $e");
}
```

Latihan Mandiri

Tugas 2 Dart Fundamental

https://bit.ly/tugas2dart

Dart Input

- Fungsi stdout.write()

```
stdout.write("Masukkan gaji Anda: ");
```

• Digunakan untuk menampilkan teks ke layar tanpa berpindah ke baris baru.
• Berbeda dengan print(), yang otomatis menambahkan baris baru setelah teks.

Dart Readline

- Fungsi stdin.readLineSync()

stdin.readLineSync()

- Digunakan untuk membaca input dari pengguna.
- Secara default, data yang dibaca berupa String.
- Karena kita membutuhkan Integer, maka perlu dikonversi.

int.parse(stdin.readLineSync()!)

- Mengubah input dari String menjadi Integer.
- ! digunakan untuk menandakan bahwa nilai tidak akan null (Non-null assertion).
- Jika input bukan angka, program akan error.

Contoh

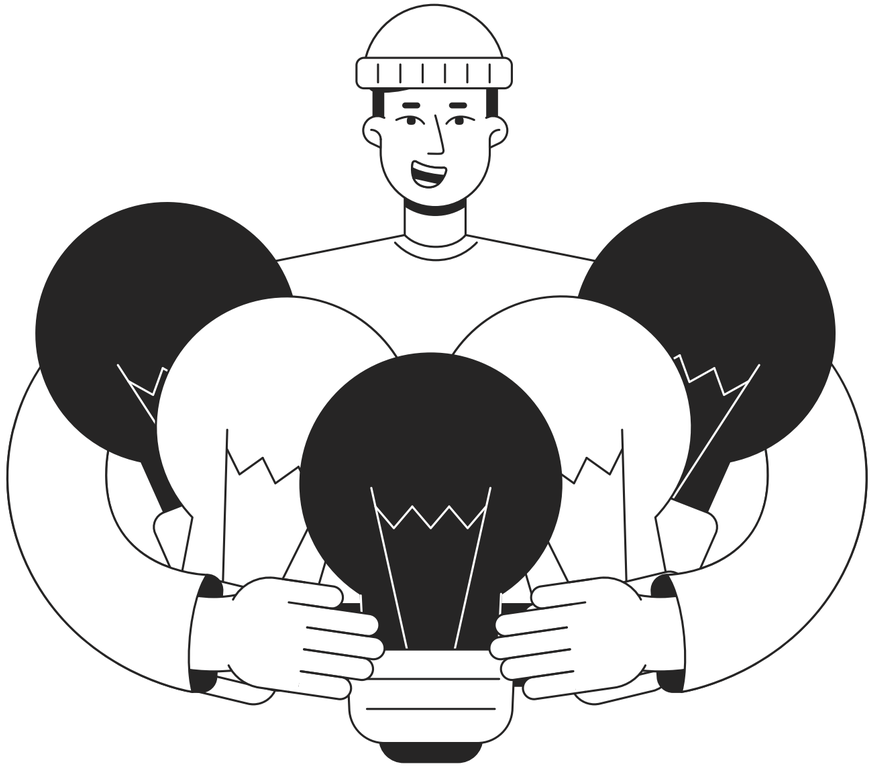
```
import 'dart:io';

void main() {
  // Meminta input gaji dari pengguna
  stdout.write("Masukkan gaji Anda: ");
  String? inputGaji = stdin.readLineSync(); // Membaca input sebagai String

  // Mengubah string input menjadi angka (int)
  int gaji = int.parse(inputGaji ?? "0");

  // Menentukan pajak 10% dari gaji
  double pajak = gaji * 0.1;

  // Menampilkan hasil
  print("Gaji Anda: Rp $gaji");
  print("Pajak (10%): Rp $pajak");
}
```



Setiap aplikasi dimulai dari satu baris kode. Kuasai Dart, dan buka peluang tak terbatas dalam pengembangan Flutter!

Terima Kasih

Semoga bermanfaat