# BAB V

# PERULANGAN

Perulangan data atau yang biasa disebut dengan “*looping*” adalah proses yang dilakukan secara berulang-ulang sampai batas yang ditentukan. Perulangan digunakan untuk mengulang sekumpulan perintah sesuai dengan kondisi yang diberikan. Dengan *loop*, suatu proses yang berulang dapat diimplementasikan menggunakan *statement-statement* yang pendek, tanpa harus menuliskan statement berulang-ulang.

Dalam bahasa C++ terdapat empat buah jenis struktur pengulangan, yaitu :

1. Struktur **For**
2. Struktur **While**
3. Struktur **Do-While**
4. Struktur **Nested Loop**

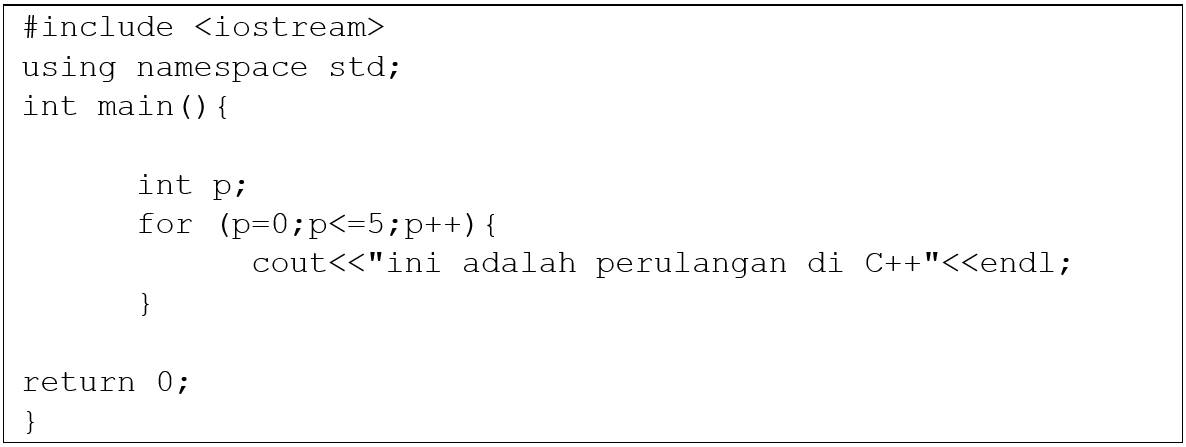
## Struktur For

Merupakan salah satu bentuk fungsi *looping*, digunakan untuk melaksanakan pernyataan berulang kali terhadap sejumlah nilai yang telah ditetapkan. *Looping* yang dilakukan oleh *for* telah diketahui batas awal, batas kondisi dan perubahannya. Tipe data dari variabel yang akan digunakan sebagai indeks haruslah tipe data yang mempunyai urutan yang teratur, misalnya tipe data int (0,1,2,…) atau char(‘a’,’b’,’c’,...).

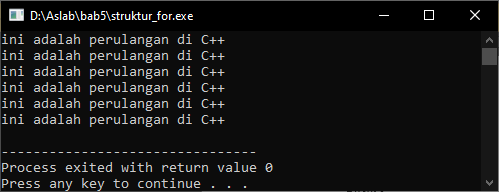
Bentuk umum:

|  |
| --- |
| for (inisialisasi ; kondisi ; perubahan) {  Statement;  } |

Contoh :



Tampilan :

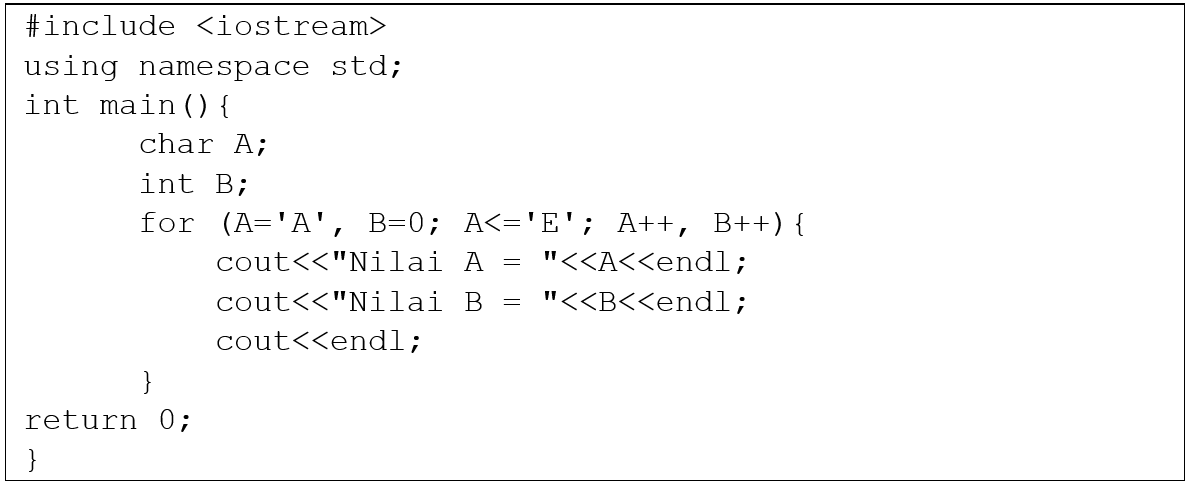


Analisa :

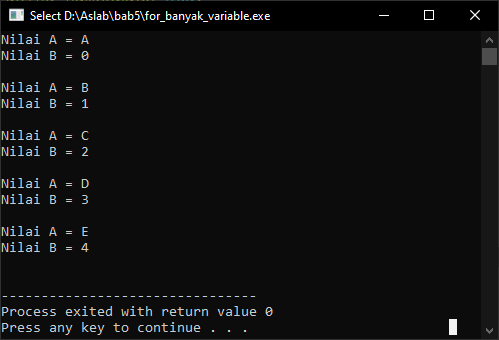
Pada program diatas, terdapat variabel p bertipe data integer. Kemudian variabel p akan di ulang dengan pernyataan for sesuai dengan kondisinya, yaitu p = 0 sampai p <=5, c++ diikuti dengan output “ini adalah perulangan di C++” yang akan di ulang sampai kondisi tersebut tidak terpenuhi lagi.

## Struktur For dengan Banyak Variabel

Struktur for di dalam C++ dapat juga melibatkan lebih dari satu variabel namun yang jelas satu diantaranya akan digunakan sebagai indeks pengulangan. Untuk memahami konsepnya, berikut ini disajikan sebuah contoh program dimana di dalamnya terdapat strukur for yang melibatkan dua buah variabel, yaitu A, B dan C.



Tampilan :



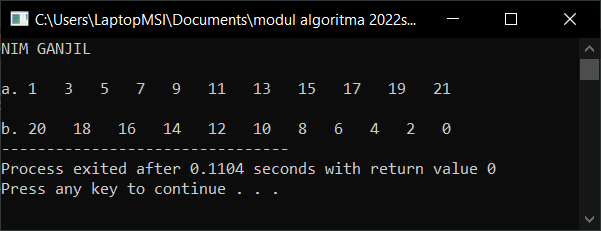
Analisa :

Pada program diatas, terdapat variabel A bertipe char akan digunakan sebagai indeks pengulangan, variabel B akan digunakan untuk menampung sebuah nilai. Penting untuk diingat bahwa dari variabel-variabel yang kita deklarasikan ada satu variabel yang dijadikan indeks perulangan.

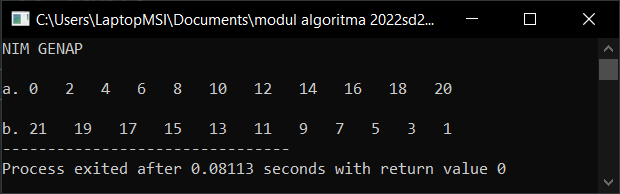
|  |  |  |
| --- | --- | --- |
| **#taugaksih**  **Jump Statement Break**  Pada saat kita menggunakan struktur perulangan, seringkali kita dituntut untuk melakukan peloncatan / lompatan statemen (jump statement). Kata "peloncatan" disini artinya kita memaksa agar eksekusi statemen berjalan sesuai urutan yang kita inginkan, yaitu dengan cara meloncat dari statemen yang satu ke statemen yang lain. Beberapa jump statement sebagai berikut   1. Break   Pernyataan break biasanya di gunakan untuk menghentikan proses perulangan yang di gunakan   |  | | --- | | break; |   Contoh Program :   |  | | --- | | #include <iostream>  using namespace std;  int main() {  int a;  for(a = 1;a <= 10;a++){  if(a==8){  break;  }  cout << a << endl;  }  cout << "\nPerulangan berhenti di = " << a;  return 0;  } |   Tampilan Program :    Analisa :  Break digunakan untuk menghentikan perulangan, seperti contoh di atas, apabila nilai adalah 8 maka proses perulangan akan di hentikan sehingga data yang tampil adalah 1 sampai 7 |

**Latihan!**

1. Buatlah program sesuai dengan tampilan program dibawah ini, dengan ketentuan :
2. NIM ganjil akan menampilkan bilangan ganjil secara ascending dan bilangan genap secara descending.



1. NIM genap akan menampilkan bilangan genap secara ascending dan bilangan ganjil secara descending.



1. Buatlah program dengan hasil seperti dibawah ini:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| a. | -5 | -12 | -19 | -26 | -33 | -40 |
| b. | 1 | -3 | 9 | -27 | 81 | -243 |
| c. | 6 | 7 | 9 | 12 | 16 | 21 |
| d. | 1 | 2 | 4 | 6 | 8 | 9 |

1. Buatlah program dengan hasil seperti dibawah ini:

