

Nama : Muhammad Fadhil Syahputra

NIM : 2211014210002

Asisten : Alvaro Putra Prasetya

## Percobaan 2 : Plotting dan Looping

Dalam MATLAB, penggunaan perintah perulangan (looping) sangat penting untuk menjalankan rangkaian kode yang terdiri dari lebih dari satu baris. Terdapat dua jenis utama perulangan, yaitu **for loop** dan **while loop**, yang berbeda dalam cara pengendaliannya.

**While loop** akan terus mengeksekusi perintah selama kondisi tertentu tetap bernilai benar (true), sedangkan **for loop** digunakan ketika jumlah iterasi sudah diketahui sebelumnya—yakni berdasarkan rentang nilai tertentu.

Perulangan `for` di MATLAB merupakan konsep dasar yang digunakan untuk mengeksekusi sekumpulan perintah sesuai jumlah elemen dalam suatu vektor. Struktur dasarnya sebagai berikut:

- Perulangan dimulai dengan kata kunci `for`, diikuti oleh sebuah variabel yang berjalan mengikuti elemen-elemen vektor tertentu. Sebagai contoh, vektor `k` dapat didefinisikan dari nilai `f` hingga `t` dengan interval `s`.
- Selanjutnya adalah kumpulan perintah (command group) yang dijalankan sebanyak elemen dalam vektor tersebut.
- Perulangan ditutup dengan `end`, yang menandakan akhir dari blok perintah. Setelah loop selesai, variabel `k` akan menyimpan nilai terakhir dari iterasi.

Dalam percobaan ini, digunakan tiga metode perulangan yaitu: **while loop**, **for loop**, dan **nested loop**.

- 

**While loop** berfungsi mengulang perintah selama kondisi yang diberikan masih benar. Perulangan ini berhenti jika kondisi tersebut tidak lagi terpenuhi.

- 
- 

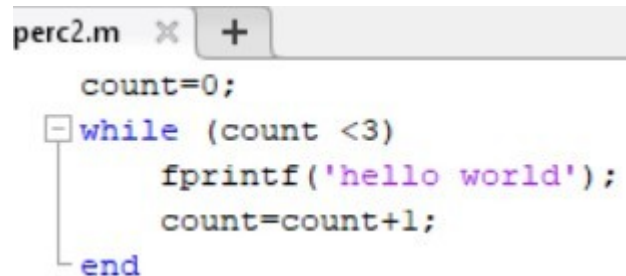
**For loop** digunakan ketika jumlah pengulangan sudah pasti, ditentukan oleh nilai awal, akhir, dan kenaikan (increment).

- 
- 

**Nested loop** adalah perulangan bersarang, artinya loop di dalam loop lainnya. Ini bisa berupa `for` dalam `for`, `while` dalam `while`, atau kombinasi keduanya. Teknik ini digunakan saat diperlukan proses berlapis, misalnya ketika memproses data matriks.

-

Contoh penggunaan while loop di MATLAB adalah menampilkan teks "hello world" sebanyak tiga kali. Meskipun kondisi perulangannya adalah `count < 3`, perulangan tetap dijalankan dari nilai `count = 0`, sehingga teks akan dicetak untuk `count = 0, 1, dan 2`, total sebanyak tiga kali. Hasil cetaknya muncul dalam satu baris (tanpa pindah baris) karena tidak ada penggunaan karakter newline (`\n`) dalam perintah cetaknya.



```
perc2.m x +
count=0;
while (count <3)
    fprintf('hello world');
    count=count+1;
end
```

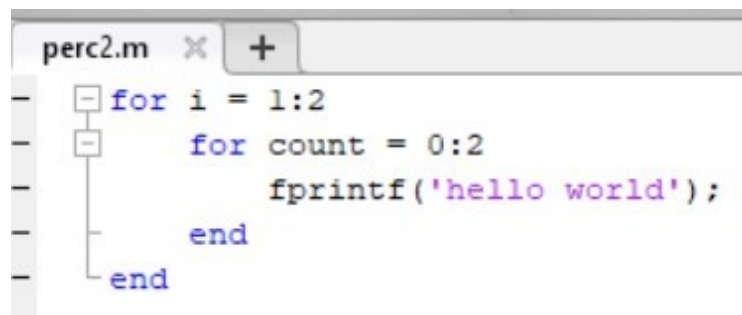
**For Loop** memiliki kemiripan dengan **while loop**, namun perbedaannya terletak pada penentuan nilai awal dan akhir perulangan yang sudah ditetapkan sejak awal. Oleh karena itu, ketika jumlah iterasi sudah diketahui, penggunaan **for loop** menjadi lebih praktis dan efisien.

Sebagai contoh, pada perintah `count = 0:2`, perulangan akan berlangsung sebanyak tiga kali dengan nilai `count` berturut-turut 0, 1, dan 2. Sama seperti pada contoh sebelumnya, perintah akan mencetak teks "hello world" sebanyak tiga kali secara berdampingan dalam satu baris, karena tidak disertai karakter pindah baris (`\n`).

**Nested Loop** adalah perulangan yang berada di dalam perulangan lainnya. Dalam struktur ini, setiap kali perulangan luar dijalankan satu kali, perulangan di bagian dalam akan dieksekusi sepenuhnya dari awal hingga akhir.

Sebagai ilustrasi, misalnya digunakan `for i = 1:2` dan di dalamnya terdapat `for count = 0:2`. Maka, untuk setiap nilai `i` (yaitu 1 dan 2), perulangan `count` akan dijalankan dari 0 hingga 2. Artinya, perulangan dalam akan terjadi 3 kali untuk setiap nilai `i`, menghasilkan total 6 kali perulangan.

Pada program ini, teks "hello world" akan dicetak sebanyak 3 kali untuk setiap iterasi `i`, sehingga secara keseluruhan akan muncul 6 kali di layar.



```
perc2.m x +
- for i = 1:2
-   for count = 0:2
-       fprintf('hello world');
-   end
- end
```