# BAB VII

# NESTED LOOP

## Struktur Nested

Pernyataan nested for adalah suatu perulangan for di dalam perulangan for yang lain. Di dalam penggunaan nested for, perulangan yang di dalam terlebih dahulu dihitung hingga selesai, kemudian perulangan yang di luar diselesaikan.

Biasanya, nested loops digunakan untuk membuat aplikasi matematika yang menggunakan baris dan kolom. Loop luar, biasanya digunakan untuk mendefinisikan kolom. Sedangkan loop dalam, digunakan untuk mendefinisikan baris.

Bentuk umum :

1. Struktur Nested For

|  |
| --- |
| for (inisialisasi1 ; kondisi1 ; perubahan1) {  for (inisialisasi2 ; kondisi2 ; perubahan2) {  Statement;  }  } |

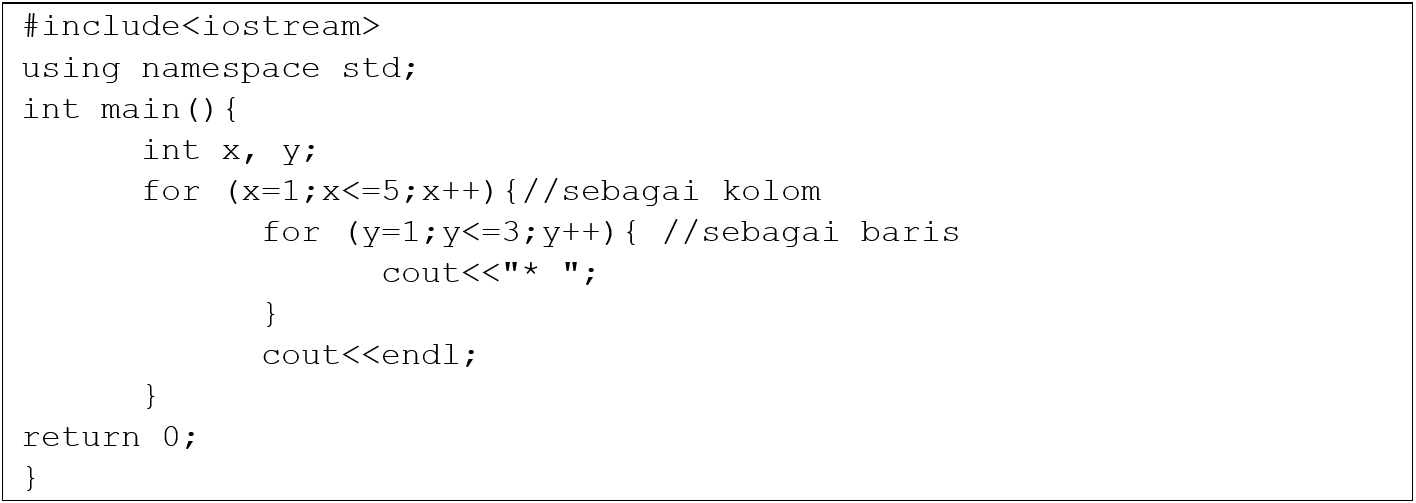
1. Struktur Nested While

|  |
| --- |
| Inisialisasi1;  while(kondisi){  inisialisasi2;  while(kondisi){  statemen;  perubahan2  }  Perubahan1;  } |

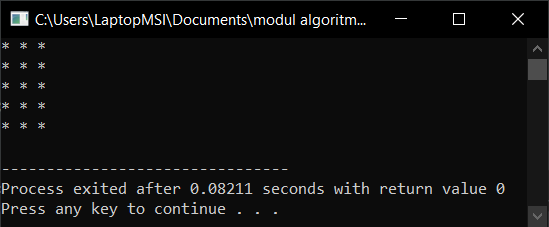
1. Struktur Nested Do-While

|  |
| --- |
| Inisialisasi1;  do{  inisialisasi2;  do{  Statement;  Perubahan2;  }while(kondisi2);  Perubahan1;  }while(Kondisi1); |

Contoh :



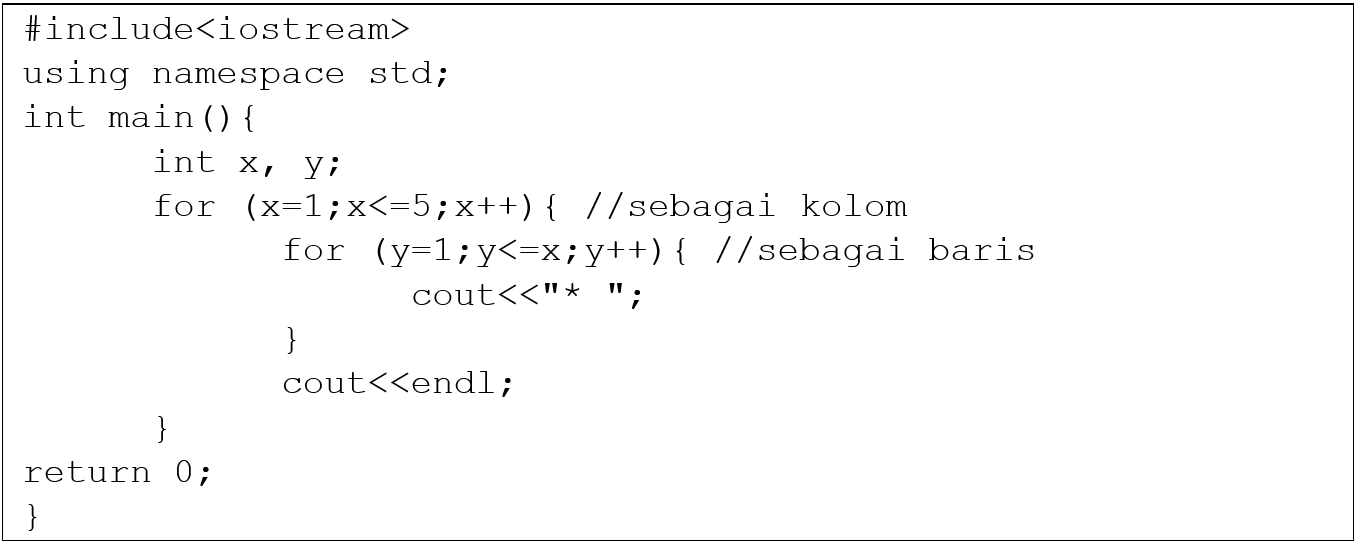
Tampilan :



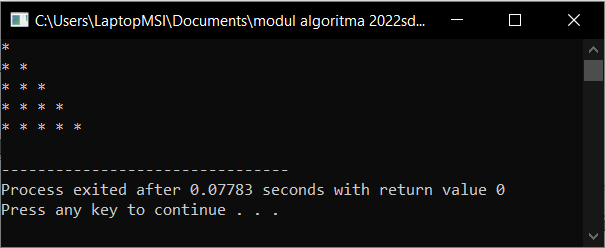
Analisa :

Pada program diatas terdapat variabel x, y, dan input bertipe integer. Pada for pertama ( for (x=1;x<=5;x++) ) digunakan untuk mengerjakan output vertical, sedangkan for kedua (for (y=1;y<=x;y++) ) digunakan untuk mengerjakan output horizontal yang akan mengoutputkan ‘\*’ .

Contoh :



Tampilan :



Analisa :

Pada program diatas terdapat variabel x, y, dan input bertipe integer. Pada for pertama ( for (x=1;x<=5;x++) ) digunakan untuk mengerjakan output vertical, sedangkan for kedua (for (y=1;y<=x;y++) ) digunakan untuk mengerjakan output horizontal yang akan mengoutputkan ‘\*’ . Dimana pada saat x=1 maka akan mengerjakan output mulai dari y=1 sampai y=x ( dimana x=1 ). Kemudian pada saat x=2 maka akan mengerjakan output mulai dari y=1 sampai y=x ( dimana x=2 ). Kemudian pada saat x=3 maka akan mengerjakan output mulai dari y=1 sampai y=x ( dimana x=3 ). Kemudian pada saat x=4 maka akan mengerjakan output mulai dari y=1 sampai y=x ( dimana x=4 ).

|  |  |  |  |
| --- | --- | --- | --- |
| **#taugaksih**  **Jump Statement Goto**   1. goto   goto adalah sebuah instruksi untuk mengarahkan program membaca pernyataan yang di awali sebuah label, label sendiri didalam kode program berfungsi untuk menyatakan lokasi yang akan dituju. untuk penamaan label bebas selama bukan identifier c++.   |  |  | | --- | --- | | goto label;  ................  Statements  ................  label:  Statements | label:  ................  Statements  ................  goto label;  Statements |   Contoh Program :   |  | | --- | | #include <iostream>  using namespace std;  int main(){  int a;  for(a=1;a<=10;a++){  if(a==8){  goto jump;  }  cout<<a<<" ";  }  jump:  return 0;  } |   Tampilan Program :    Analisa :  Goto di gunakan untuk mengarahkan eksekusi program ke label yang kita tentukan, seperti contoh perulangan di atas, apabila nilai a=8 maka program akan mengeksekusi goto ke label “jump”, lalu menampilkan output “program berhenti” |

**LATIHAN !**

1. Buatlah program dengan hasil sebagai berikut :

|  |  |  |  |
| --- | --- | --- | --- |
| **a** | **b** | **c** | **d** |
|  |  |  |  |

1. Buatlah program dengan hasil sebagai berikut :

|  |  |
| --- | --- |
| **Opsi 1** | **Opsi 2** |
|  |  |