

**ANALISA ALGORITMA DENGAN TABEL KEBENARAN OUTPUT PADA LOGIKA LOOPING  
DALAM PROGRAM APLIKASI BAHASA PEMROGRAMAN PASCAL VERSI 7.0**

Oleh: Rini Nuraini 2008

**ABSTRACT**

*Repetition is considered a tiring and boring activity for human. Thus is not so for a machine. One of the superiorities of this machine called computer over human being is the ability to execute repeated commands without getting tired. The repetition or loop can run until it reaches its set condition. Logic is a form of reasoning. Algorithm is a series of procedures or instructions to solve a problem. Analysis is a study on a certain language to examine carefully the language structure by breaking it up into smaller parts that easier to learn, and therefore to draw a conclusion. Pascal is one of computer language programs. Language program is instructions or commands that are understood by the computer to execute certain tasks. These tasks are the application programs, which are designed by using one of the computer language programs. Visualization helps define a concept. Therefore, the writer uses flowcharts to visualize the concept. Flowchart is a series of symbols used to explain either the logic or the undergoing program.*

Untuk melakukan hal berulang-ulang bagi manusia merupakan hal yang melelahkan, dan membosankan. Lain halnya bila sebuah mesin yang menjalankannya. Itulah salah satu kelebihan mesin yang diberi nama komputer itu dibandingkan dengan manusia adalah kemampuannya untuk melaksanakan suatu instruksi berulang kali tanpa mengenal lelah dan bosan. Pengulangan atau kalang (repetition atau loop) dapat dilakukan sejumlah kali, atau sampai kondisi berhenti pengulangan tercapai. Logika adalah suatu bentuk penalaran atau pemikiran, sedangkan Algoritma adalah suatu urutan dari barisan langkah-langkah atau instruksi untuk menyelesaikan suatu masalah. Analisis atau analisa adalah kajian yang dilaksanakan terhadap sebuah bahasa guna meneliti struktur bahasa tersebut secara mendalam dengan cara memecah-mecah bahasa tersebut menjadi bagian-bagian kecil yang lebih mudah dipelajari, kemudian mempelajari bagian-bagian kecil tersebut, lalu mengambil kesimpulannya. Pascal adalah salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Tugas-tugas tertentu ini berupa program aplikasi yang dibentuk dengan menggunakan salah satu bahasa pemrograman komputer. Dengan memvisualisasikan sesuatu dengan sebuah simbol dari apa yang ada dalam fikir maka akan memudahkan untuk menjelaskan sesuatu tersebut. Dalam hal ini, penulis menggunakan simbol flowchart untuk mevisualisasikan sesuatu tersebut. Flowchart adalah urutan dari simbol-simbol yang digunakan untuk menjelaskan logika program atau jalannya program.

**I. PENDAHULUAN**

Perulangan (*loop*) merupakan bentuk yang sering ditemui di dalam suatu program aplikasi, misalkan mencetak tulisan 'BSI' sebanyak 10 kali; mencetak angka 1, 2, 3, ..., 20 di layar, dengan satu angka pada setiap baris; mencetak angka 0, 1, 2, ..., N, yang dalam hal ini N dibaca dari piranti masukan; menghitung

jumlah angka dari 1 sampai N, nilai N dibaca dari keyboard, misalnya  $N = 5$ , maka  $1 + 2 + 3 + 4 + 5 = 15$ ; menghitung rata-rata dari sejumlah bilangan bulat, data bilangan dibaca dari keyboard, nilai rata-rata adalah jumlah seluruh bilangan dibagi dengan banyak bilangan, misalkan ada 5 buah data bilangan, yaitu 12, 10, 6, 2, 4, maka rata-

ratanya adalah  $(12 + 10 + 6 + 2 + 4) / 5 = 34 / 5 = 6.8$ ; menghitung jumlah sekumpulan bilangan bulat positif yang dibaca dari *keyboard*; memilih menu secara berulang; menghitung upah karyawan berulang; dan lain sebagainya.

Dalam pembahasan ini, sebagai contoh, penulis membuat program aplikasi untuk mengulang bilangan 1 sampai 6 dengan menggunakan *statemen while-do, repeat-until, dan for-do*; program aplikasi menjumlah bilangan 1 sampai 5 dengan *statemen for-do*; dan program aplikasi memproses kolom baris dengan *statemen for-do nested (for-do bersarang)*. Dalam contoh angka-angka yang digunakan angka-angka sederhana, angka satu ditambah dua, dua ditambah tiga, dan seterusnya hanya sampai hitungan sepuluh, sebagian orang tentu mampu melakukannya. Tetapi yang perlu untuk diperhatikan di sini adalah bagaimana logikanya komputer dalam hal melakukan perhitungan tersebut.

Begitu pula jumlah perulangannya pun hanya terbatas sampai lima atau enam kali *looping*. Dalam hal ini, hanya untuk memudahkan menjelaskan saja. Untuk program yang besar dan kompleks pun, dapat menggunakan metode ini untuk menganalisanya.

Untuk menulis struktur algoritma dari pengulangan (logika *looping*) terdiri dari dua bagian, yaitu:

- a. Kondisi pengulangan.  
Yaitu ekspresi *boolean* (bernilai *true* atau *false*) yang harus dipenuhi untuk melaksanakan pengulangan.
- b. Badan (*body*) pengulangan.  
Yaitu satu atau lebih aksi yang akan diulang.

Selain dua hal di atas, struktur logika *looping* juga disertai dengan bagian:

- a. Inisialisasi.  
Yaitu aksi yang dilakukan sebelum pengulangan dilakukan pertama kali.
- b. Terminasi.

Yaitu aksi yang dilakukan setelah pengulangan selesai dilaksanakan. Inisialisasi dan terminasi tidak selalu harus ada, namun pada berbagai kasus inisialisasi umumnya diperlukan.

Dalam pembahasan ini, ada tiga macam notasi algoritma dari logika *looping*, yaitu:

### 1. Struktur WHILE-DO

Bentuk penulisan: **while** <kondisi>  
aksi  
**endwhile**

Dengan logika:

Aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> *boolean* masih tetap bernilai *true*. Jika <kondisi> bernilai *false*, badan (*body*) pengulangan tidak akan dilaksanakan, atau pengulangan selesai.

### 2. Struktur REPEAT-UNTIL

Bentuk umum penulisan: **repeat**  
aksi  
**until** <kondisi>

Dengan logika:

Aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> *boolean* masih tetap bernilai *false*. Jika <kondisi> bernilai *true*, badan (*body*) pengulangan tidak akan dilaksanakan, atau pengulangan selesai.

### 3. Struktur FOR-DO

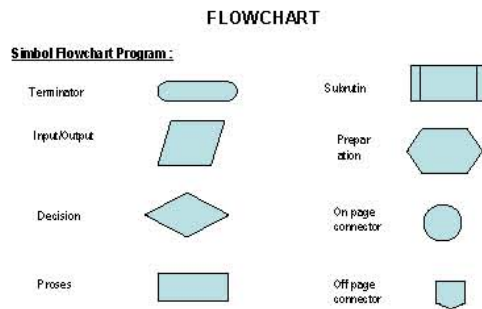
Bentuk umum penulisan : **for** peubah  
← nilai\_awal to nilai akhir **do**  
aksi  
**endfor**

Dengan logika:

Pengulangan akan dilakukan hingga batas tertentu, atau berulang sebanyak nilai yang sudah ditentukan pada nilai\_awal hingga nilai\_akhir. Peubah (variabel) diinisialisasi dengan nilai\_awal. Nilai peubah (variabel) secara otomatis bertambah satu setiap kali aksi pengulangan dilakukan, sampai akhirnya nilai

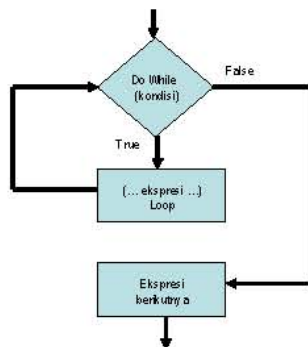
peubah (variabel) sama dengan nilai\_akhir.

Untuk memudahkan memahami logika looping, dibutuhkan alat bantu berupa *flowchart*. Untuk menggambarkan logika dari program dibutuhkan simbol-simbol yang sudah terstandarisasi, supaya bisa saling memahami antara si *programmer* yang menggambarkan logika program dan user pembaca logika program tersebut. Adapun simbol-simbol tersebut diberi nama *Flowchart*, berikut adalah simbol dari program flowchart:

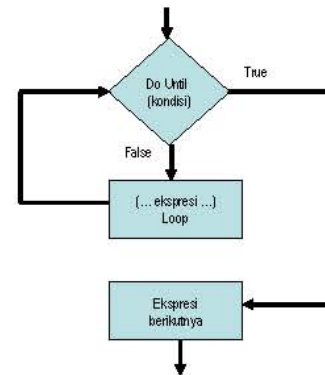


Berikut adalah *flowchart* dari masing-masing logika looping:

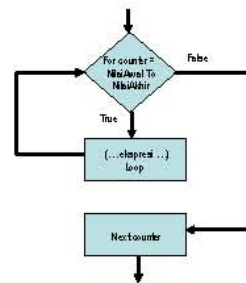
### 1. Flowchart WHILE-DO



### 2. Flowchart REPEAT-UNTIL



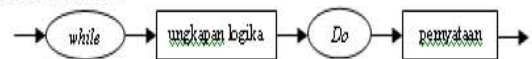
### 3. Flowchart FOR-DO



Di dalam bahasa pemrograman Pascal, dikenal tiga macam perulangan, yaitu dengan menggunakan *statement While-Do*, *Repeat-Until*, dan *For-Do*. Untuk penulisan logika *looping* dalam notasi bahasa pemrograman Pascal, adalah sebagai berikut:

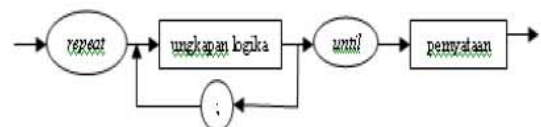
#### 1. Perulangan WHILE-DO

Statement :

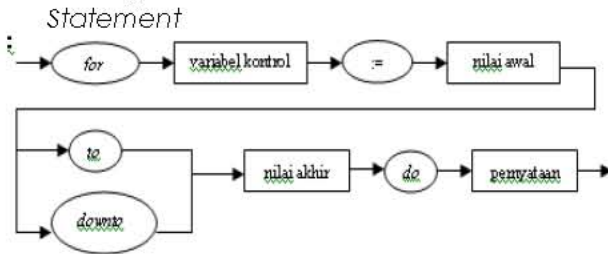


#### 2. Perulangan REPEAT-UNTIL

Statement :



3. Perulangan FOR-DO



II. PEMBAHASAN

2.1. PROGRAM APLIKASI: MENGULANG BILANGAN 6 sampai dengan 10.

2.1.1. PROGRAM PASCAL : Mengulang Bilangan 6 sampai dengan 10, Dengan DO-WHILE (WHILED01.PAS)

Program:  
Mengulang\_bilangan\_dengan\_while\_do;

```
uses crt;
var
angka : integer;
begin
clrscr;
angka := 5;
while angka < 10 do
begin
angka := angka + 1;
writeln(angka);
end;
readln;
end.
```

Tabel II.1. Tabel Kebenaran Output Analisa Algoritma Aplikasi Program Mengulang Bilangan 6 sampai dengan 10 dengan WHILE-DO (WHILED01.PAS)

Angka := 5	While Angka < 10 do	Hasil Nilai Boolean	Angka := angka + 1	Writeln(angka)
5	5 < 10	T	6 := 5 + 1	6
	6 < 10	T	7 := 6 + 1	7
	7 < 10	T	8 := 7 + 1	8
	8 < 10	T	9 := 8 + 1	9
	9 < 10	T	10 := 9 + 1	10
	10 < 10	F	-	-

Analisa algoritma:

Logika *statement while-do* adalah aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> boolean masih tetap bernilai *True*. Jika <kondisi> bernilai *False* badan (*body*) pengulangan tidak akan dilaksanakan, atau pengulangan selesai. Dalam program Pascal Badan (*body*) pengulangan yaitu intruksi diantara *begin* sampai *end* setelah *while-do*.

Jika nilai awal yang dimasukkan ke dalam program aplikasi tersebut adalah angka lima (5) dengan instruksi angka:=5; selanjutnya terjadi proses pengujian pada variabel angka tersebut yang berisi angka 5, apakah 5 < 10, hasil nilai boolean *true* maka masuk ke badan (*body*) looping mengerjakan intruksi yang berada di antara *begin-end* yaitu angka = angka + 1; berarti 6 = 5 + 1, dengan demikian variabel angka berubah menjadi angka enam (6). Dengan intruksi *writeln(angka)*; maka tercetak 6.

Sekarang variabel angka sudah menjadi angka enam (6) berikutnya dilakukan pengujian lagi apakah 6 < 10, hasil nilai boolean *true* maka masuk ke badan (*body*) looping lagi yaitu mengerjakan intruksi angka = angka + 1; berarti 7 = 6 + 1, dengan demikian variabel angka berubah menjadi tujuh (7). Dengan intruksi *writeln(angka)*; maka tercetak 7.

Sekarang variabel angka sudah menjadi tujuh (7) berikutnya dilakukan pengujian lagi apakah 7 < 10, hasil nilai boolean *true* maka masuk ke badan (*body*) looping lagi yaitu mengerjakan intruksi angka = angka + 1; berarti 8 = 7 + 1, dengan demikian variabel angka berubah menjadi delapan (8). Dengan intruksi *writeln(angka)*; maka tercetak 8.

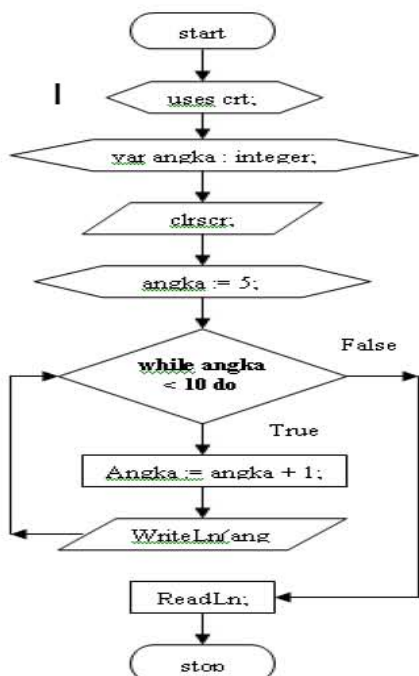
Sekarang variabel angka sudah menjadi delapan (8) berikutnya dilakukan pengujian lagi apakah 8 < 10, hasil nilai boolean *true* maka masuk ke badan (*body*) looping yaitu mengerjakan intruksi angka = angka + 1; berarti 9 = 8 +

1, dengan demikian variabel angka berubah menjadi sembilan (9). Dengan intruksi `writeln(angka);` maka tercetak 9.

Sekarang variabel angka sudah menjadi sembilan (9) berikutnya dilakukan pengujian lagi apakah  $9 < 10$ , hasil nilai *boolean true* maka masuk ke badan (*body*) looping yaitu mengerjakan intruksi `angka = angka + 1;` berarti  $10 = 9 + 1$ , dengan demikian variabel angka berubah menjadi 10. Dengan intruksi `writeln(angka);` maka tercetak 10.

Berikutnya adalah apakah  $10 < 10$  hasil nilai *boolean false*, maka proses pengulangan dihentikan. Karena logika statement *Repeat-Until* adalah aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> *boolean* masih tetap bernilai *true*. Jika <kondisi> bernilai *False* badan (*body*) pengulangan tidak akan dilaksanakan, atau pengulangan selesai.

**Flowchart:**



**1.2.2. PROGRAM PASCAL : Mengulang Bilangan 6 sampai dengan 10,**

**Dengan REPEAT-UNTIL (RPUNTIL.PAS)**

**Program\_Mengulang\_bilangan\_dengan\_repeat\_until;**

```

uses crt;
var
angka : integer;
begin
clrscr;
angka := 5;
repeat
begin
angka := angka + 1;
writeln(angka);
end;
until angka >= 10;
readln;
end.
    
```

**Tabel II.2. Tabel Kebenaran Output Analisa Algoritma Aplikasi Program Mengulang Bilangan 6 sampai dengan 10 dengan REPEAT-UNTIL (RPUNTIL.PAS)**

Angka := 5	Repeat	Angka := angka + 1	Writeln (angka)	until angka >= 10	Hasil Nilai Boolean
5		6 := 5 + 1	6	6 >= 10	F
		7 := 6 + 1	7	7 >= 10	F
		8 := 7 + 1	8	8 >= 10	F
		9 := 8 + 1	9	9 >= 10	F
		10 := 9 + 1	10	10 >= 10	T

**Analisa algoritma:**

Logika statement *Repeat-Until* adalah aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> *boolean* masih tetap bernilai **False**. Jika <kondisi> bernilai **True**, badan (*body*) pengulangan tidak akan dilaksanakan, atau pengulangan selesai. Dalam program Pascal Badan (*body*) pengulangan yaitu intruksi diantara *repeat* sampai *until*.

Jika nilai awal yang dimasukkan ke dalam program aplikasi tersebut adalah angka lima (5) dengan intruksi `angka:=5;`

selanjutnya terjadi proses pengujian pada variabel angka tersebut yang berisi angka 5, apakah  $5 \geq 10$ , hasil nilai *boolean false* maka masuk ke badan (*body*) *looping* mengerjakan intruksi yang berada di antara *repeat-until* yaitu angka = angka + 1; berarti  $6 = 5 + 1$ , dengan demikian variabel angka berubah menjadi angka enam (6). Dengan intruksi *writeln(angka)*; maka tercetak 6.

Sekarang variabel angka sudah menjadi angka enam (6) berikutnya dilakukan pengujian lagi apakah  $6 \geq 10$ , hasil nilai *boolean false* maka masuk ke badan (*body*) *looping* lagi yaitu mengerjakan intruksi angka = angka + 1; berarti  $7 = 6 + 1$ , dengan demikian variabel angka berubah menjadi tujuh (7). Dengan intruksi *writeln(angka)*; maka tercetak 7.

Sekarang variabel angka sudah menjadi tujuh (7) berikutnya dilakukan pengujian lagi apakah  $7 \geq 10$ , hasil nilai *boolean false* maka masuk ke badan (*body*) *looping* lagi yaitu mengerjakan intruksi angka = angka + 1; berarti  $8 = 7 + 1$ , dengan demikian variabel angka berubah menjadi delapan (8). Dengan intruksi *writeln(angka)*; maka tercetak 8.

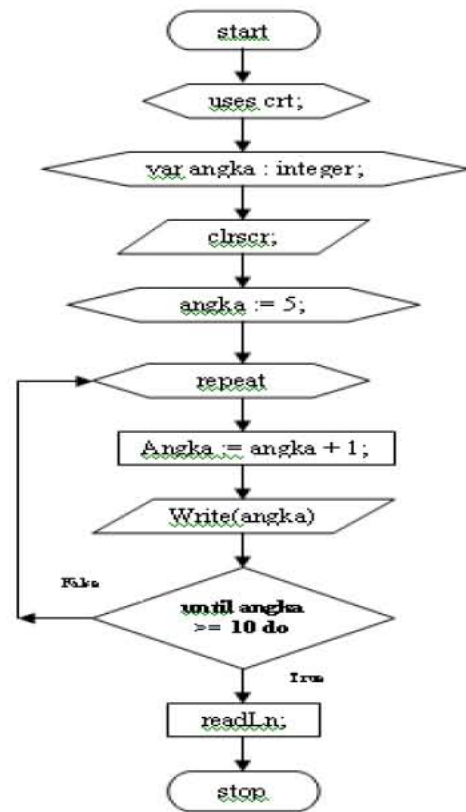
Sekarang variabel angka sudah menjadi delapan (8) berikutnya dilakukan pengujian lagi apakah  $8 \geq 10$ , hasil nilai *boolean false* maka masuk ke badan (*body*) *looping* yaitu mengerjakan intruksi angka = angka + 1; berarti  $9 = 8 + 1$ , dengan demikian variabel angka berubah menjadi sembilan (9). Dengan intruksi *writeln(angka)*; maka tercetak 9.

Sekarang variabel angka sudah menjadi sembilan (9) berikutnya dilakukan pengujian lagi apakah  $9 \geq 10$ , hasil nilai *boolean false* maka masuk ke badan (*body*) *looping* yaitu mengerjakan intruksi angka = angka + 1; berarti  $10 = 9 + 1$ , dengan demikian variabel angka berubah menjadi 10. Dengan intruksi *writeln(angka)*; maka tercetak 10.

Berikutnya adalah apakah  $10 \geq 10$  hasil nilai *boolean true*, maka proses

pengulangan dihentikan. Karena logika *statement Repeat-Until* adalah aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> *boolean* masih tetap bernilai **False**. jika <kondisi> bernilai **True**, badan (*body*) pengulangan tidak akan dilaksanakan, atau pengulangan selesai.

**Flowchart:**



**2.1.3. PROGRAM PASCAL : Mengulang Bilangan 6 sampai dengan 10, Dengan FOR-DO (FORDO1.PAS)**

**Program Menghitung\_bilangan\_dengan\_for\_do;**

```

uses crt;
var
angka := integer;
begin
    
```

```

clrscr;
for angka := 6 to 10 do
    writeln(angka);
readln;
end.
    
```

**Tabel II.3. Tabel Kebenaran Output Analisa Algoritma Aplikasi Program Mengulang Bilangan 6 sampai dengan 10 dengan FOR-DO (FORDO1.PAS)**

Angka	Proses Otomatis (tidak ada instruksi program)	for angka := 6 to 10 do	Hasil Nilai Boolean	Writeln (angka)
6	6	Apakah 6 diantara 6 s.d. 10?	T	6
	7 := 6 + 1	Apakah 7 diantara 6 s.d. 10?	T	7
	8 := 7 + 1	Apakah 8 diantara 6 s.d. 10?	T	8
	9 := 8 + 1	Apakah 9 diantara 6 s.d. 10?	T	9
	10 := 9 + 1	Apakah 10 diantara 6 s.d. 10?	T	10
	11 := 10 + 1	Apakah 11 diantara 6 s.d. 10?	F	.

**Analisa algoritma:**

Pengulangan akan dilakukan hingga batas tertentu, atau berulang sebanyak nilai yang sudah ditentukan pada nilai\_awal hingga nilai\_akhir. Peubah (variabel) diinisialisasi dengan nilai\_awal. Nilai peubah (variabel) secara otomatis bertambah satu setiap kali aksi pengulangan dilakukan, sampai akhirnya nilai peubah (variabel) sama dengan nilai\_akhir.

Logika *statement for-do* untuk menentukan nilai *boolean* sama dengan logika *statement while-do* yaitu aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> *boolean* masih tetap bernilai **True**. Jika <kondisi> bernilai *False* badan (*body*) pengulangan tidak akan dilaksanakan, atau pengulangan selesai. Dalam program Pascal Badan (*body*) pengulangan yaitu intruksi diantara *begin* sampai *end* setelah *for-do*. Bila

tidak ada intruksi *begin-end* berarti hanya satu instruksi setelah *for-do*.

Instruksi *for angka := 6 to 10 do* berarti nilai awal dari variabel angka adalah 6, selanjutnya terjadi proses pengujian pada variabel angka tersebut, apakah nilai angka 6 diantara nilai angka 6 sampai 10 ? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi *writeln(angka)*; maka tercetak 6

Secara otomatis, Instruksi *for angka := 6 to 10 do* akan menambah nilai satu pada variabel angka, dengan demikian variabel angka sekarang adalah 7, selanjutnya terjadi proses pengujian pada variabel angka tersebut, apakah nilai angka 7 diantara nilai angka 6 sampai 10? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi *writeln(angka)*; maka tercetak 7.

Secara otomatis, Instruksi *for angka := 6 to 10 do* akan menambah nilai satu pada variabel sebelumnya, dengan demikian variabel angka sekarang adalah 8, selanjutnya terjadi proses pengujian pada variabel angka tersebut, apakah nilai angka 8 diantara nilai angka 6 sampai 10 ? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi *writeln(angka)*; maka tercetak 8.

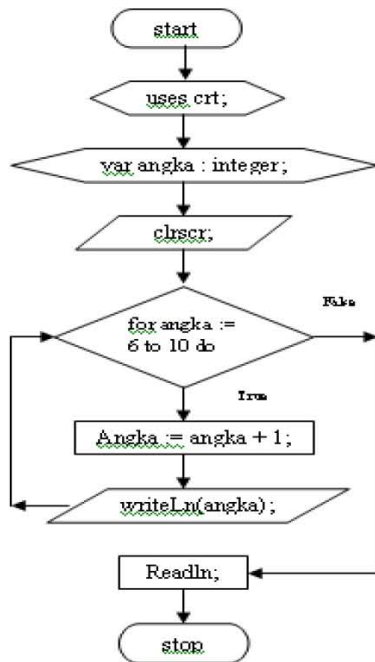
Secara otomatis, Instruksi *for angka := 6 to 10 do* akan menambah nilai satu pada variabel sebelumnya, dengan demikian variabel angka sekarang adalah 9, selanjutnya terjadi proses pengujian pada variabel angka tersebut, apakah nilai angka 9 diantara nilai angka 6 sampai 10 ? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi *writeln(angka)*; maka tercetak 9.

Secara otomatis, Instruksi *for angka := 6 to 10 do* akan menambah nilai satu pada variabel sebelumnya, dengan demikian variabel angka sekarang adalah 10, selanjutnya terjadi proses pengujian pada variabel angka tersebut,

apakah nilai angka 10 diantara nilai angka 6 sampai 10 ? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi `writeln(angka);` maka tercetak 10.

Secara otomatis, Instruksi `for angka := 6 to 10 do` akan menambah nilai satu pada variabel sebelumnya, dengan demikian variabel angka sekarang adalah 11, selanjutnya terjadi proses pengujian pada variabel angka tersebut, apakah nilai angka 11 diantara nilai angka 6 sampai 10 ? Bila hasil nilai *boolean false* maka proses pengulangan dihentikan. Karena logika *statement for-do* adalah aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> *boolean* masih tetap bernilai **true**. Jika <kondisi> bernilai *false* badan (*body*) pengulangan tidak akan dilaksanakan, atau pengulangan selesai.

**Flowchart :**



**2.2. PROGRAM APLIKASI: MENJUMLAH BILANGAN 1 sampai dengan 5 DENGAN FOR-DO (FORDO2.PAS)**

Program\_Mejumlah\_bilangan\_dengan\_for\_do;

```

uses crt;
var
angka, jumlah : integer;
begin
clrscr;
jumlah := 0;
for angka := 1 to 5 do
    jumlah := jumlah + angka;
writeln(jumlah);
readln;
end.
    
```

**Tabel II.1. Tabel Kebenaran Output Analisa Algoritma Aplikasi Program Menjumlah Bilangan 1 sampai dengan 5 dengan FOR-DO (FORDO2.PAS)**

angka	Proses Otomatis (tidak ada instruksi program)	for angka := 1 to 5 do	Hasil Nilai Boolean	Jumlah := jumlah + angka	WriteLn (angka)
1	1	Apakah 1 diantara 1 s.d. 5?	T	1 := 0 + 1	
2	2 := 1 + 1	Apakah 2 diantara 1 s.d. 5?	T	3 := 1 + 2	
3	3 := 2 + 1	Apakah 3 diantara 1 s.d. 5?	T	6 := 3 + 3	
4	4 := 3 + 1	Apakah 4 diantara 1 s.d. 5?	T	10 := 6 + 4	
5	5 := 4 + 1	Apakah 5 diantara 1 s.d. 5?	T	15 := 10 + 5	
6	6 := 5 + 1	Apakah 6 diantara 1 s.d. 5?	F	.	15

**Analisa algoritma:**

Instruksi `for angka := 1 to 5 do` berarti nilai awal dari variabel angka adalah 1, selanjutnya terjadi proses pengujian pada variabel angka tersebut, apakah nilai angka 1 diantara nilai angka 1 sampai 5 ? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi `jumlah := jumlah + angka;` menghasilkan proses 1 := 0 + 1, berarti variabel jumlah bernilai 1.

Secara otomatis, Instruksi `for angka := 1 to 5 do` akan menambah nilai satu pada variabel angka, dengan demikian variabel angka sekarang adalah 2, selanjutnya terjadi proses pengujian



pada variabel angka tersebut, apakah nilai angka 2 diantara nilai angka 1 sampai 5 ? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi jumlah := jumlah + angka; menghasilkan proses 3 := 1 + 2, variabel jumlah sekarang bernilai 3.

Secara otomatis, Instruksi *for* angka := 1 to 5 *do* akan menambah nilai satu pada variabel angka, dengan demikian variabel angka sekarang adalah 3, selanjutnya terjadi proses pengujian pada variabel angka tersebut, apakah nilai angka 3 diantara nilai angka 1 sampai 5 ? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi jumlah := jumlah + angka; menghasilkan proses 3 := 3 + 3, variabel jumlah sekarang bernilai 6.

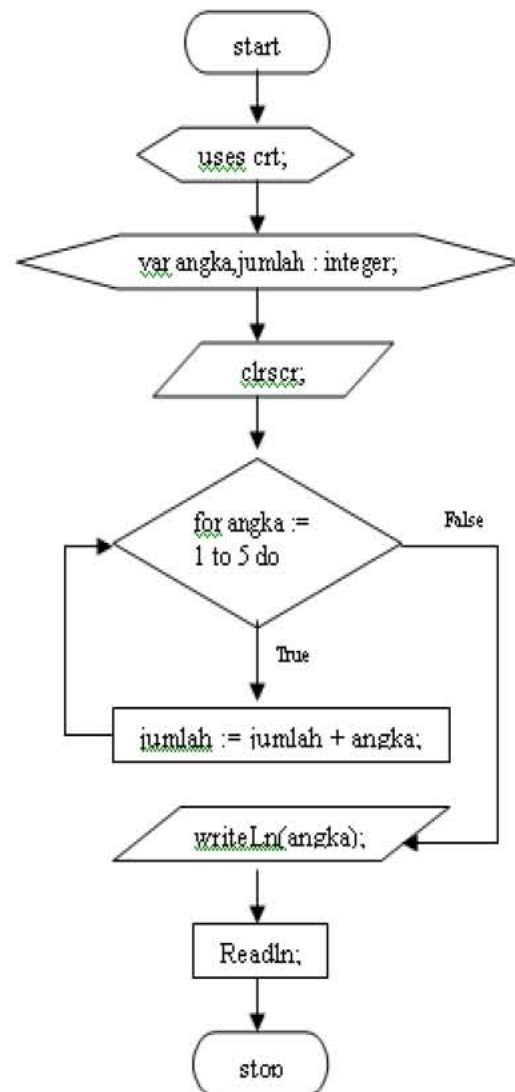
Secara otomatis, Instruksi *for* angka := 1 to 5 *do* akan menambah nilai satu pada variabel angka, dengan demikian variabel angka sekarang adalah 4, selanjutnya terjadi proses pengujian pada variabel angka tersebut, apakah nilai angka 4 diantara nilai angka 1 sampai 5 ? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi jumlah := jumlah + angka; menghasilkan proses 10 := 6 + 4, variabel jumlah sekarang bernilai 10.

Secara otomatis, Instruksi *for* angka := 1 to 5 *do* akan menambah nilai satu pada variabel angka, dengan demikian variabel angka sekarang adalah 5, selanjutnya terjadi proses pengujian pada variabel angka tersebut, apakah nilai angka 5 diantara nilai angka 1 sampai 5 ? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan intruksi jumlah := jumlah + angka; menghasilkan proses 15 := 10 + 5, variabel jumlah sekarang bernilai 15.

Secara otomatis, Instruksi *for* angka := 1 to 5 *do* akan menambah nilai satu pada variabel angka, dengan demikian variabel angka sekarang adalah 6, selanjutnya terjadi proses pengujian pada variabel angka tersebut, apakah nilai angka 6 diantara nilai angka 1

sampai 5 ? Bila hasil nilai *boolean false* maka proses pengulangan dihentikan. Karena logika *statement for-do* adalah aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> *boolean* masih tetap bernilai **true**. Jika <kondisi> bernilai *false* badan (*body*) pengulangan tidak akan dilaksanakan, atau pengulangan selesai.

**Flowchart:**



**2.3. PROGRAM APLIKASI : MEMPROSES KOLOM BARIS DENBGAN FOR-DO (FORDO3.AS)**

Untuk menggambarkan logika dari *statement For – Do Nested (For-Do Bersarang)*, berlaku juga untuk *statement While-Do Nested*, dan *Repeat-Until Nested*, penulis membuat contoh program aplikasi memproses kolom baris.

**Program Memproses\_kolom\_baris\_dengan\_for\_do;**

```

uses crt;
var
kolom, baris : integer;
begin
for kolom := 1 to 2 do
begin
writeln('Loop Kolom Ke- : ',kolom);
for baris := 1 to 3 do
begin
writeln('Loop Baris Ke- : ',baris);
writeln(['kolom,baris] = ',['kolom,',',baris,']');
end;
writeln;
end;
readln;
end.
    
```

**Tabel 3.1. Tabel Kemungkinan Output Analisa Algoritma Aplikasi Memproses Kolom Baris dengan FOR-DO (FORDO3.PAS)**

kolom	Proses Otomatis (tidak ada instruksi program)	For kolom := 1 to 2	Hasil Nilai Boolean	Write (kolom)	baris	Proses Otomatis (tidak ada instruksi program)	for baris := 1 to 3	Hasil Nilai Boolean	write (baris),(kolom,baris)
1	1	Apakah 1 diantara 1 s.d. 2?	T	1	1	1	Apakah 1 diantara 1 s.d. 3?	T	1 (1,1)
						2 := 1 + 1	Apakah 2 diantara 1 s.d. 3?	T	2 (1,2)
						3 := 2 + 1	Apakah 3 diantara 1 s.d. 3?	T	3 (1,3)
						4 := 3 + 1	Apakah 4 diantara 1 s.d. 3?	F	-
	2 := 1 + 1	Apakah 2 diantara 1 s.d. 2?	T	2	1	1	Apakah 1 diantara 1 s.d. 3?	T	1 (2,1)
						2 := 1 + 1	Apakah 2 diantara 1 s.d. 3?	T	2 (2,2)
						3 := 2 + 1	Apakah 3 diantara 1 s.d. 3?	T	3 (2,3)
						4 := 3 + 1	Apakah 4 diantara 1 s.d. 3?	F	-
	3 := 2 + 1	Apakah 3 diantara 1 s.d. 2?	F	-	-	-	-	-	-

**Analisa algoritma :**

Instruksi *for kolom := 1 to 2 do* berarti nilai awal dari variabel kolom adalah 1, selanjutnya terjadi proses pengujian pada variabel kolom tersebut, apakah nilai angka 1 diantara nilai angka 1 sampai 2 ? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) looping mengerjakan intruksi berikutnya *writeln(kolom)*, mencetak 1, dilanjutkan dengan intruksi *for baris := 1 to 3 do*

Instruksi *for baris := 1 to 3 do* berarti nilai awal dari variabel baris adalah 1, selanjutnya terjadi proses pengujian pada variabel baris tersebut, apakah nilai angka 1 diantara nilai angka 1 sampai 3 ? Bila hasil nilai *boolean true* maka masuk ke badan (*body*) looping mengerjakan intruksi berikutnya *writeln(baris); writeln(kolom,baris);* mencetak 1; mencetak 1,1.

Secara otomatis, Instruksi *for* angka := 1 to 3 do akan menambah nilai satu pada variabel baris, dengan demikian variabel baris sekarang adalah 2, selanjutnya terjadi proses pengujian pada variabel baris tersebut, apakah nilai angka 2 diantara nilai angka 1 sampai 3 ?, bila hasil nilai *boolean true*

maka masuk ke badan (*body*) *looping* mengerjakan instruksi berikutnya *writeLn*(baris); *writeLn*(kolom,baris); mencetak 2; mencetak 1,2.

Secara otomatis, Instruksi *for* angka := 1 to 3 *do* akan menambah nilai satu pada variabel baris, dengan demikian variabel baris sekarang adalah 3, selanjutnya terjadi proses pengujian pada variabel baris tersebut, apakah nilai angka 3 diantara nilai angka 1 sampai 3 ?, bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi berikutnya *writeLn*(baris); *writeLn*(kolom,baris); mencetak 3; mencetak 1,3.

Secara otomatis, Instruksi *for* angka := 1 to 3 *do* akan menambah nilai satu pada variabel baris, dengan demikian variabel baris sekarang adalah 4, selanjutnya terjadi proses pengujian pada variabel baris tersebut, apakah nilai angka 4 diantara nilai angka 1 sampai 3 ?, bila hasil nilai *boolean false* maka proses pengulangan dihentikan. Karena logika *statement for-do* adalah aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> *boolean* masih tetap bernilai **true**. Jika <kondisi> bernilai *false* badan (*body*) pengulangan tidak akan dilaksanakan, atau pengulangan selesai.

Setelah selesai melakukan *looping for* baris := 1 to 3 *do*, kembali melanjutkan *looping for* kolom := 1 to 2 *do*.

Instruksi *for* kolom := 1 to 2 *do* akan menambah nilai satu pada variabel kolom, dengan demikian variabel kolom sekarang adalah 2, selanjutnya terjadi proses pengujian pada variabel kolom tersebut, apakah nilai angka 2 diantara nilai angka 1 sampai 2 ?, Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi berikutnya *writeLn*(kolom), mencetak 2, dilanjutkan dengan instruksi *for* baris := 1 to 3 *do*.

Instruksi *for* baris := 1 to 3 *do* berarti nilai awal dari variabel baris adalah 1, selanjutnya terjadi proses pengujian

pada variabel baris tersebut, apakah nilai angka 1 diantara nilai angka 1 sampai 3 ?, Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi berikutnya *writeLn*(baris); *writeLn*(kolom,baris); mencetak 1; mencetak 2,1.

Secara otomatis, Instruksi *for* angka := 1 to 3 *do* akan menambah nilai satu pada variabel baris, dengan demikian variabel baris sekarang adalah 2, selanjutnya terjadi proses pengujian pada variabel baris tersebut, apakah nilai angka 2 diantara nilai angka 1 sampai 3 ?, Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi berikutnya *writeLn*(baris); *writeLn*(kolom,baris); mencetak 2; mencetak 2,2.

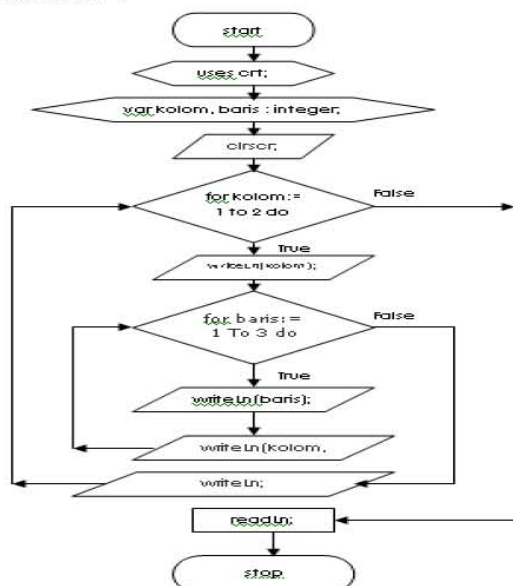
Secara otomatis, Instruksi *for* angka := 1 to 3 *do* akan menambah nilai satu pada variabel baris, dengan demikian variabel baris sekarang adalah 3, selanjutnya terjadi proses pengujian pada variabel baris tersebut, apakah nilai angka 3 diantara nilai angka 1 sampai 3 ?, Bila hasil nilai *boolean true* maka masuk ke badan (*body*) *looping* mengerjakan instruksi berikutnya *writeLn*(baris); *writeLn*(kolom,baris); mencetak 3; mencetak 2,3.

Secara otomatis, Instruksi *for* angka := 1 to 3 *do* akan menambah nilai satu pada variabel baris, dengan demikian variabel baris sekarang adalah 4, selanjutnya terjadi proses pengujian pada variabel baris tersebut, apakah nilai angka 4 diantara nilai angka 1 sampai 3 ?, Bila hasil nilai *boolean false* maka proses pengulangan dihentikan. Karena logika *statement for-do* adalah aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> *boolean* masih tetap bernilai **true**. Jika <kondisi> bernilai *false* badan (*body*) pengulangan tidak akan dilaksanakan, atau pengulangan selesai.

Setelah selesai melakukan *looping for* baris := 1 to 3 *do*, kembali melanjutkan *looping for* kolom := 1 to 2 *do*. Instruksi *for*

kolom := 1 to 2 do akan menambah nilai satu pada variabel kolom, dengan demikian variabel kolom sekarang adalah 3, selanjutnya terjadi proses pengujian pada variabel kolom tersebut, apakah nilai angka 3 diantara nilai angka 1 sampai 2 ?, Bila hasil nilai *boolean false* maka proses pengulangan dihentikan. Karena logika *statement for-do* adalah aksi (atau runtunan aksi) akan dilaksanakan berulang kali **selama** <kondisi> *boolean* masih tetap bernilai **true**. Jika <kondisi> bernilai *false* badan (body) pengulangan tidak akan dilaksanakan, atau pengulangan selesai.

**Flow chart :**



**III. KESIMPULAN**

Sebelum aplikasi dijalankan oleh komputer, pembuat program (*programmer*) yang membuat program aplikasi dengan menggunakan Tabel Kebenaran *Output* untuk menganalisa program aplikasi tersebut sudah dapat mengetahui hasil *output* dari program aplikasi tersebut. Baik program skala sederhana atau pun skala besar, dasar pemikirannya sama.

Begitu pula *user* dapat mengetahui *output* dari *listing* program pada aplikasi-aplikasi program, dengan menggunakan

tabel Kebenaran *output* tersebut. Sebagai contoh mahasiswa dapat dengan mudah mengetahui *output* dari *listing* program dari aplikasi program yang akan dianalisa *output* programnya dengan menggunakan Tabel Kebenaran *Output* ini, tanpa harus mencoba atau mengetik *listing* program tersebut dan mengeksekusinya.

**DAFTAR PUSTAKA**

Hasbi, M. 2002. Belajar Sendiri PEMROGRAMAN PASCAL TURBO PASCAL 7.0. Penerbit Graha Ilmu. Yogyakarta. Cetakan Pertama.

H.M., Jogiyanto. 1997. TURBO PASCAL. Penerbit Andi Offset. Yogyakarta. Cetakan Kedelapan.

Kadir, Abdul, Ir. 2001. Perograman Pascal. Penerbit Andi. Yogyakarta. Cetakan Ketiga.

K. O'brien, Stephen. 1976. Turbo Pascal. Published by Osborne McGraw-Hill

Munir, Rinaldi. 2001. Algoritma & Pemrograman Dalam Bahasa Pascal Dan C. Penerbit Informatika Bandung. Cetakan ke lima. Jilid 1.

Santoso, P. Insap, Ir, M.Sc. 2000. Struktur Data Menggunakan Turbo Pascal 6.0. Penerbit Andi Offset. Yogyakarta. Cetakan Keenam.

Schildt, Herbert. 1996. Advanced Turbo Pascal. Published by Osborne McGraw-Hill

Supardi, Yuniar, Ir. 2000. Pascal Dan Flowchart Lewat Praktek. Dinastindo. Jakarta.

Sutanta, Edhy. 2004. Algoritma Teknik Penyelesaian Permasalahan Untuk Komputansi. Graha Ilmu. Yogyakarta.