

PROPERTI LIST(ARRAY) DART

1. **first**

Mengembalikan elemen pertama dalam List.

```
void main() {  
  List lst = [];  
  lst.add(12);  
  lst.add(13);  
  print("Elemen pertama pada list adalah: ${lst.first}");  
}
```

2. **isEmpty**

Mengembalikan nilai true jika List tidak memiliki elemen

```
void main() {  
  List lst = [];  
  lst.add(12);  
  lst.add(13);  
  print(lst.isEmpty);  
}
```

3. **isNotEmpty**

Mengembalikan nilai true jika List memiliki setidaknya satu elemen.

```
void main() {  
  List lst = [];  
  lst.add(12);  
  lst.add(13);  
  print(lst.isNotEmpty);  
}
```

4. **length**

Mengembalikan ukuran dari List.

```
void main() {  
  List lst = [];  
  lst.add(12);  
  lst.add(13);  
  print("Panjang dari list adalah: ${lst.length}");  
}
```

5. **last**

Mengembalikan elemen terakhir dalam List.

```
void main() {  
  List lst = [];  
  lst.add(12);  
  lst.add(13);  
  print("Elemen terakhir pada list adalah: ${lst.last}");  
}
```

6. **PROPERTI REVERSED**

Properti ini mengembalikan Object iterable yang berisi nilai List dalam urutan element terbalik.

```
void main() {  
  List lst = [];  
  lst.add(12);  
  lst.add(13);  
  print("List nilai dalam urutan terbalik: ${lst.reversed}");  
}
```

7. Single

Properti ini memeriksa apakah List hanya memiliki satu elemen dan mengembalikan nilai elemen tersebut

```
void main() {
  List lst = [];
  lst.add(12);
  print("List hanya memiliki satu elemen: ${lst.single}");
}
```

FUNGSI PADA LIST

FUNGSI PADA LIST

Dalam bagian ini akan membahas cara melakukan beberapa operasi dasar pada List dengan memakai fungsi-fungsi yang sudah tersedia, seperti tercantum di bawah ini :

No	Pengoperasian & Deskripsi Dasar
1	Memasukkan Elemen ke dalam List List dapat diubah dan dapat tumbuh secara dinamis saat runtime. Fungsi List.add() berfungsi untuk menambahkan suatu nilai ke dalam suatu List dan mengembalikan object List yang dimodifikasi.
2	Memperbarui List List di Dart dapat diperbarui dengan cara : <ul style="list-style-type: none">• Memperbarui Indeks• Menggunakan fungsi List.replaceRange()
3	Menghapus elemen pada List Beberapa fungsi didukung oleh class List di library dart:core dan dapat digunakan untuk menghapus elemen dalam suatu List.

MENAMBAH – MEMASUKAN ELEMEN PADA LIST #

- **Fungsi** List.add()

Fungsi List.add() **digunakan** untuk menambahkan nilai tertentu ke akhir List dan **mengembalikan objek List** yang telah dimodifikasi

```
void main() {
  List l = [1,2,3];
  l.add(12);
  print(l);
}
```

- **List.addAll()**

dipakai untuk menerima banyak nilai yang akan ditambahkan, nilai-nilai tersebut dipisahkan dengan koma dan semua nilai akan ditambahkan ke dalam List

```
void main() {
  List l = [1,2,3];
  l.addAll([12,13]);
  print(l);
}
```

Dart mendukung penambahan elemen pada posisi tertentu dalam List dengan menggunakan fungsi **insert()**. Fungsi ini menerima nilai dan menyisipkannya pada indeks yang ditentukan. Selain itu, fungsi **insertAll()** juga dapat digunakan untuk menyisipkan banyak nilai yang diberikan, dimulai dari indeks yang ditentukan. Berikut adalah cara penulisan dari fungsi insert dan insertAll:

List.**insert**(index,value)

List.**insertAll**(index, iterable_list_of _values)

Contoh berikut mengilustrasikan penggunaan fungsi **insert()** dan **insertAll()** secara berurutan.

Contoh : List.insert()

```
void main() {  
    List l = [1,2,3];  
    l.insert(0,4);  
    print(l);  
}
```

Contoh : List.insertAll()

```
void main() {  
    List l = [1,2,3];  
    l.insertAll(0,[120,130]);  
    print(l);  
}
```

MEMPERBARUI ATAU UPDATE LIST

MEMPERBARUI INDEKS

Dart memungkinkan untuk memodifikasi nilai element dalam List. Dengan kata lain, seseorang dapat menimpa nilai elemen pada suatu List dengan nilai yang baru. Contohnya seperti berikut ini:

```
void main() {  
    List l = [1, 2, 3,];  
    l[0] = 123;  
    print (l);  
}
```

MENGGUNAKAN FUNGSI LIST.REPLACERANGE()

Class List dari library dart:core menyediakan fungsi **replaceRange()** untuk mengubah elemen dalam List. Fungsi ini menggantikan nilai elemen dalam rentang yang ditentukan.

Cara penulisan fungsi List.replaceRange() adalah sebagai berikut :

```
List.replaceRange(int start_index,int end_index,Iterable <items>)
```

Di mana,

- **Start_index** – bilangan bulat yang mewakili posisi indeks untuk memulai penggantian.
- **End_index** – bilangan bulat yang mewakili posisi indeks untuk berhenti mengganti.
- **<items>** – Object iterable yang mewakili nilai-nilai baru.

```
void main() {  
List l = [1, 2, 3,4,5,6,7,8,9];  
print('Nilai list sebelum diganti ${1}');  
  
l.replaceRange(0,3,[11,23,24]);  
print('Nilai list setelah mengganti item di antara rentang [0-3] adalah ${1}');  
}
```

MENGHAPUS ELEMEN PADA LIST

MENGGUNAKAN LIST.REMOVE()

Fungsi List.remove() menghapus suatu elemen Object yang pertama ditemukan dari suatu List. Fungsi ini mengembalikan nilai true jika nilai yang ditentukan berhasil dihapus dari List.

Cara Penulisan

```
List.remove(Object value)
```

```
void main() {  
List l = [1, 2, 3,4,5,6,7,8,9];  
print('Nilai list sebelum menghapus elemen dalam list ${1}');  
bool res = l.remove(1);  
print('Nilai list setelah menghapus elemen dalam list ${1}');  
}
```

MENGGUNAKAN LIST.REMOVEAT()

Fungsi List.removeAt menghapus elemen pada indeks yang ditentukan dan mengembalikan elemen tersebut.

Cara Penulisan

```
List.removeAt(int index)
```

```
void main() {  
List l = [1, 2, 3,4,5,6,7,8,9];  
print('Nilai list sebelum menghapus elemen dalam list ${1}');  
dynamic res = l.removeAt(1);  
print('Nilai elemen ${res}');  
print('Nilai list setelah menghapus elemen dalam list ${1}');  
}
```

MENGGUNAKAN LIST.REMOVELAST()

Fungsi List.remove() menghapus elemen terakhir pada suatu List.

```
List.removeLast()
```

Contoh

```
void main() {  
    List l = [1, 2, 3,4,5,6,7,8,9];  
    print('Nilai list sebelum menghapus elemen dalam list ${l}');  
    dynamic res = l.removeLast();  
    print('Nilai item yang muncul ${res}');  
    print('Nilai list setelah menghapus elemen dalam list ${l}');  
}
```

MENGGUNAKAN LIST.REMOVE RANGE()

Fungsi List.removeRange() digunakan untuk menghapus elemen dalam rentang yang ditentukan.

```
List.removeRange(int start, int end)
```

Di mana,

- **start**– mewakili posisi awal untuk menghapus item.
- **end**– mewakili posisi dalam daftar untuk berhenti menghapus item.

```
void main() {  
    List l = [1, 2, 3,4,5,6,7,8,9];  
    print('Nilai list sebelum menghapus elemen dalam list ${l}');  
    l.removeRange(0,3);  
    print('Nilai list setelah menghapus elemen pada list di antara rentang 0-3 ${l}');  
}
```