

Algoritma Pemrograman dan **Struktur Data**

Materi 2: OPERATOR DAN STRUKTUR DASAR ALGORITMA

Dosen pengampu:

Suamanda Ika Novichasari, M.Kom. Imam Adi Nata, S.Kom., M.Kom.









Learning Objective

Mahasiswa mampu menjelaskan jenis-jenis operator

Mahasiswa mampu menjelaskan Struktur Dasar Algoritma

Pre Test 10 menit

Jelaskan alur dari pseudocode berikut dan tentukan outputnya!

```
//PROGRAM Do while
// Mengimplementasikan kombinasi struktur dasar
algoritma
//DECLARATION
      DECLARE integer A, b, x
//IMPLEMENTATION / DEFINITION
      x ← 1
      Do {
           If (x \mod 2) = 0 then {
                  A \leftarrow A + 2
            }else{
                  b \leftarrow b + 1
            x \leftarrow x + 1
      \}while (x>5)
      Print (A,b)
```

Course Material



BAB MATERI



Operator

Subbab ini mempelajari jenis-jenis operator



- Operator adalah lambang operasi yang dipakai dalam perhitungan.
- Contoh:
 - 4 3, tanda merupakan operator sedangkan angka 4 dan 3 adalah operand. Operand dapat berupa tetapan (konstanta), peubah (variabel), atau hasil dari suatu fungsi.
- Terdapat beberapa jenis operator yang dipakai dalam algoritma yaitu sebagai berikut : Operator Aritmatika, Operator Relasional, Operator Logika, Operator String

Jenis Operator

Operator Aritmatika

- fungsi aritmetika
- + (menjumlah)
- - (mengurangkan)
- * (mengalikan)
- / (membagi)

Operator Relasional

- perbandingan antara dua operand
- > (lebih besar)
- < (lebih kecil)
- >= (lebih besar atau sama)
- <= (lebih kecil atau sama)
- == (sama)
- != (tidak sama) atau ><.

Operator Logika

- merelasikan operand secara logis
- && (and)
- || (or)
- •! (not)

Operator String

- memanipulasi string
- + (concatenation)
- Substr (substring, mencuplik).



BAB MATERI



Sekuensial

Subbab ini membahas deskripsi tentang struktur algoritma sekuensial

Sekuensial

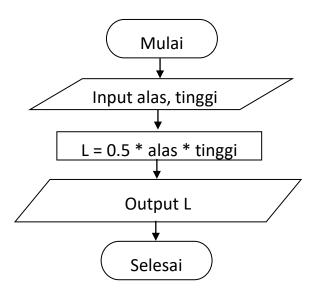
- Dalam struktur dasar sekuensial, algoritma dibangun dengan langkah-langkah (instruksi/perintah) dikerjakan secara berurutan, tidak boleh melompati satu langkah perintah pun.
- Oleh karena itu, pembuat algoritma harus mampu menganalisa dan menentukan intruksi mana yang harus ditulis lebih awal dan seterusnya dan yang mana harus paling akhir agar menghasilkan output yang sesuai.

Sekuensial

- Contoh, algoritma untuk menghitung luas segitiga sebagai berikut :
- Dengan untaian kalimat :
 - 1. Mulai
 - 2. Input nilai alas dan tinggi
 - 3. Masukkan kedalam rumus Luas = 0.5 * alas * tinggi
 - 4. Tampilkan hasil Luas
 - 5. Selesai



Contoh, algoritma untuk menghitung luas segitiga sebagai berikut :



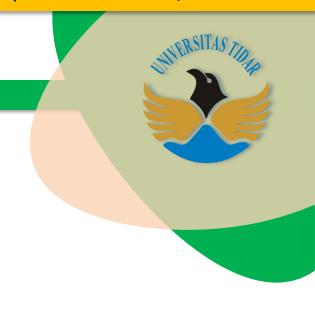


Contoh, algoritma untuk menghitung luas segitiga sebagai berikut :

```
//PROGRAM Luas Segitiga
// Algoritma untuk menghitung luas segitiga
//DECLARATION
    DECLARE
        integer a, t
        float L

//IMPLEMENTATION / DEFINITION
    INPUT (a,b)
    L ← 0.5 * a * t
    DISPLAY ("Luas segitiga = ", L)
```

BAB MATERI

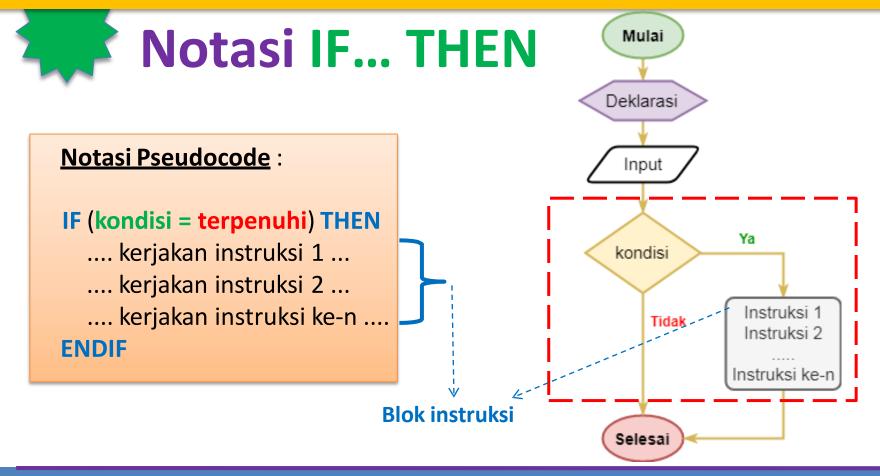


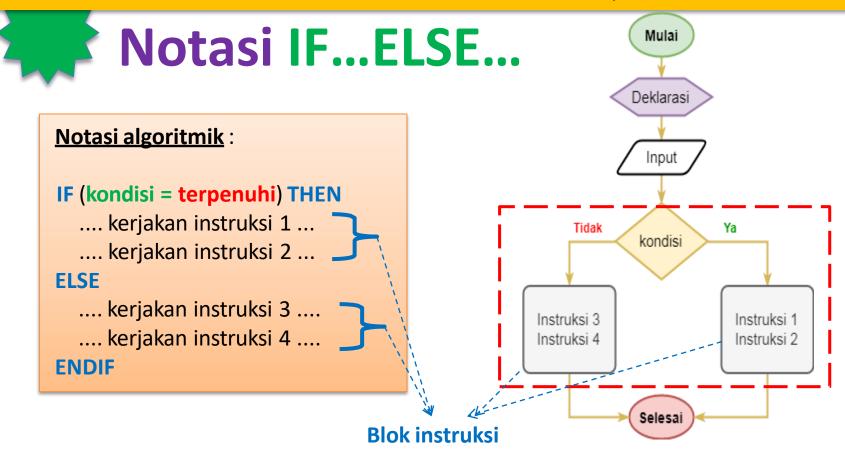
Seleksi

Subbab ini membahas tentang deskripsi struktur algoritma seleksi



- Pada struktur seleksi terdapat suatu **kondisi** yang **disyaratkan**, kemudian akan berpindah urutan pelaksanaan jika kondisi tersebut dipenuhi.
- Implementasi pada flowchart menggunakan **simbol Decision**, yang berisi pernyataan yang akan diuji kebenarannya.
- Nilai hasil pengujian akan menentukan cabang mana yang akan ditempuh.
- **Ciri** permasalahan yang menggunakan struktur seleksi ditandai dengan adanya pernyataan **jika ..maka**.







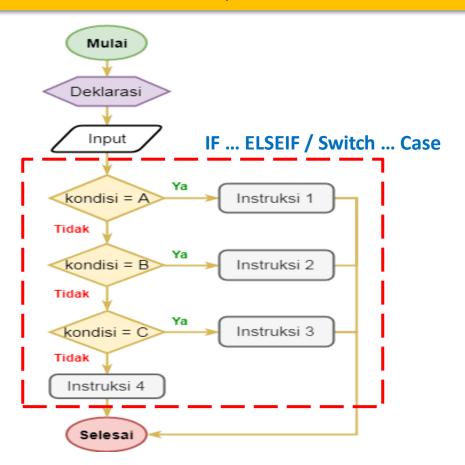
Notasi IF...ELSE IF ... dan SWITCH...CASE

```
Notasi algoritmik IF ... ELSEIF...:
IF (kondisi A terpenuhi) THEN
      .... kerjakan instruksi 1 ....
ELSEIF (kondisi B terpenuhi)
      .... kerjakan instruksi 2 ....
ELSEIF (kondisi C terpenuhi)
      .... kerjakan instruksi 3 ....
ELSE
      .... kerjakan instruksi 4 ....
ENDIF
```

```
Notasi algoritmik SWITCH ... CASE:
SWITCH (variabel)
      CASE A:
             .... kerjakan instruksi 1 ....
             BREAK:
      CASE B:
             .... kerjakan instruksi 2 ....
             BREAK;
      CASE C:
             .... kerjakan instruksi 3 ....
             BREAK;
      DEFAULT
             .... kerjakan instruksi 4 ....
             BREAK;
ENDSWITCH
```

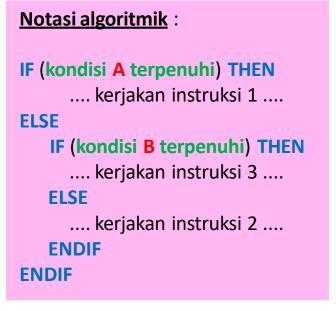
(lanjutan)

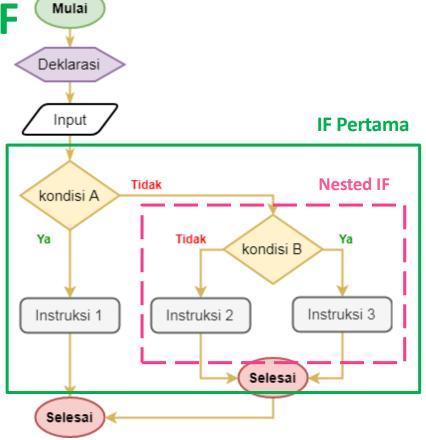
Notasi IF...ELSE IF ...
dan SWITCH...CASE





Notasi Nested IF





Beda IF ... ELSEIF & SWITCH ... CASE

Beda	IF ELSEIF	SWITCH CASE
Input kondisi	Numerik (bulat/pecahan)KarakterString	Numerik (bulat)KarakterString
Jumlah kondisi	lebih dari satu kondisi	Hanya satu kondisi
Operator	Operator relasionalOperator boolean	Tidak dapat menggunakan operator
Akhir blok instruk	Tidak ada	Break



- Contoh Algoritma untuk masalah batasan tinggi badan.
 - Untuk menaiki wahana permainan tertentu disyaratkan hanya untuk anak dengan tinggi badan lebih dari 100 cm. Jika tinggi badan anak kurang dari 100 cm, maka tidak diperbolehkan menaiki wahana permainan tersebut. Buatlah algoritma untuk permasalahan tersebut.



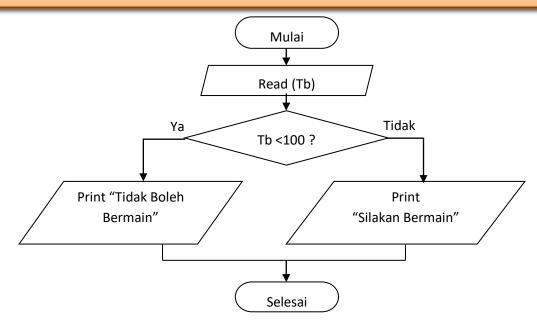
Penyelesaian:

Algoritma dengan untaian kalimat:

- 1. Mulai
- 2. Inputkan tinggi badan anak
- 3. Jika tinggi badan lebih dari 100 cm, maka cetak "Silakan Bermain". Jika kurang dari 100 cm, maka cetak "Tidak Boleh Bermain".
- 4. Selesai



• Penyelesaian dengan flowchart:





Penyelesaian dengan pseudocode :



Contoh: Penggunaan IF...ELSEIF...

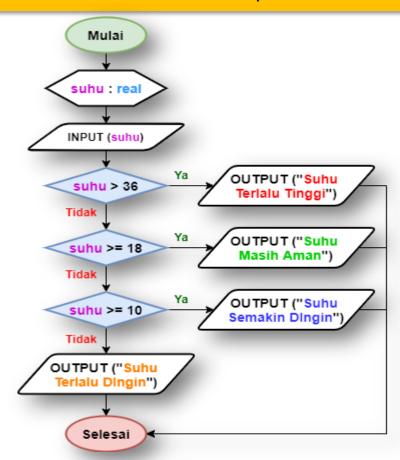
Buatlah algoritma untuk menampilkan peringatan suhu dengan ketentuan seperti tabel berikut:

Suhu	Peringatan
> 36°C	Suhu terlalu tinggi
18ºC sampai 36ºC	Suhu masih aman
10°C sampai 17,9°C	Suhu semakin dingin
< 10°C	Suhu terlalu dingin



Solusi: Notasi Flowchart

IF ... ELSEIF





Contoh: Penggunaan SWITCH...CASE

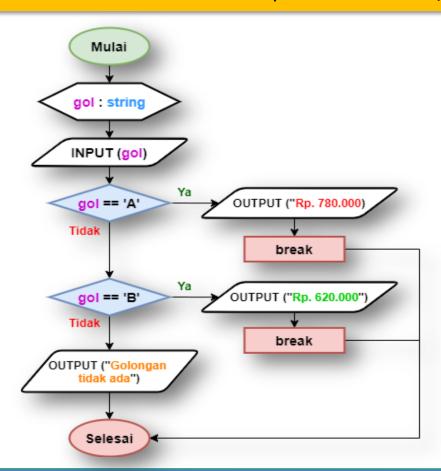
Buatlah algoritma untuk menentukan upah karyawan berdasarkan golongan. Berikut ini tabel golongan karyawan dan upahnya:

Karyawan	Upah
Golongan A	Rp. 780.000
Golongan B	Rp. 620.000



Solusi: Notasi Flowchart

SWITCH...CASE



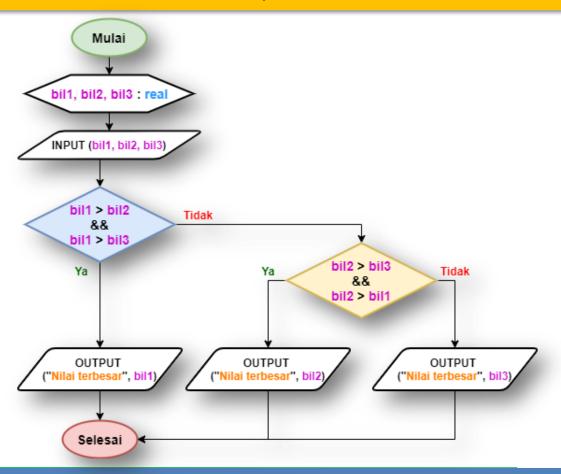


Contoh: Penggunaan Nested IF

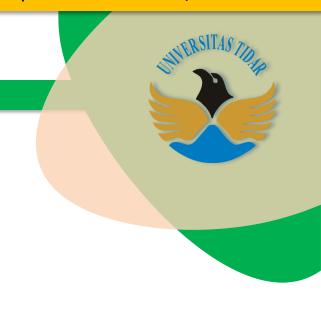
Buatlah algoritma untuk menentukan bilangan terbesar dari 3 buah bilangan!



Solusi: Notasi Flowchart Nested IF



BAB MATERI



Pengulangan

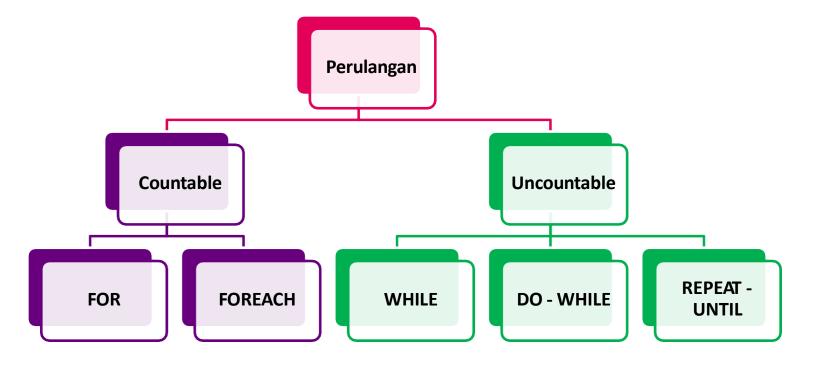
Subbab ini membahas mengenai pengulangan



Apa itu perulangan?

- Perulangan adalah instruksi yang diperintahkan untuk mengerjakan suatu tugas secara berulangulang berdasarkan kondisi tertentu.
- Perulangan = looping = iteratif.
- Perulangan akan terus dilakukan selama kondisi memenuhi syarat (bernilai true)

Hirarki Perulangan





Konsep WHILE ...

- Kondisi/syarat diperiksa terlebih dahulu, baru bisa menjalankan instruksi di dalam body perulangan.
- Perulangan tidak akan terjadi jika kondisi/syarat tidak terpenuhi, artinya ada kemungkinan while tidak menghasilkan output.
- While digunakan ketika jumlah perulangan tidak diketahui.



Konsep WHILE ...

- Kondisi/syarat diperiksa terlebih dahulu, baru bisa menjalankan instruksi di dalam body perulangan.
- Perulangan tidak akan terjadi jika kondisi/syarat tidak terpenuhi, artinya ada kemungkinan while tidak menghasilkan output.
- While digunakan ketika jumlah perulangan tidak diketahui.



```
Notasi algoritmik:

Insialisasi
WHILE (kondisi = true)
.... kerjakan instruksi 1 ...
.... kerjakan instruksi 2 ...
pencacah
END-WHILE
```



Flowchart: WHILE

Contoh

Hasil perulangan increment :

Cetak 1

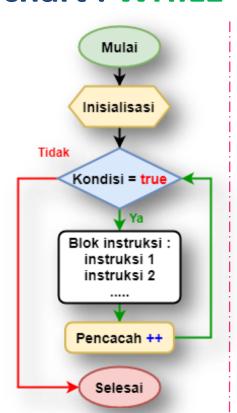
Cetak 2

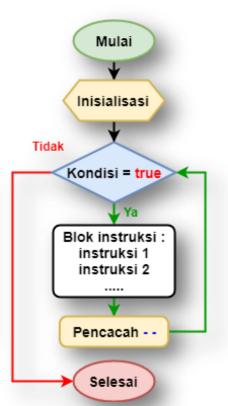
Cetak 3

Cetak 4

Cetak 5

Pencacah Increment





<u>Contoh</u>

Hasil perulangan decrement :

Cetak 5

Cetak 4

Cetak3

Cetak 2

Cetak 1

Pencacah decrement



Konsep DO ... WHILE

- Instruksi di dalam *body* perulangan dikerjakan terlebih dahulu, baru memeriksa kondisi/syarat perulangan.
- Instruksi di dalam body perulangan minimal dikerjakan sekali, jika kondisi/syarat perulangan tidak terpenuhi, artinya do...while pasti menghasilkan minimal 1 output.



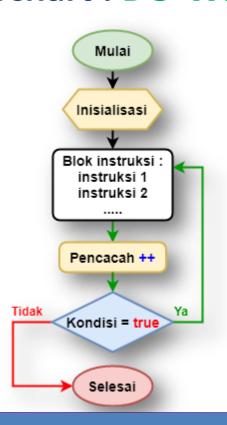
Notasi: DO ... WHILE

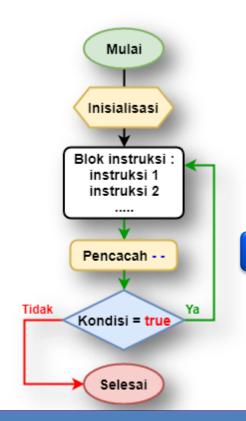
```
Notasi algoritmik:
inisialisasi
DO
  .... kerjakan instruksi 1 ...
  .... kerjakan instruksi 2 ...
  pencacah
WHILE (kondisi = true)
```



Pencacah Increment

Flowchart: DO-WHILE





Pencacah decrement



Konsep FOR ...

- FOR mirip WHILE
- Kondisi/syarat diperiksa terlebih dahulu, baru bisa menjalankan instruksi di dalam body perulangan.
- Perulangan tidak akan terjadi jika kondisi/syarat tidak terpenuhi, artinya ada kemungkinan while tidak menghasilkan output
- **Digunakan** = ketika jumlah perulangan **diketahui**.



Notasi: FOR ...

Notasi algoritmik:

FOR (nilai_awal;nilai_akhir;pencacah)

.... kerjakan instruksi 1 ...

.... kerjakan instruksi 2 ...

ENDFOR

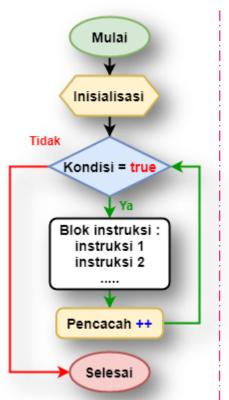


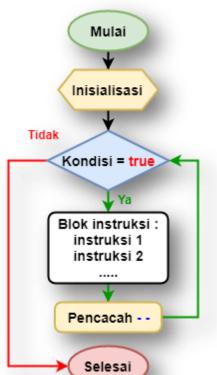
Pencacah Increment

Flowchart: For

Catatan:

Flowchart untuk FOR sama seperti WHILE

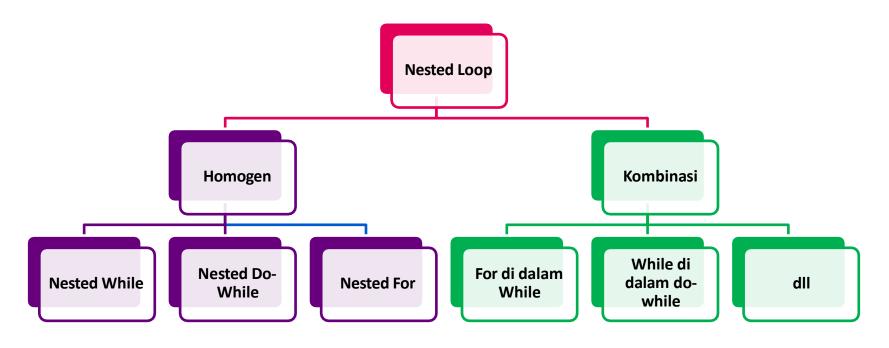




Pencacah decrement



Jenis-jenis perulangan bersarang





Notasi umum: NESTED-LOOP

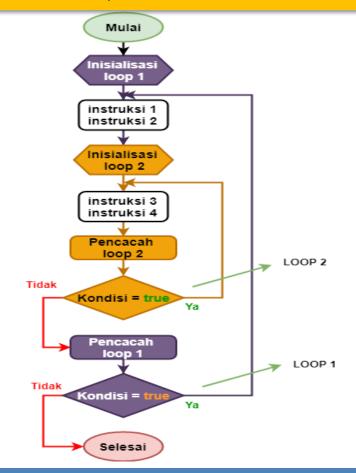
```
Loop 1
   // Blok instruksi (body) loop 1
    ... kerjakan instruksi 1 ...
    ... kerjakan instruksi 2 ...
    Loop 2
       // Blok instruksi (body) loop 2
       ... kerjakan instruksi 3 ...
       ... kerjakan instruksi 4 ...
```

- Bentuk sederhana dari nested loop terdiri dari dua buah loop.
- Jumlah loop yang bersarang tidak ada ketentuan yang membatasinya.
- bersarang di dalam *loop*, maka algoritma tersebut akan **semakin kompleks**.



Nested Do-While

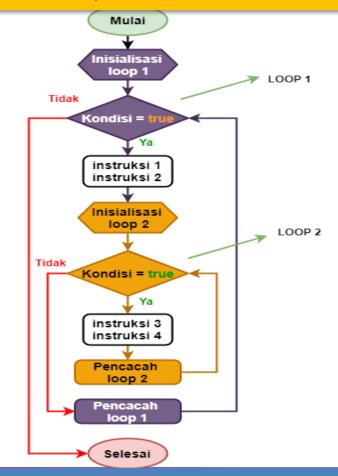
```
inisialisasi loop 1
Do {
   // Blok instruksi (body) while 1
   ... kerjakan instruksi 1 ...
   ... kerjakan instruksi 2 ...
   inisialisasi loop 2
   Do {
      // Blok instruksi (body) while 2
      ... kerjakan instruksi 3 ...
      ... kerjakan instruksi 4 ...
      pencacah 2
   } While (kondisi2 == true)
   pencacah 1
} While (kondisi1 == true)
```





Nested While

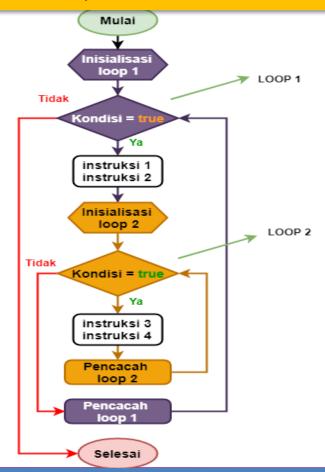
```
inisialisasi 1
While (kondisi1 == true)
   // Blok instruksi (body) while 1
   ... kerjakan instruksi 1 ...
   ... kerjakan instruksi 2 ...
   inisialisasi 2
   While (kondisi2 == true)
      // Blok instruksi (body) while 2
      ... kerjakan instruksi 3 ...
      ... kerjakan instruksi 4 ...
      pencacah 2
   pencacah 1
```





Nested FOR

```
FOR (nilai_awal;nilai_akhir;pencacah 1)
    // Blok instruksi (body) for 1
    .... kerjakan instruksi 1 ...
    .... kerjakan instruksi 2 ...
    FOR (nilai_awal;nilai_akhir;pencacah 2)
         // Blok instruksi (body) for 1
         .... kerjakan instruksi 3 ...
         .... kerjakan instruksi 4 ...
```





Nested Kombinasi While dan For

```
inisialisasi loop 1
While (kondisi1 == true)
   // Blok instruksi (body) while
   ... kerjakan instruksi 1 ...
    ... kerjakan instruksi 2 ...
   FOR (nilai_awal;nilai_akhir;pencacah 2)
           // Blok instruksi (body) for
         .... kerjakan instruksi 3 ...
         .... kerjakan instruksi 4 ...
   pencacah 1
```

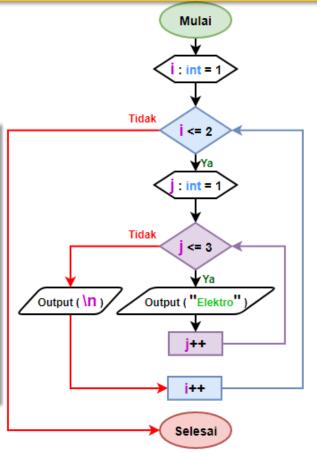
Contoh 1

Tabel Pelacakan nilai looping:

i	i≤2	J	j≤3	Output	j++	i++
1	True	1	True	Elektro 2		
		2	True	Elektro	3	
		3	True	Elektro	4	
		4	False	\n		2
2	True	1	True	Elektro 2		
		2	True	Elektro	3	
		3	True	Elektro	4	
		4	False	\n		3
3	False	Selesai				

Pseudocode – Nested While

```
Algoritma Untuk Cetak Data
Deklarasi :
   i : integer
    j : intger
Algoritma :
    2. WHILE (i <= 2)
         WHILE (j <= 3)
             OUTPUT ("Elektro")
             j++
    ENDWHILE 2
    8. OUTPUT("\n")
         1++
    10.ENDWHILE 1
```



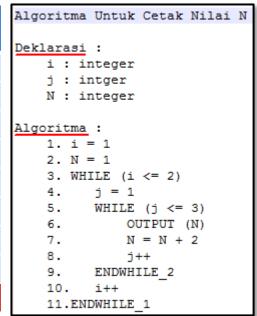


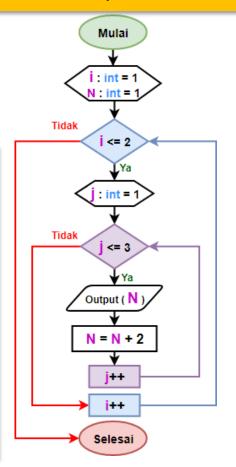
Contoh 2

Tabel Pelacakan nilai looping:

i	N	i≤2	J	j ≤ 3	Output (N)	N=N+2	j++	i++
1	1	True	1	True	1	3	2	
	3		2	True	3	5	3	
	5		3	True	5	7	4	
			4	False	keluar loop		2	
2	7	True	1	True	7	9	2	
	9		2	True	9	11	3	
	11		3	True	11	13	4	
			4	False	Keluar loop		3	
3	13	False	Selesai					

Pseudocode - Nested While





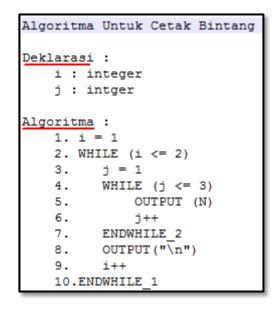


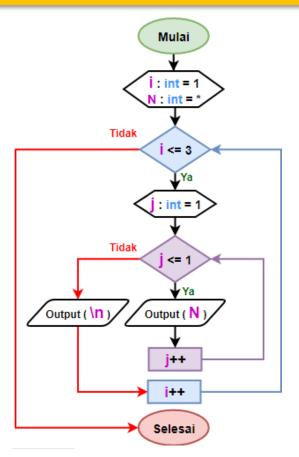
Contoh 3

Tabel Pelacakan nilai looping:

i	i≤3	j	j≤i	Output (N)	j++	i++
1	True	1	True	*	2	
		2	False	\n		2
2	True	1	True	*	2	
		2	True	*	3	
		3	False	\n		3
3	True	1	True	*	2	
		2	True	*	3	
		3	True	*	4	
		4	False	\n		3
3	False			Selesai		

Pseudocode - Nested While





Post Test 10 menit

Jelaskan alur dari pseudocode berikut dan tentukan outputnya!

```
//PROGRAM Do while
// Mengimplementasikan kombinasi struktur dasar
algoritma
//DECLARATION
      DECLARE integer A, b, x
//IMPLEMENTATION / DEFINITION
      A ← 1
      x ← 1
      Do {
           If (x \mod 2) = 0 then {
                  A \leftarrow A + 2
            }else{
                  b \leftarrow b + 1
            x \leftarrow x + 1
      \}while (x>5)
      Print (A,b)
```

Tugas!

Buatlah pseudocode untuk menampilkan segitiga bintang sama sisi seperti berikut :

