# Contoh Program C++ Perulangan Sederhana dan Penjelasannya

Mari kita sedikit mengenal apa itu perulangan:

Mungkin anda juga pernah dihukum guru anda untuk menulis sebuah kalimat perjanjian yang intinya tidak akan mengulangi perbuatan salah anda sampai papan tulis penuh.

Misalnya menulis:

Saya tidak akan bolos sekolah lagi

……………………………………….

……………………………………….

……………………………………….

Saya tidak akan bolos sekolah lagi

**Catatan:** Anggaplah bagian titik-titik merupakan tulisan yang sama.

Seperti yang bisa anda lihat diatas, kita diminta untuk menulis kalimat yang sama sebanyak (contoh: 50 kali). Tentunya hal ini menjadi pekerjaan yang melelahkan.

Namun kabar baiknya, kita mengenal fungsi perulangan atau looping.

Dalam bahasa pemrograman kita akan membutuhkan *looping* untuk mengerjakan sesuatu yang sifatnya berulang. Dalam contoh diatas mencetak tulisan 50 kali.

Dengan perulangan, anda bisa menghemat sekian banyak baris kode yang pada dasarnya cukup diketik satu kali. Untuk itulah program perulangan digunakan.

( Silahkan baca: [Algoritma Perulangan](https://kodedasar.com/algoritma-perulangan/) )

**Table Of Contents**

* [Pengertian Perulangan atau Looping](https://kodedasar.com/perulangan-cpp/#pengertian-perulangan-atau-looping)
* [#1 Perulangan dengan fungsi for](https://kodedasar.com/perulangan-cpp/#1-perulangan-dengan-fungsi-for)
* [#2 Perulangan dengan fungsi while](https://kodedasar.com/perulangan-cpp/#2-perulangan-dengan-fungsi-while)
* [#3 Perulangan dengan fungsi do while](https://kodedasar.com/perulangan-cpp/#3-perulangan-dengan-fungsi-do-while)
* [Manarik Kesimpulan](https://kodedasar.com/perulangan-cpp/#manarik-kesimpulan)

**Pengertian Perulangan atau Looping**

Perulangan dapat diartikan sebagai berikut:

Perulangan (atau dalam bahasa inggris disebut dengan *loop*) adalah instruksi program yang bertujuan untuk mengulang beberapa baris perintah.

Fungsi paling sederhana yaitu untuk mempermudah melakukan suatu proses yang berulang-ulang, seperti mencetak angka dari 1 – 100.

Ada 3 cara untuk melakukan perulangan di C++ yaitu: for, while, & do while.

**#1 Perulangan dengan fungsi for**

Penulisan dasar format perulangan for dalam C++ adalah sebagai berikut:

for (start; condition; increment)

{

statement;

}

**Start** adalah kondisi pada saat awal perulangan.

Biasanya kondisi awal ini digunakan untuk membuat dan memberikan nilai kepada variabel yang digunakan untuk mengontrol perulangan.

Misalkan, kita akan membuat variabel i, maka untuk kondisi start ini, kita juga harus memberikan nilai awal untuk variabel i, misalnya dengan 1, maka i=1.

Misalkan kita ingin menghentikan perulangan jika variabel i telah mencapai nilai 10, maka pada bagian *condition* ini kita membuat perintah i<=10.

Yang berarti selama nilai i kurang atau sama dengan 10, terus lakukan perulangan.

**Increment** adalah bagian yang digunakan untuk memproses variabel agar bisa memenuhi kondisi akhir perulangan.

Umumnya nilai variable tersebut bertambah (i++) / berkurang (i--) 1 (satu).

**Condition** adalah kondisi yang harus dipenuhi agar perulangan dijalankan. Selama kondisi ini terpenuhi, maka C++ akan terus melakukan perulangan.

**Statement** adalah bagian kode program yang akan diproses secara terus-menerus selama proses perulangan berlangsung.

Kita membuat blok program di antara tanda kurung kurawal ({ dan }) sebagai penanda bahwa bagian di dalam kurung kurawal inilah yang akan dikenai proses perulangan.

Contoh looping dengan fungsi for pada program c++:

#include <iostream>

using namespace std;

void CetakAngka (int i, int j){

for (i = 1; i <= j; i++){

cout << i << '\n';

}

}

int main ()

{

CetakAngka (1,10);

}

**#2 Perulangan dengan fungsi while**

Dalam perulangan while, program akan terus melakukan perulangan dengan mengeksekusi pernyataan target selama kondisi tertentu bernilai benar.

Penulisan dasar format perulangan while dalam C++ adalah sebagai berikut:

while(condition){

statement;

}

**Condition** adalah kondisi yang harus dipenuhi agar perulangan berlangsung.

Kondisi ini mirip seperti dalam perulangan for.

Condition ini akan diperiksa pada tiap perulangan, dan hanya jika hasilnya FALSE, maka proses perulangan berhenti. Artinya kita tidak tahu berapa banyaknya perulangan.

Karena, selama condition bernilai TRUE, maka perulangan akan terus dilakukan.

**Statement** adalah kode program yang akan diulang.

Tanda **kurung kurawal** diperlukan untuk membatasi blok program yang akan diulang. Jika statement hanya terdiri dari 1 baris, maka tidak diperlukan.

Contoh perulangan dengan fungsi while dalam program c++:

#include <iostream>

using namespace std;

int main () {

// Local variable declaration:

int a = 1;

// while loop execution

while( a < 10 ) {

cout << "value of a: " << a << endl;

a++;

}

return 0;

}

**#3 Perulangan dengan fungsi do while**

Perulangan while dan do-while pada dasarnya hampir sama.

do {

statement;

} while (condition);

Perbedaan terletak pada ’lokasi’ pengecekan kondisi perulangan.

Dalam struktur while, pengecekan untuk kondisi perulangan di lakukan di awal, sehingga jika kondisi tidak terpenuhi, maka perulangan tidak akan pernah dijalankan.

Namun pada perulangan do-while:

Pengecekan kondisi akan dilakukan di akhir perulangan, sehingga walaupun kondisi adalah FALSE, perulangan akan tetap berjalan minimal 1 kali.

**Statement** adalah kode program yang akan diulang & **condition** adalah kondisi yang harus dipenuhi agar perulangan berlangsung.

Contoh program looping dengan fungsi do while dalam pemrograman c++:

#include <iostream>

using namespace std;

int main () {

// Local variable declaration:

int a = 1;

// while loop execution

do {

cout << "value of a: " << a << endl;

a = a + 1;

} while ( a < 10 );

return 0;

}

**Manarik Kesimpulan**

Meskipun perulangan atau looping memiliki fungsi yang sama, namun ketiga fungsi tersebut digunakan pada kondisi yang berbeda.

* **While** - Perulangan while digunakan untuk kondisi perulangan dimana banyaknya perulangan tidak dapat dipastikan pada saat penulisan program.
* **For** - Perulangan for digunakan untuk kondisi perulangan dimana jumlah perulangan telah di tentukan di awal.
* **do-while** - Sama seperti perulangan while, dalam perancangan perulangan do-while ini kita harus memahami alur logika program yang dibuat. Banyaknya perulangan di tentukan pada saat program mencapai kondisi FALSE.

Kesalahan dalam alur logika akan membuat C++ tidak pernah berhenti (*infinity loop*).