

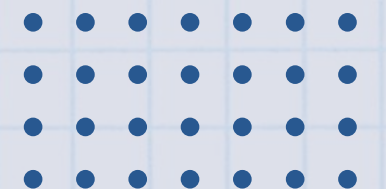
perbedaan Null Safety dan Late variabel

Disusun oleh :
Kelompok 6

Anggota Kelompok

Eddo Dava Alfarisi

(2241720232)



Fanesabirawaning S

(2241720027)

Lintang Aprill S.

(2241720231)

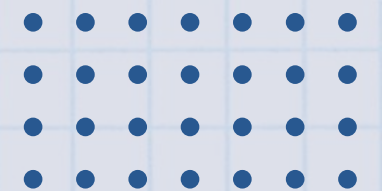
Wahyudi

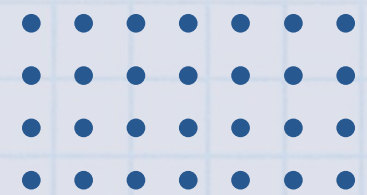
(2241720018)



Pengantar

- 01.** Dalam pengembangan aplikasi menggunakan Dart, dua konsep penting yang perlu dipahami adalah Null Safety dan Late Variable.
- 02.** Null Safety memastikan bahwa variabel tidak mengandung nilai null kecuali secara eksplisit diizinkan, untuk mencegah error runtime.
- 03.** Late Variable digunakan saat kita ingin mendeklarasikan variabel yang belum diinisialisasi pada saat deklarasi, namun akan diinisialisasi sebelum digunakan.
- 04.** Pada slide ini, kita akan menjelaskan perbedaan antara kedua konsep tersebut, serta melihat contoh kode eksekusi untuk masing-masing.





Null safety



- Berfungsi untuk menghindari error karena nilai null pada sebuah variable. dengan null safety, kita dapat menentukan apakah sebuah variable boleh null atau tidak.
- Null Safety memastikan bahwa variabel tidak dapat mengandung nilai null kecuali secara eksplisit diizinkan.
- Dalam Dart, variabel yang dapat memiliki nilai null harus ditandai dengan tanda tanya ?.

```
int? age; // age bisa bernilai null
```

- Saat Null Safety diaktifkan, kode menjadi lebih aman dari error NullPointerException.



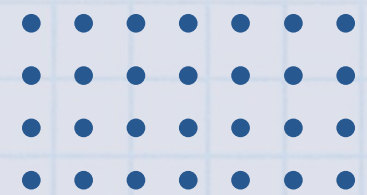
○ Contoh kode Null Safety

Contoh Source Code untuk mencari tahu panjang nama sesuai inputan :

```
1 void main() {  
2   String? name; //bertipe nullable String?  
3  
4   name = "Fanesa";  
5   print(name); // Print Fanesa  
6  
7   if (name != null) { //Mengecek apakah name tidak null sebelum mengakses length  
8     print("Panjang Nama: ${name.length}"); //Mencetak nama sesuai inputan  
9   }  
10 }
```

Output :

```
Fanesa  
Panjang Nama: 6
```

Late Variable



- Variabel dengan kata kunci late ditandai untuk diinisialisasi di lain waktu, namun hanya boleh digunakan setelah diinisialisasi.
- late digunakan saat variabel tidak dapat langsung diinisialisasi tetapi kita yakin variabel tersebut akan diinisialisasi sebelum digunakan.

```
late String name;
```

- Jika late variabel diakses sebelum diinisialisasi, akan menghasilkan error.



○ Contoh kode Late Variable

Berikut adalah contoh kode late variable

```
1 void main() {  
2   late String lateDescription; // Deklarasi variabel late  
3  
4   // Inisialisasi variabel late sebelum digunakan  
5   lateDescription = "Ini adalah contoh late variable.";  
6  
7   // Sekarang kita bisa mengaksesnya  
8   print(lateDescription);  
9  
10  // Contoh inisialisasi lazy (tertunda) pada fungsi berat  
11  late String result = heavyComputation();  
12  print(result);  
13 }  
14  
15 // Fungsi yang memerlukan waktu lama untuk eksekusi  
16 String heavyComputation() {  
17   print("Melakukan perhitungan berat...");  
18   return "Hasil perhitungan berat";  
19 }  
20
```

Output

```
Ini adalah contoh late variable.  
Melakukan perhitungan berat...  
Hasil perhitungan berat
```

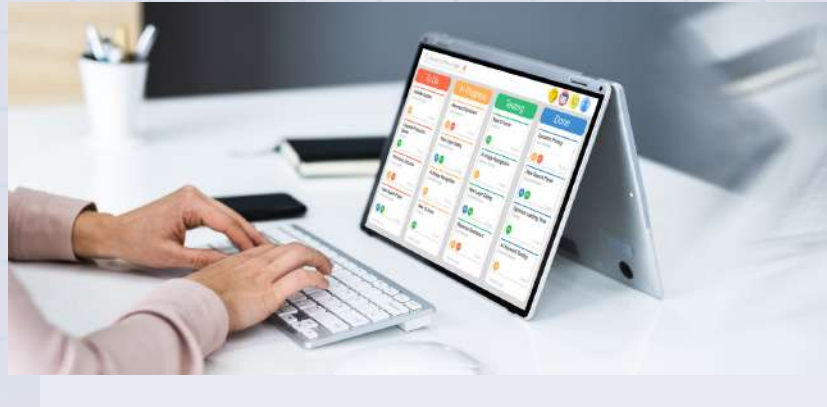
+ +
+ +
+ +
+ +
+ +

Tabel Perbandingan



Fitur	Null Safety	Late Variable
Inisialisasi	Harus bernilai null atau diberi nilai	Diinisialisasi sebelum digunakan
Penggunaan	Untuk menangani variabel yang nullable	Untuk variabel yang inisialisasinya ditunda
Keuntungan	Mengurangi bug null	Mendukung lazy initialization
Kekurangan	Harus cek null secara eksplisit	Error runtime jika tidak diinisialisas





○ Kesimpulan



Null Safety mencegah variabel menyimpan nilai null secara sembarangan, memastikan program lebih aman dari error yang sulit dilacak.

Late Variable memungkinkan variabel diinisialisasi di kemudian waktu tanpa langsung memberikan nilai awal.



Terima Kasih

Kelompok 6