

MODUL MATA KULIAH

ANALISIS DAN DESAIN ALGORITMA

PG167 – 3 SKS



**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

TIM PENYUSUN

**JAKARTA
SEPTEMBER 2019**

Ita Novita, S.Kom., M.T.I
Dr. Achmad Solichin, S.Kom., M.T.I
Atik Ariesta, S.Kom., M.Kom



MODUL PERKULIAHAN #6

STRUKTUR PERULANGAN BERTINGKAT (NESTED LOOP)

Capaian Pembelajaran	:	Mahasiswa mampu memahami dasar penggunaan struktur perulangan bertingkat serta penggunaan break dan continue
Sub Pokok Bahasan	:	7.1. Penggunaan break dan continue 7.2. Struktur perulangan bertingkat (Nested Loop) 7.3. Contoh penggunaan Nested Loop
Daftar Pustaka	:	1. Gaddis, nd.2011. Starting Out with C++ from Control Structures through Objects .8th. Boston: Addison-Wesley. 2. Institue of Distance & Open Learning, n.d. UNIT I Algorithms, Flowcharts & Program Design in: INTRODUCTION TO C++. p. 205 3. Sjukani,Moh .2014. Algoritma (Algoritma & Struktur Data 1) Dengan C, C++, dan Java Edisi 9", Mitra Wacana Media.

STRUKTUR PERULANGAN BERTINGKAT (NESTED LOOP)

7.1. PENGGUNAAN BREAK DAN CONTINUE PADA PERULANGAN

PENGGUNAAN BREAK UNTUK MEMAKSA KELUAR DARI LOOP

Perintah **break** jika digunakan didalam perulangan berfungsi untuk 'menghentikan paksa' proses perulangan yang berlangsung.

Break biasanya digunakan setelah **kondisi IF**, untuk menyelesaikan 'kapan' perulangan harus diberhentikan.

Contoh:

Dalam lembar dokumen ada 10 buah bilangan integer. Susun algoritma untuk menginput bilangan-bilangan tersebut dan menghitung total nilai bilangan-bilangan yang diinput. Bila totalnya sudah >1000, maka berhenti menginput dan mencetak totalnya serta proses selesai walaupun belum semua bilangan diinput. Bila totalnya belum > 1000 maka input dan hitung terus total terus-menerus sampai 10 buah bilangan diinput dan dihitung totalnya, kemudian cetak total dan proses selesai.



Jawab:

Algoritma:

Deklarasi variabel I, N, Total

Total = 0

I = 1

Lakukan selama I ≤ 10

 Input nilai N

 Total = Total + N

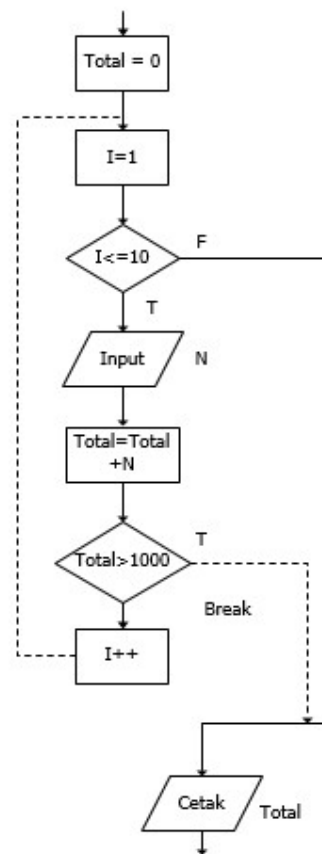
 Jika Total > 1000

 Maka Berhenti (break)

I = I + 1

Cetak Total

Flowchart



Program 7.1 Contoh Program Penggunaan Break Menggunakan For dalam Bahasa C

```
#include <stdio.h>
void main()
{ int I, N, Total;
  Total = 0;

  for(I=1;I<=10;I++) //berhenti input bila sudah 10x input
  { scanf ("%i", &N);
    Total= Total + N;
    If (Total > 1000) break;//berhenti bila Total>1000
  }

  printf("%i", Total)
}
```



Program 7.2 Contoh Program Penggunaan Break Menggunakan While dalam Bahasa C

```
#include <stdio.h>
void main()
{ int I, N, Total;
  Total = 0;

  I=1;
  while (I<=10) //berhenti input bila sudah 10x input
  {   scanf ("%i", &N);
      Total= Total + N;
      If (Total > 1000) break;//berhenti bila Total>1000
      I++;
  }

  printf("%i", Total);
}
```

Program 7.3 Contoh Program Penggunaan Break Menggunakan Do.. While dalam Bahasa C

```
#include <stdio.h>
void main()
{ int I, N, Total;
  Total = 0;

  I=1;
  do
  {   scanf ("%i", &N);
      Total= Total + N;
      If (Total > 1000) break;//berhenti bila Total>1000
      I++;
  }
  while (I<=10); //berhenti input bila sudah 10x input
  printf("%i", Total);
}
```

Sebenarnya untuk menyelesaikan persoalan diatas, selain menggunakan break bisa memasukkan 2 syarat diatas kedalam kondisi.

2 syarat tersebut yaitu:

1. Data yang diinput paling banyak 10 buah
2. Berhenti menginput bila Total > 1000



Sehingga bila menggunakan while, logikanya dapat dibuat sebagai berikut:

**Selama data yang diinput belum 10 buah dan
Totalnya belum >1000 (Total<=1000) maka lanjutkan input**

Contoh program 7.4 Penggunaan kondisi:

```
#include <stdio.h>
void main()
{ int I, N, Total;
  Total = 0; I=1;
  while (I<=10 && Total <=1000)
  { scanf ("%i", &N);
    Total= Total + N;
    I++;
  }
  printf("%i", Total);
}
```

PENGUNAAN CONTINUE UNTUK MEMAKSA MELANJUTKAN LOOP

Perintah **continue** digunakan untuk menghentikan perulangan yang saat ini terjadi (1 iterasi saja), dan kemudian melanjutkan perulangan iterasi berikutnya.

Perintah **continue** digunakan setelah **kondisi IF** yang digunakan untuk menyeleksi 'kapan' perulangan harus di-skip atau dilewati.

Contoh:

Dalam lembar dokumen ada banyak sekali bilangan integer yang merupakan nilai ujian mahasiswa. Susun algoritma untuk mengambil 5 nilai pertama yang nilainya ≥ 60 dan mencetak nilai rata-rata. Dipastikan yang mendapat nilai ≥ 60 lebih dari 5 orang.

Ilustrasi:

Diinput satu persatu dari awal, tapi yang diambil hanya yang nilainya ≥ 60 . Bila nilai yang diinput < 60 , maka lanjutkan menginput nilai berikutnya. Bila telah dapat 5 orang yang nilainya ≥ 60 , maka proses input selesai, kemudian cetak nilai rata-rata.



Jawab:

Algoritma:

Deklarasi variabel I, N, Total, Rata, Count

Count = 0

Total = 0

I = 1

Lakukan selama $I \leq 5$

Input nilai N

Jika $N < 60$ maka lanjutkan (continue)

Total=Total+N

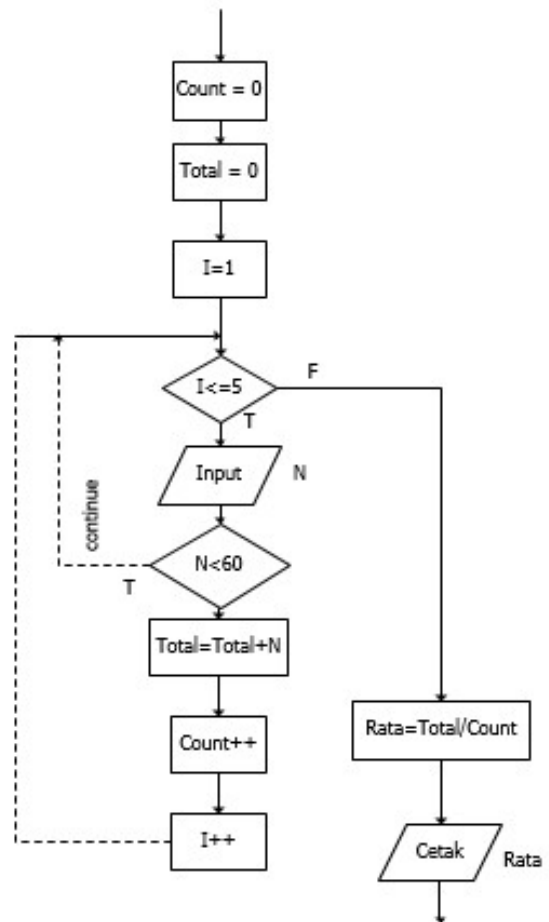
Count=Count+1

I++

Rata=Total/Count

Cetak Rata

Flowchart:



Program 7.5 Contoh Program Penggunaan Continue Menggunakan While dalam Bahasa C

```
#include <stdio.h>
void main()
{ int I, N, Total, Count;
  Total = 0;
  Count = 0;
  I=1;
  while (I<=5)
  { scanf ("%i", &N);
    if (N<60) continue;
    Total= Total + N;
    Count++;
    I++;
  }
  Rata= Total/Count;
  printf("%i", Rata);
}
```

Keterangan:

Misal data yang ada dalam dokumen sebagai berikut:

75, 50, 60, 55, 80, 70, 45, 52, 65, 70, 86, 50, dst

Akan diinput: 75, 50, 60, 55, 80, 70, 45, 52, 65

Yang ditambahkan ke Total hanya 5 buah bilangan pertama yang ≥ 60 yaitu: 75, 60, 80, 70, 65

Sehingga Total = 350

Dan Nilai rata-rata yang tercetak adalah $350/5 = 70$



7.2. STRUKTUR PERULANGAN BERTINGKAT (NESTED LOOP)

PENGERTIAN NESTED LOOP

Nested loop atau perulangan bertingkat adalah struktur perulangan yang berada di dalam struktur perulangan lainnya, biasanya melibatkan 2 kali perulangan.

BENTUK UMUM PERULANGAN BERTINGKAT (NESTED LOOP)

Program A

```
#include<stdio.h>
void main()
{  int J;
   for ( J=1; J<=3; J++ )
   {
       -
       -
       -      Loop
       -
   }
}
```

Loop ini dikerjakan 3x

Program B

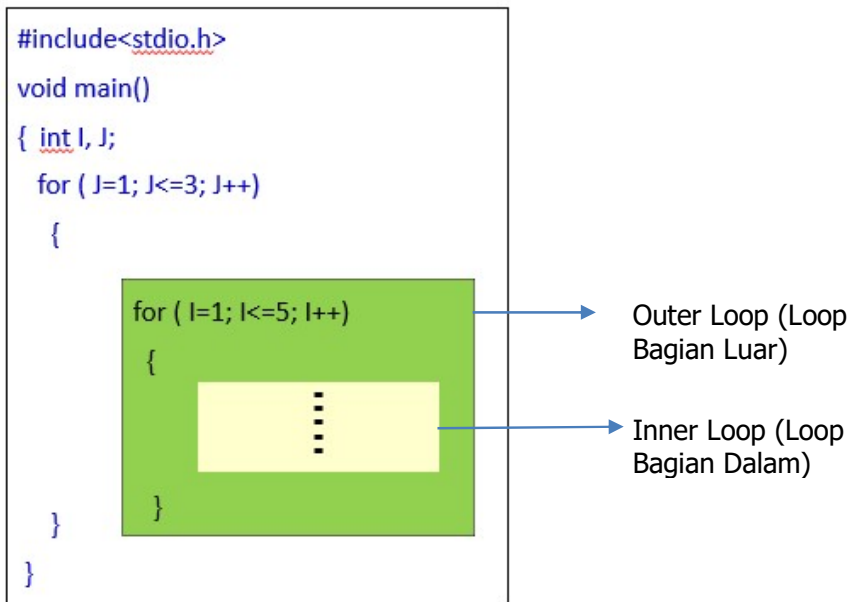
```
for ( I=1; I<=5; I++ )
{
    -
    -
    -
    -
}
```

Loop ini dikerjakan 5x

Dalam program A ada loop yang dikerjakan sebanyak 3x, dan di penggalan program B ada loop yang dikerjakan sebanyak 5x.

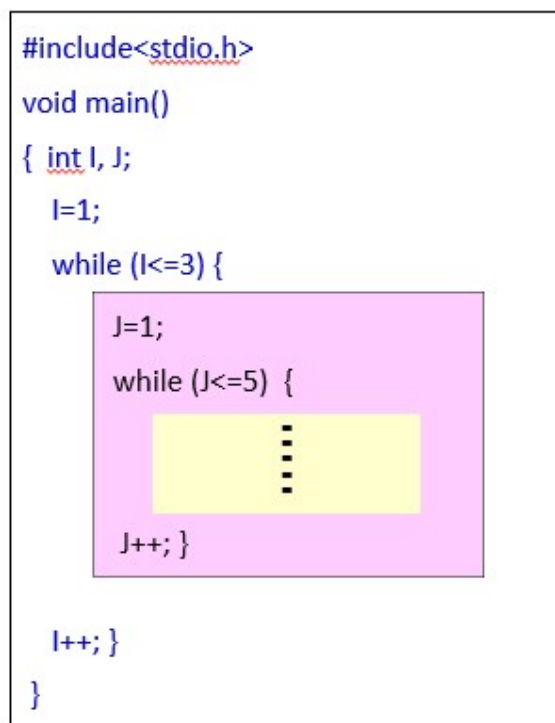
Bila penggalan program B dimasukkan kedalam program A menggantikan kotak 1 maka berbentuk **suatu Loop didalam Loop** yang biasa disebut **Nested Loop** atau **Loop Bertingkat**.

Hasil Penggabungan :

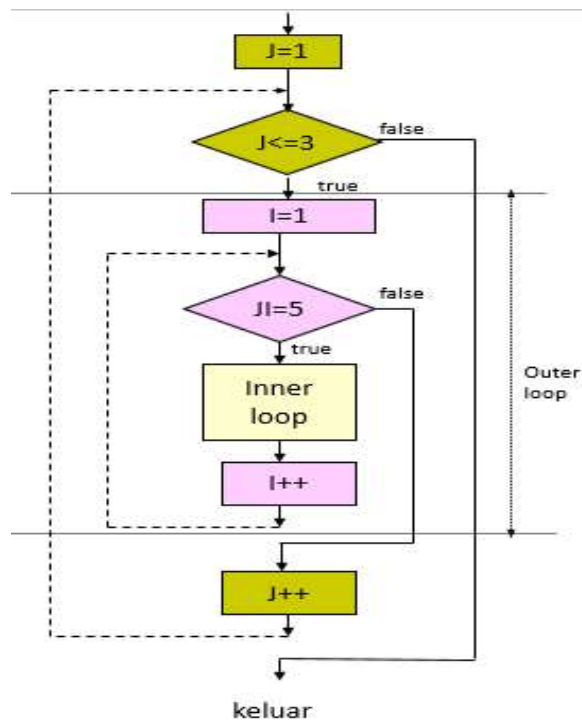


Instruksi yang ada didalam loop ini akan dilaksanakan sebanyak 3x5 kali = 15 kali

Untuk loop yang dibentuk dengan while, maka nested loopnya sebagai berikut:



Flowchart (bentuk for maupun while sama):



Nested Loop dapat merupakan gabungan antara for() dan while() seperti berikut ini:

```
#include<stdio.h>
void main()
{ int I, J;
  I=1;
  while (I<=3) {
    for ( J=1; J<=5; J++)
    {
      // ...
    }
    I++;
  }
}
```

```
#include<stdio.h>
void main()
{ int I, J;
  for ( I=1; I<=3; I++)
  {
    J=1;
    while (J<=5) {
      // ...
      J++;
    }
  }
}
```

7.3. CONTOH PENGGUNAAN NESTED LOOP

Contoh 1: (Looping 3 x 5)

```
#include<stdio.h>
main()
{ int I, J;
  for ( I=1; I<=3; I++)
  {
    for ( J=1; J<=5; J++)
    {
      printf("\nJakarta")
    }
  }
}
```

Perhatikan tabel berikut untuk contoh 1:

I	J	Cetak
1	1	Jakarta
	2	Jakarta
	3	Jakarta
	4	Jakarta
	5	Jakarta
2	1	Jakarta
	2	Jakarta
	3	Jakarta
	4	Jakarta
	5	Jakarta
3	1	Jakarta
	2	Jakarta
	3	Jakarta
	4	Jakarta
	5	Jakarta

Hasil:

Jakarta

Jakarta

-

-

Jakarta

Mencetak Jakarta sebanyak 15 kali

Contoh 2: (Looping 3 x 5)

```
#include<stdio.h>
main()
{ int I, J;
  for ( I=1; I<=3; I++)
  {
    for (J=1; J<=5; J++)
    {
      printf(" %i", J);
    }
  }
}
```

Perhatikan tabel berikut untuk contoh 2:

I	J	Cetak
1	1	1
	2	2
	3	3
	4	4
	5	5
2	1	1
	2	2
	3	3
	4	4
	5	5
3	1	1
	2	2
	3	3
	4	4
	5	5

Hasil:

1 2 3 4 5 1 2 3 4 5 1 2 3 4 5



Contoh 3: (Loop 3 x 5)

```
#include<stdio.h>
main()
{ int I, J;
  for ( I=1; I<=3; I++)
  {
    for (J=1; J<=5; J++)
    {
      printf(" %i", J);
    }
    printf("\n");
  }
}
```

Perhatikan tabel berikut untuk contoh 3:

I	J	Cetak
1	1	1
	2	2
	3	3
	4	4
	5	5
2	1	1
	2	2
	3	3
	4	4
	5	5
3	1	1
	2	2
	3	3
	4	4
	5	5

Hasil:

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

Setiap 5x cetak nilai J, turun 1 baris

Contoh 4: (Loop 3 x 5)

```
#include<stdio.h>
main()
{ int I, J;
  for ( I=1; I<=3; I++)
  {
    for (J=1; J<=5; J++)
    {
      printf(" %i", I);
    }
    printf("\n");
  }
}
```

Perhatikan tabel berikut untuk contoh 4:

I	J	Cetak
1	1	1
	2	1
	3	1
	4	1
	5	1
2	1	2
	2	2
	3	2
	4	2
	5	2
3	1	3
	2	3
	3	3
	4	3
	5	3

Hasil:

1 1 1 1 1

2 2 2 2 2

3 3 3 3 3



Contoh 5: (Loop 5 x 3)

```
#include<stdio.h>
main()
{ int I, J;
  for ( I=1; I<=5; I++)
  {
    for (J=1; J<=3; J++)
    {
      printf(" %i", J);
    }
    printf ("\n");
  }
}
```

```
for (J=1; J<=3; J++)
{
  printf(" %i", J);
}
```

Hasil:

```
1 2 3
1 2 3
1 2 3
1 2 3
1 2 3
```

Perhatikan tabel berikut untuk contoh 5:

I	J	Cetak
1	1	1
	2	2
	3	3
2	1	1
	2	2
	3	3
3	1	1
	2	2
	3	3
4	1	1
	2	2
	3	3
5	1	1
	2	2
	3	3

Contoh 6:

```
#include<stdio.h>
main()
{ int I, J, N;
  N = 1;
  for ( I=1; I<=3; I++)
  {
    for (J=1; J<=5; J++)
    {
      printf(" %3i", N);
      N=N+1;
    }
    printf ("\n");
  }
}
```

```
for (J=1; J<=5; J++)
{
  printf(" %3i", N);
  N=N+1;
}
```

Hasil:

```
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
```

Perhatikan tabel berikut untuk contoh 6:

I	J	N
1	1	1
	2	2
	3	3
	4	4
	5	5
2	1	6
	2	7
	3	8
	4	9
	5	10
3	1	11
	2	12
	3	13
	4	14
	5	15



Contoh 7:

```
#include<stdio.h>
main()
{ int I, J, N;
  N = 1;
  for ( I=1; I<=5; I++)
  {
    for (J=1; J<=3; J++)
    {
      printf(" %3i", N);
      N=N+1;
    }
    printf("\n");
  }
}
```

Perhatikan tabel berikut untuk contoh 7:

I	J	N
1	1	1
	2	2
	3	3
2	1	4
	2	5
	3	6
3	1	7
	2	8
	3	9
4	1	10
	2	11
	3	12
5	1	13
	2	14
	2	15

Hasil:

```
1  2  3
4  5  6
7  8  9
10 11 12
13 14 15
```

Contoh 8:

```
#include<stdio.h>
main()
{ int I, J;
  for ( I=1; I<=5; I++)
  {
    for (J=1; J<=5; J++)
    {
      printf(" %3i", J);
    }
    printf("\n");
  }
}
```

Perhatikan tabel berikut untuk contoh 8:

I	J
1	1
	2
	3
	4
	5
2	2
	3
	4
	5
	5
3	3
	4
	5
	5
	5
4	4
	5
	5
	5
	5
5	5
	5
	5
	5
	5

Hasil:

```
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5
```



Contoh 9:

```
#include<stdio.h>
main()
{ int I, J, N=1;
  for ( I=1; I<=5; I++)
  {
    for (J=1; J<=I; J++)
    {
      printf(" %3i", N);
      N=N+1;
    }
    printf("\n");
  }
}
```

Perhatikan tabel berikut untuk

Contoh 9:

I	J	N
1	1	1
2	1	2
2	2	3
3	1	4
3	2	5
3	3	6
4	1	7
4	2	8
4	3	9
4	4	10
5	1	11
5	2	12
5	3	13
5	4	14
5	5	15

Hasil:

```
1
2  3
4  5  6
7  8  9 10
11 12 13 14 15
```

Contoh 10:

```
#include<stdio.h>
main()
{ int I, J; char C='A';
  for ( I=1; I<=5; I++)
  {
    for (J=1; J<=3; J++)
    {
      printf(" %3c", C);
      C++;
    }
    printf("\n");
  }
}
```

Perhatikan tabel berikut untuk

contoh 10:

I	J	Char C
1	1	A
	2	B
	3	C
2	1	D
	2	E
	3	F
3	1	G
	2	H
	3	I
4	1	J
	2	K
	3	L
5	1	M
	2	N
	3	O

Hasil:

```
A B C
D E F
G H I
J K L
M N O
```



Contoh11:

Buatlah algoritma dan flowchart untuk mencetak deret bilangan berikut ke layar!

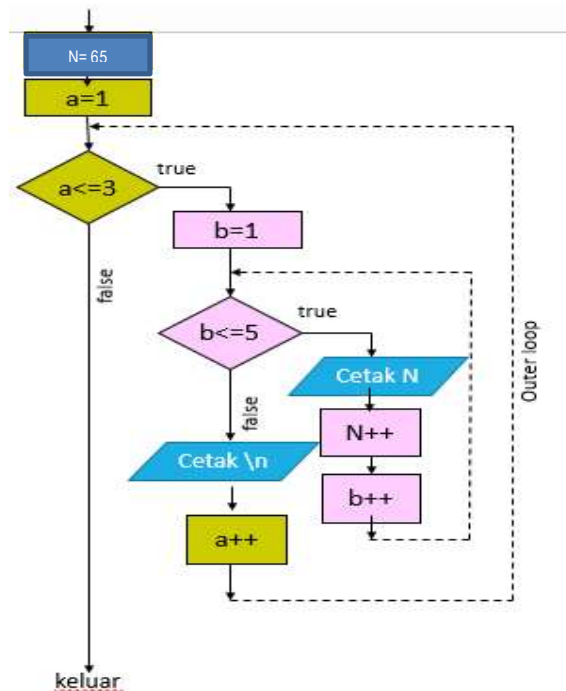
A B C D E
F G H I J
K L M N O

Jawab:

Algoritma:

Deklarasi variabel a,b,N
N=65
a = 1
Lakukan selama a<=3
 b = 1
 Lakukan selama b<=5
 Cetak N
 N=N+1
 b=b+1
Cetak untuk pindah baris
a=a+1

Flowchart:



Program 7.6 Contoh Program Mencetak Deret Menggunakan Nested For dalam Bahasa C

```
#include <stdio.h>
void main()
{ int a,b, char N;
  N = 65;
  for (a=1; a<=3; a++)
  { for (b=1; b<=5; b++)
    { printf ("%3c", N);
      N++;
    }
    printf("\n");
  }
}
```



Program 7.7 Contoh Program Mencetak Deret Menggunakan Nested While dalam Bahasa C

```
#include <stdio.h>
void main()
{ int a,b,char N;
  N = 65;
  a=1;
  while (a<=3)
  { b=1;
    while (b<=5)
    { printf ("%3c", N);
      N++;
      b++; }
    printf("\n");
    a++;
  }
}
```



KESIMPULAN

Struktur perulangan bertingkat atau nested loop adalah perulangan di dalam perulangan. Umumnya, perulangan di dalam perulangan memiliki hubungan yang saling terkait dalam menyelesaikan sebuah kasus.

Jika perulangan luar (outer loop) tidak memiliki hubungan terkait dan tidak memiliki kepentingan dalam melakukan proses komputasi, sebaiknya hindari penggunaan nested loop karena akan menghabiskan waktu eksekusi yang sia-sia dan program tidak berjalan optimal.



SOAL LATIHAN

1. Dalam lembar dokumen banyak sekali nilai-nilai integer yang kalau nilainya di total akan lebih besar dari 1000. Susun algoritma untuk menginput bilangan-bilangan tersebut satu per satu dan menghitung totalnya. Input berhenti apabila total nilai yang diinput lebih besar dari 500. Kemudian mencetak total tersebut dan proses selesai.

2. Susun algoritma untuk mencetak deret berikut ini:

a. 5 5 5 5 5
 10 10 10 10 10
 15 15 15 15 15

b. 1 1 1 1 1
 2 2 2 2
 3 3 3
 4 4
 5

3. Apa yang tercetak jika program berikut dijalankan:

a.

```
#include<stdio.h>
void main()
{   int I, J, T;
    T = 0;
    for ( I=1; I<=5; I+=2)
    {   for(J = I; J<=9; J+=3)
        {   printf("%4i", J );
        }
        printf("\n");
    }
}
```

b.

```
#include<stdio.h>
void main()
{   int I, J, T;
    T = 0;
    for ( I=1; I<=3; I++)
        {   for(J = I; J<=5; J++)
            {   T = T + J;
                printf("%3i", T );
            }
            printf("\n");
        }
}
```





FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan
Jakarta Selatan, 12260
Telp: 021-5853753 Fax : 021-5853752
<http://fti.budiluhur.ac.id>