**Apa itu Constructor?**

* **Constructor** adalah **method khusus** di dalam class yang otomatis dipanggil saat kita membuat **objek baru**.
* Fungsinya untuk **menginisialisasi nilai properti** atau melakukan setup awal.
* Nama constructor **sama dengan nama class** dan **tidak punya return type**.

📌 Analogi:

Blueprint (class) = Desain rumah  
Constructor = Tukang bangun rumah sesuai desain  
Object = Rumah jadi

**🔥 Jenis-Jenis Constructor di Dart**

**1. Default Constructor**

Jika tidak kita tulis sendiri, Dart otomatis membuat constructor kosong.

class Laptop {

String? brand;

int? ram;

}

void main() {

var laptop = Laptop(); // pakai constructor default

laptop.brand = "Asus";

laptop.ram = 8;

print("${laptop.brand}, ${laptop.ram}GB");

}

**2. Constructor Biasa (Dengan Parameter)**

Bisa langsung isi nilai properti saat bikin objek.

class Laptop {

String brand;

int ram;

// Constructor dengan parameter

Laptop(this.brand, this.ram);

}

void main() {

var laptop = Laptop("Lenovo", 16);

print("${laptop.brand}, ${laptop.ram}GB");

}

**3. Named Constructor**

Bikin beberapa cara berbeda untuk bikin objek.

class Laptop {

String brand;

int ram;

// Constructor biasa

Laptop(this.brand, this.ram);

// Named constructor

Laptop.basic() : brand = "Generic", ram = 4;

}

void main() {

var laptop1 = Laptop("HP", 8);

var laptop2 = Laptop.basic(); // pakai named constructor

print("${laptop1.brand}, ${laptop1.ram}GB"); // HP, 8GB

print("${laptop2.brand}, ${laptop2.ram}GB"); // Generic, 4GB

}

**4. Redirecting Constructor**

Constructor yang memanggil constructor lain untuk menghindari duplikasi kode.

class Laptop {

String brand;

int ram;

Laptop(this.brand, this.ram);

// Redirecting ke constructor utama

Laptop.minSpec() : this("Default", 4);

}

void main() {

var laptop = Laptop.minSpec();

print("${laptop.brand}, ${laptop.ram}GB");

}

**5. Constant Constructor (Immutable Object)**

Dipakai untuk membuat objek **immutable** (tidak bisa diubah).

class Laptop {

final String brand;

final int ram;

// Const constructor

const Laptop(this.brand, this.ram);

}

void main() {

const laptop1 = Laptop("Dell", 16);

const laptop2 = Laptop("Dell", 16);

print(identical(laptop1, laptop2)); // true (pakai memori yang sama)

}

**🔥 Ringkasan Jenis Constructor**

| **Jenis Constructor** | **Kegunaan** | **Contoh Pemakaian** |
| --- | --- | --- |
| **Default** | Disediakan otomatis jika tidak kita buat | var obj = ClassName(); |
| **Parameterized** | Mengisi properti langsung saat membuat objek | ClassName(param1, param2); |
| **Named** | Memberi cara lain untuk membuat objek | ClassName.named(); |
| **Redirecting** | Constructor memanggil constructor lain | : this(...) |
| **Const** | Membuat objek immutable, efisien memori | const ClassName(...); |

**Diagram: Konsep Constructor di Dart**

┌─────────────────────────────────────┐

│ CLASS │

│ Blueprint / cetak biru objek │

└─────────────────────────────────────┘

│

▼

┌─────────────────────────┐

│ CONSTRUCTOR │

│ - Method khusus │

│ - Nama sama dgn class │

│ - Tidak ada return │

│ - Inisialisasi objek │

└─────────────────────────┘

│

┌────────────────┼────────────────┐

▼ ▼ ▼

┌────────────────┐ ┌─────────────────┐ ┌─────────────────┐

│ Default │ │ Parameterized │ │ Named │

│ - Otomatis ada │ │ - Pakai argumen │ │ - Nama berbeda │

│ - Kosong │ │ saat buat obj │ │ │

└────────────────┘ └─────────────────┘ └─────────────────┘

│

▼

┌────────────────┼────────────────┐

▼ ▼

┌───────────────────────┐ ┌───────────────────────┐

│ Redirecting │ │ Const │

│ - Panggil ctor lain │ │ - Objek immutable │

└───────────────────────┘ └───────────────────────┘

│

▼

┌─────────────────────────────────────┐

│ OBJECT │

│ Objek nyata hasil instance class │

└─────────────────────────────────────┘

**📝 Penjelasan Diagram**

1. **Class**: Blueprint atau cetak biru.
2. **Constructor**: Jembatan dari class ke object.
3. **Jenis Constructor**:
   * **Default** → Otomatis ada, kosong.
   * **Parameterized** → Isi properti lewat argumen.
   * **Named** → Cara alternatif buat objek.
   * **Redirecting** → Constructor memanggil constructor lain.
   * **Const** → Objek immutable, hemat memori.
4. **Object**: Hasil nyata dari class.