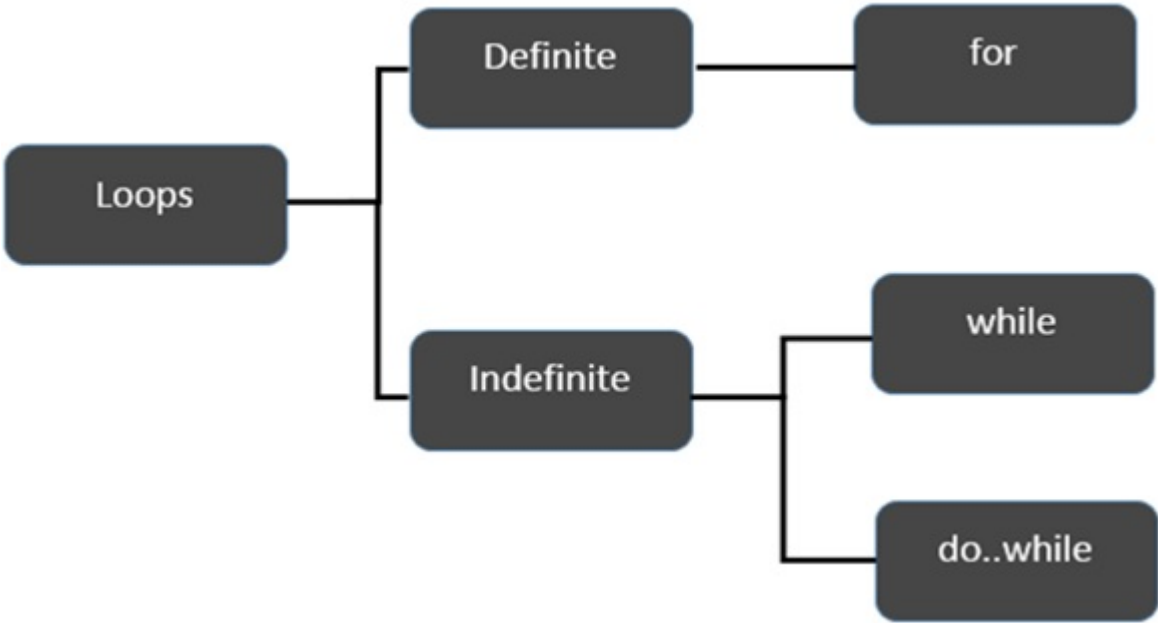


## Pemrograman Dart - Loops

Terkadang, instruksi tertentu membutuhkan eksekusi berulang. Loop adalah cara ideal untuk melakukan hal yang sama. Perulangan mewakili sekumpulan instruksi yang harus diulang. Dalam konteks perulangan, pengulangan disebut sebagai **iterasi** .

Gambar berikut mengilustrasikan klasifikasi loop -



Mari kita mulai diskusi dengan Definite Loops. Perulangan yang jumlah iterasinya pasti / tetap disebut sebagai a **loop yang pasti** .

Sr Tidak	Loop & Deskripsi
1	<div>untuk loop</div> <div>The <b>untuk</b> loop implementasi dari sebuah loop yang pasti. Perulangan for mengeksekusi blok kode untuk beberapa kali. Ini dapat digunakan untuk mengulangi serangkaian nilai tetap, seperti array</div>
2	<div>untuk... di Loop</div> <div>Perulangan for ... in digunakan untuk melakukan perulangan melalui properti objek.</div>

Selanjutnya, sekarang mari kita bahas loop tak terbatas. Perulangan tak tentu digunakan jika jumlah iterasi dalam satu perulangan tidak pasti atau tidak diketahui. Loop tak terbatas dapat diimplementasikan menggunakan -

Sr Tidak	Loop & Deskripsi
1	sementara Loop  Perulangan while mengeksekusi instruksi setiap kali kondisi yang ditentukan bernilai true. Dengan kata lain, loop mengevaluasi kondisi sebelum blok kode dieksekusi.
2	lakukan... saat Loop  Perulangan do... while mirip dengan perulangan while kecuali bahwa perulangan do ... while tidak mengevaluasi kondisi untuk pertama kalinya perulangan dijalankan.

Sekarang mari kita lanjutkan dan bahas **Pernyataan Kontrol Loop** Dart.

Sr Tidak	Pernyataan & Deskripsi Kontrol
1	Pernyataan break  The <b>istirahat</b> pernyataan digunakan untuk mengambil kontrol dari konstruk a. Menggunakan <b>break</b> in a loop menyebabkan program keluar dari loop. Berikut adalah contoh pernyataan <b>break</b> .
2	Lanjutkan Pernyataan  The <b>melanjutkan</b> pernyataan melompat laporan berikutnya pada iterasi saat ini dan mengambil kontrol kembali ke awal loop.

Menggunakan Label untuk Mengontrol Aliran

Sebuah **label** hanyalah sebuah identifier yang diikuti dengan titik dua (:) yang diterapkan pada pernyataan atau blok kode. Label dapat digunakan dengan **istirahat** dan **terus** mengontrol aliran dengan lebih tepat.

Jeda baris tidak diperbolehkan antara pernyataan '**lanjutkan**' atau '**putus**' dan nama labelnya. Juga, tidak boleh ada pernyataan lain di antara nama label dan loop terkait.

Contoh: Label dengan Break

Demo Langsung

```
void main() {
  outerloop: // This is the label name

  for (var i = 0; i < 5; i++) {
    print("Innerloop: ${i}");
    innerloop:

    for (var j = 0; j < 5; j++) {
      if (j > 3 ) break ;

      // Quit the innermost loop
      if (i == 2) break innerloop;

      // Do the same thing
      if (i == 4) break outerloop;

      // Quit the outer loop
```

```

        print("Innerloop: ${j}");
    }
}
}

```

**Output** berikut ditampilkan pada eksekusi yang berhasil dari kode di atas.

```

Innerloop: 0
Innerloop: 0
Innerloop: 1
Innerloop: 2
Innerloop: 3
Innerloop: 1
Innerloop: 0
Innerloop: 1
Innerloop: 2
Innerloop: 3
Innerloop: 2
Innerloop: 3
Innerloop: 0
Innerloop: 1
Innerloop: 2
Innerloop: 3
Innerloop: 4

```

**Contoh: Label dengan lanjutkan**

Demo Langsung

```

void main() {
    outerloop: // This is the label name

    for (var i = 0; i < 3; i++) {
        print("Outerloop:${i}");

        for (var j = 0; j < 5; j++) {
            if (j == 3){
                continue outerloop;
            }
            print("Innerloop:${j}");
        }
    }
}

```

Output berikut ditampilkan pada eksekusi yang berhasil dari kode di atas.

```

Outerloop: 0
Innerloop: 0
Innerloop: 1
Innerloop: 2

Outerloop: 1
Innerloop: 0
Innerloop: 1
Innerloop: 2

Outerloop: 2
Innerloop: 0
Innerloop: 1

```

