

LOOPS

132307264

Loop/ Perulangan

Merupakan struktur pengkodean untuk melakukan perintah yang sama secara terus menerus



Perulangan dibagi menjadi

- **Counted loop** merupakan perulangan yang jelas dan sudah tentu banyak kali perulangannya.
(perulangan *for*)
- **Uncounted loop**, merupakan perulangan yang tidak jelas berapa kali ia harus mengulang.
(perulangan *while* dan *do... while*)

Beberapa istilah yang sering ada pada perulangan

Start (kondisi pada saat awal perulangan)

Biasanya kondisi awal ini berisi perintah untuk memberikan nilai kepada **variabel counter**. *Variabel counter* sendiri adalah sebuah variabel yang menentukan berapa banyak perulangan dilakukan. Kebanyakan programmer menggunakan variabel `i` sebagai variabel counter (tidak harus, boleh memakai variabel lain).

Condition adalah kondisi yang harus dipenuhi agar perulangan berjalan. Selama kondisi ini terpenuhi, maka *compiler* bahasa C++ akan terus melakukan perulangan. Misalnya *condition* ini berisi perintah `i < 7`, maka selama *variabel counter* `i` berisi angka yang kurang dari 7, terus lakukan perulangan.

Increment adalah bagian yang dipakai untuk memproses variabel counter agar bisa memenuhi kondisi akhir perulangan. Bagian ini akan selalu di eksekusi di setiap perulangan.

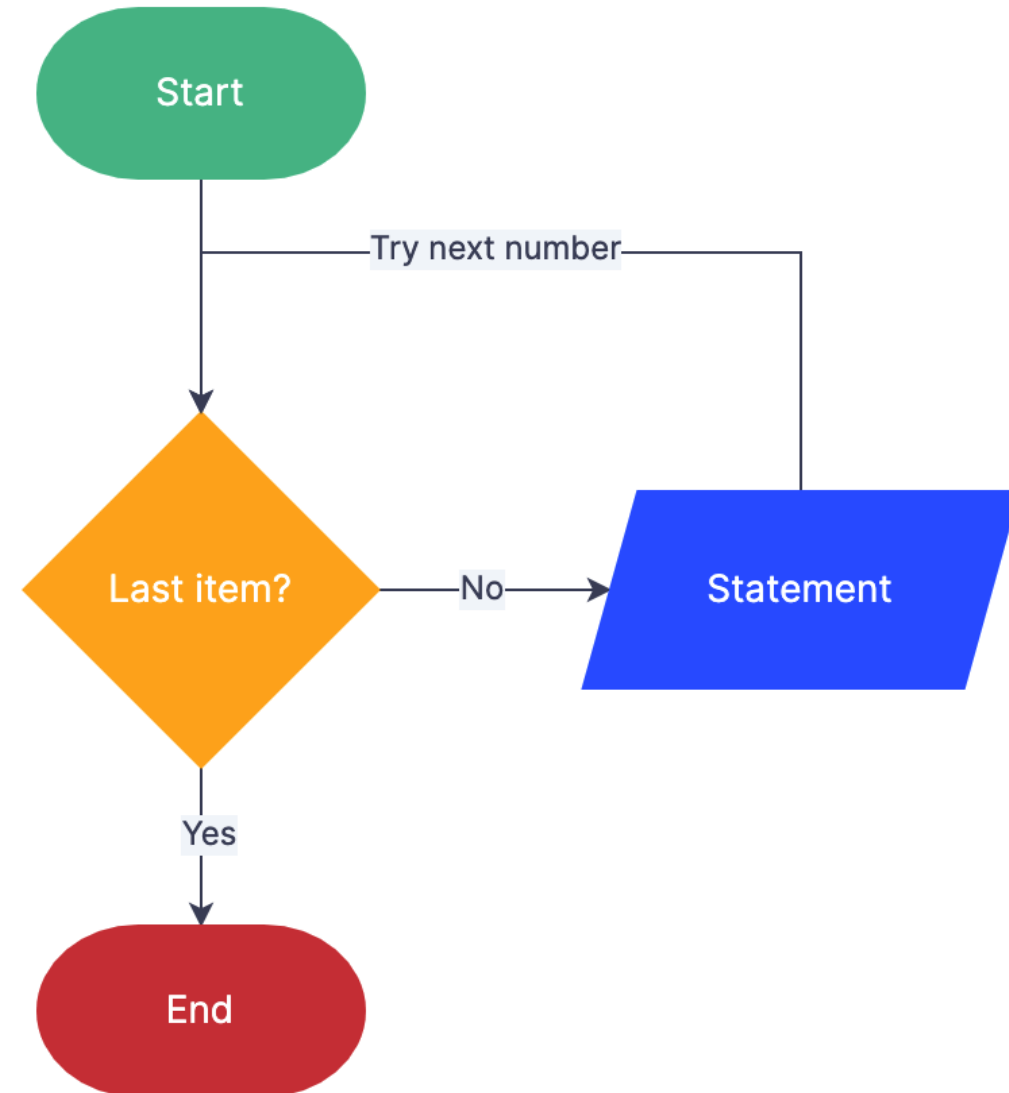
(Contoh penggunaan `i++`, yang sebenarnya sama dengan `i = i + 1`, artinya dalam setiap perulangan naikan variabel `i` sebanyak 1 angka. Namun pemberian nilai lain seperti `i = i + 2` juga diperbolehkan, sehingga variabel counter akan naik 2 angka setiap perulangan)

for

Yang harus diperhatikan pada saat membuat LOOP dengan *for*:

- ✓ Kondisi awal perulangan.
- ✓ Kondisi pada saat perulangan.
- ✓ Kondisi yang harus dipenuhi agar perulangan berhenti.

For Loop Flowchart



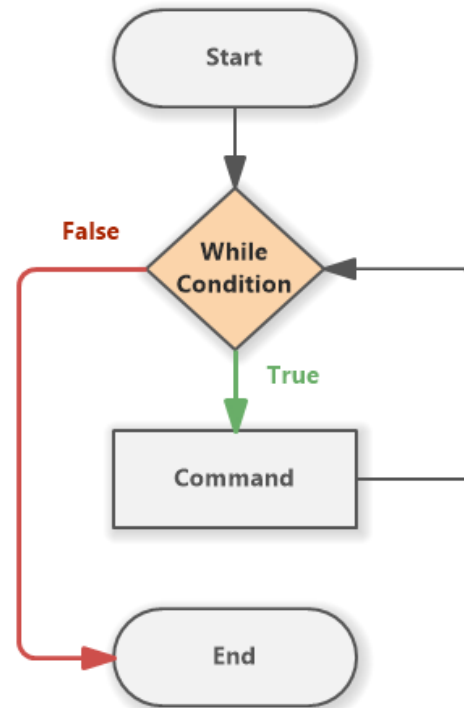
main.cpp	Run	Output
<pre>1 #include <iostream> 2 using namespace std; 3 4 int main() { 5 for (int i = 0; i < 5; i++) { 6 cout << i << "\n"; 7 } 8 return 0; 9 }</pre>		<pre>/tmp/J08iSDWqca.o 0 1 2 3 4</pre>

- Pernyataan 1 menetapkan variabel sebelum loop dimulai (int i = 0).
- Pernyataan 2 mendefinisikan kondisi untuk menjalankan loop (i harus kurang dari 5). Jika true, maka loop akan diulang, jika false, loop akan berakhir.
- Pernyataan 3 meningkatkan nilai i sebanyak 1 (i++) setiap kali blok kode dalam loop telah dieksekusi.

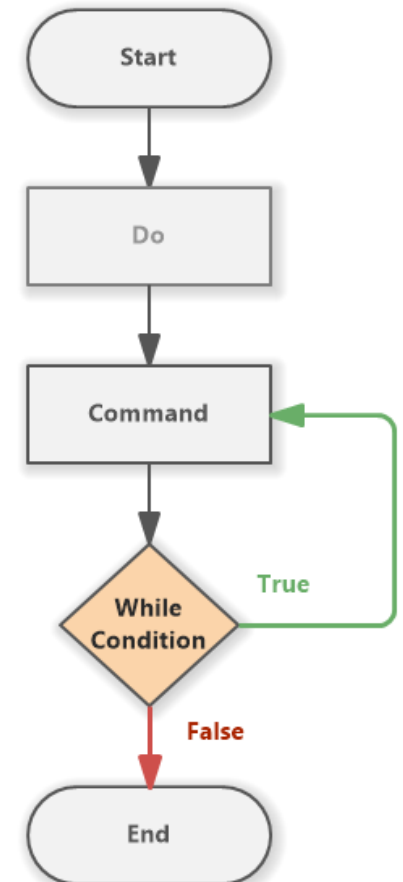
While dan Do ... While

- Perulangan while akan melakukan perulangan kalau kondisi (syarat) terpenuhi.
- Sedangkan do/while melakukan perulangan dulu, kemudian memeriksa kondisinya atau syaratnya. Kalau kondisi terpenuhi, maka do/while akan melanjutkan perulangan. Sebaliknya, dia akan berhenti (break).

WHILE



DO-WHILE



main.cpp		Run	Output
<pre>1 #include <iostream> 2 using namespace std; 3 4 int main() 5 { 6 int i = 0; 7 while (i < 5){ 8 cout << i << endl; 9 i++; 10 } 11 12 return 0; 13 }</pre>			<pre>/tmp/J08iSDWqca.o 0 1 2 3 4 </pre>

while

main.cpp		Run	Output
<pre>1 #include <iostream> 2 3 using namespace std; 4 5 int main() 6 { 7 int i = 0; 8 do { 9 cout << i << endl; 10 i++; 11 } 12 while (i < 5); 13 14 return 0; 15 }</pre>			<pre>/tmp/J08iSDWqca.o 0 1 2 3 4 </pre>

do ... while