

SOAL DAN JAWABAN ALGORITMA.

Assalamualaikum Wr.Wb (http://2.bp.blogspot.com/-ydwRmax8Q8g/VEoNY7gsjfl/AAAAAAAAABM/o_3GppGvS3s/s1600/image010.gif)
kali ini saya akan share soal dan jawaban algoritma.

Berikut adalah soal dan jawaban algoritma:

Soal:

1. Apa definisi algoritma dan contohnya .!
2. Apa definisi Pseudo Code dan contohnya .!
3. Apa definisi flowchart dan contohnya .!
4. Jelaskan dengan contoh kasus algoritma yang memiliki struktur runtunan.!
5. Jelaskan dengan contoh kasus algoritma yang memiliki struktur percabangan.!
6. Jelaskan dengan contoh kasus algoritma yang memiliki struktur perulangan:
 - a. for
 - b. while

Jawaban:

1. Definisi Algoritma.

"Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis". Kata logis merupakan kata kunci dalam algoritma. Langkah-langkah dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar. Dalam beberapa konteks, algoritma adalah spesifikasi urutan langkah untuk melakukan pekerjaan tertentu. Pertimbangan dalam pemilihan algoritma adalah, pertama, algoritma haruslah benar. Artinya algoritma akan memberikan keluaran yang dikehendaki dari sejumlah masukan yang diberikan. Tidak peduli sebegus apapun algoritma, kalau memberikan keluaran yang salah, pastilah algoritma tersebut bukanlah algoritma yang baik.

Contoh:

Algoritma mendapatkan minyak dengan volume 4 liter.

- 1 Isi penuh ember 3 liter dengan minyak. *{ember 3 liter berisi minyak 3 liter}*
- 2 Tuangkan minyak dari ember 3 liter ke dalam ember 5 liter. *{ember 5 liter berisi minyak 3 liter}*.
- 3 Isi penuh ember 3 liter dengan minyak. *{ember 3 liter berisi minyak 3 liter}*
- 4 Tuang minyak dari ember 3 liter ke ember 5 liter hingga ember 5 liter penuh. *{di dalam ember 3 liter sekarang berisi minyak sebanyak 1 liter}*
- 5 Kembalikan minyak dari ember 5 liter ke dalam drumnya. *{ember 5 liter kosong}*
- 6 Tuangkan minyak dari ember 3 liter ke ember 5 liter. *{ember 3 liter kosong, ember 5 liter berisi minyak 1 liter}*
- 7 Isi penuh ember 3 liter dengan minyak, lalu tuang ke dalam ember 5 liter. Maka akan diperoleh minyak sebanyak 4 liter *{1 + 3 = 4 liter minyak }*.

2. Definisi Pseudo code.

Pseudo Code adalah urutan baris algoritma seperti kode pemrograman dan tidak memiliki sintak yang baku. Pseudo Code lebih umum digunakan oleh programmer yang berpengalaman. Akan tetapi, flowchart lebih mudah dimengerti oleh programmer pemula, pseudo code sangat mudah diimplementasikan ke dalam kode program dibandingkan dengan flowchart. Kita bisa bebas menulis pseudo code selama itu mudah dimengerti bagi orang lain. Tetapi disarankan untuk menggunakan keyword yang umum digunakan seperti : if, then, else, while, do, repeat, for, dan lainnya. Dan ikuti gaya penulisan pemrograman seperti Pascal, C++, dll.

Contoh:

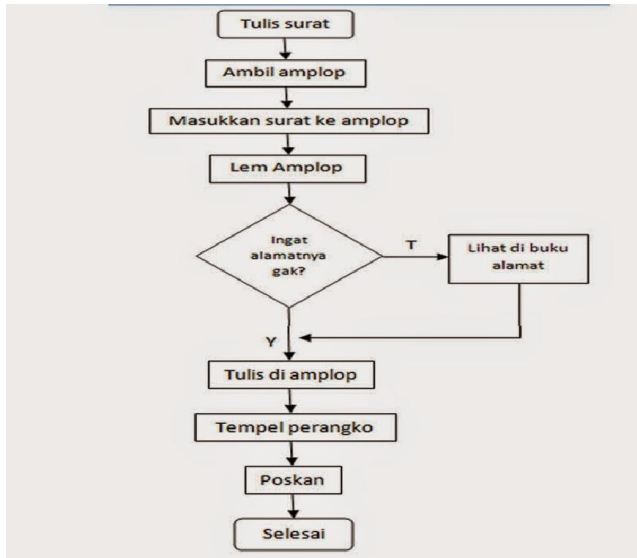
Pseudocode dari luas Persegi adalah:

```
-  
read(panjang, lebar)  
  Luas = panjang * lebar  
  
  write(Luas)  
end
```

3. Definisi Flowchart.

flowchart adalah suatu representasi secara diagram yang mengilustrasikan urutan dari operasi yang dilakukan untuk mendapatkan suatu hasil. Dengan kata lain, flowchart membantu kita untuk mengerti dan melihat bentuk algoritma dengan menampilkan algoritma dalam simbol-simbol gambar.

Contoh:



(<http://1.bp.blogspot.com/->

HI2dUEtSqOg/VEoWKVvkEVI/AAAAAAAAABc/-0fIBYFCzX4/s1600/Contoh%2Bflowchart%2Bsederhana%2Bbe
serta%2Bpenjelasannya%2Blengkap%2BTerbaru.jpg)

4. Contoh algoritma yang memiliki struktur runtutan:

Dibaca waktu tempuh seorang pelari marathon dalam jam-menit-detik (hh:mm:ss). Diminta mengkonversi waktu tempuh tersebut ke dalam detik. Tuliskan algoritmanya.

Ingatlah

1 menit = 60 detik

1 jam = 3600 detik

Misalnya waktu tempuh seorang pelari marathon adalah 1 jam, 5 menit, 40 detik. Dalam detik, waktu tempuh seluruhnya adalah $(1 \times 3600) + (5 \times 60) + 40 = 3940$ detik.

Penyelesaian :

Algoritma KONVERSI_JAM_KE_DETIK

{ dibaca jam-menit-detik (hh:mm:ss). Nilai jam-menit-detik dikonversi ke dalam detik, lalu ditampilkan ke piranti keluaran }

DEKLARASI

Type jam : record <hh : integer {0..23}, {jam}

```
mm : integer {0..59}, {menit}
ss : integer {0..59}, {detik}
>
```

J : jam

TotalDetik : integer

DESKRIPSI

```
read(J.hh,J.mm,J.ss))
```

```
TotalDetik ← (J.hh*3600) + (J.mm*60) + J.ss
```

```
write(TotalDetik)
```

Jika anda mentranslasikan algoritma KONVERSI_JAM_KE_DETİK ke dalam bahasa pascal, anda harus memperhatikan tipe bilangan bulat yang digunakan. Karena ranah nilai tipe integer terbatas, maka ada kemungkinan hasil pengubahan jam-menit-detik ke total detik bernilai negatif, sebab nilai $(J.hh*3600) + (J.mm*60) + J.ss$ berada di luar rentang tipe integer. Tipe longint yang mempunyai ranah yang lebih besar dapat dipakai untuk masalah ini.

Jadi, program KONVERSI_JAM_KE_DETİK dalam bahasa pascal adalah sebagai berikut :

```
program KONVERSI_JAM_KE_DETİK;
```

```
{ dibaca jam-menit-detik (hh:mm:ss). Nilai jam-menit-detik dikonversi ke dalam detik, lalu ditampilkan ke
piranti keluaran.}
```

```
uses wincrt;
```

```
(* DEKLARASI *)
```

```
type Jam = record
```

```
    hh : longint;    {jam}
```

```
    mm : longint;    {menit}
```

```
    ss : longint;    {detik}
```

```
end;
```

```
var
```

```
    J : Jam;
```

```
    TotalDetik : longint;
```

```
(* deskripsi *)
```

```
begin
```

```
    write('Jam :'); readln(J.hh);
```

```
    write('Menit:'); readln(J.mm);
```

```
    write('Detik:'); readln(J.ss);
```

```
    TotalDetik:= (J.hh*3600) + (J.mm*60) + J.ss;
```

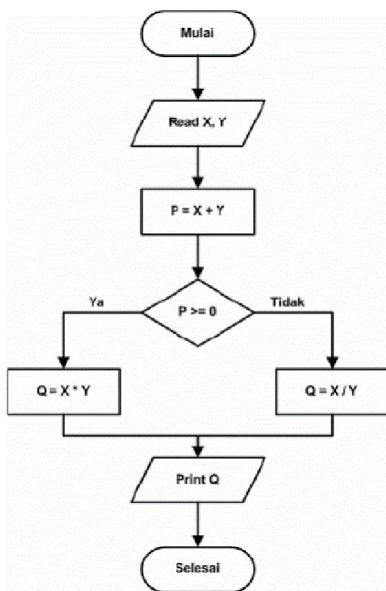
```
writeln('Total detik = ', TotalDetik);  
end.
```

5. Contoh kasus algoritma yang memiliki struktur percabangan:

Dalam suatu perhitungan, nilai $P = X + Y$. Jika P positif, maka $Q = X * Y$, sedangkan jika negative maka nilai $Q = X/Y$. Buatlah flowchart untuk mencari nilai P dan Q

Penyelesaian:

Pada contoh ini input yang dibutuhkan adalah nilai X dan Y , sedangkan proses pemeriksaan kondisi dilakukan pada nilai P apakah positif (termasuk 0) ataukah negative. Perhatikan flowchart penyelesaian masalah pada gambar 5.10.



(<http://2.bp.blogspot.com/->

Gambar 5.10. *Flowchart* penyelesaian untuk perhitungan dua buah bilangan.

ydwRmax8Q8g/VEoNY7gsjfl/AAAAAAAAABM/o_3GppGvS3s/s1600/image010.gif)

6. a. Contoh kasus algoritma yang memiliki struktur perulangan FOR:

Salah satu kelebihan komputer adalah kemampuannya untuk mengerjakan pekerjaan yang sama berulang kali tanpa mengenal lelah.

Sewaktu duduk di sekolah dasar, bila anda pernah dihukum oleh guru untuk menuliskan sebuah kalimat sebanyak 500 kali, misalnya dikarenakan anda nakal tidak mengerjakan PR. Misalkan kalimat yang harus ditulis 500 kali tersebut adalah:

"Saya berjanji tidak akan nakal dan malas lagi"

Bila pekerjaan menulis kalimat ini diserahkan kepada komputer, maka pemrogram mungkin menuliskannya sebagai berikut:

Algoritma menulis 500 kalimat

Deskripsi :

```
1. Saya berjanji tidak akan nakal dan malas lagi
2. Saya berjanji tidak akan nakal dan malas lagi
3. Saya berjanji tidak akan nakal dan malas lagi
4. Saya berjanji tidak akan nakal dan malas lagi
....
500
```

Tentu saja algoritma untuk menuliss 500 buah kalimat seperti diatas tidak elegan, karena instruksi

"Saya berjanji tidak akan nakal dan malas lagi" harus dibuat di dalam teks algoritma sebanyak 500 kali. Untuk mengatasi hal ini, maka anda dapat menggunakan struktur pengulangan for-do sehingga algoritma menulis 500 buah kalimat dapat ditulis sebagai berikut:

Algoritma menulis 500 kalimat

Menuliskan kalimat " Saya berjanji tidak akan nakal dan malas lagi " Sebanyak 500 kali.

Deskripsi :

```
for J dari 1 sampai 500 do
    t tulis kalimat " Saya berjanji tidak akan nakal dan malas lagi "
```

J adalah pencacah pengulangan yang dari 1 sampai 500. Komputer akan melaksanakan aksi tulis kalimat tersebut sebanyak 500 kali. Struktur pengulangan ini dapat ditulis secara umum sebagai berikut:

```
for pencacah pengulangan dari 1 sampai N do
    aksi
```

b. Contoh kasus algoritma yang memiliki struktur perulangan WHILE:

Bentuk umum struktur WHILE-DO adalah

```
while <kondisi> do  
  Aksi  
endwhile
```

aksi (atau runtunan aksi) akan dilaksanakan berulang kali sepanjang <kondisi> boolean masih tetap bernilai *true*, jika <kondisi> bernilai *false*, badan pengulangan tidak akan dilaksanakan. Pengulangan selesai.

Contoh 1:

Tuliskan algoritma untuk mencetak banyak HALO sebanyak 10 kali .

Algoritma cetak_banyak_halo

Deklarasi

K : integer {pencacah pengulangan}

Deskripsi

$K \leftarrow 1$ {inisialisasi}

While $k \leq 10$ do

Write ('HALO')

$K \leftarrow K+1$

Endwhile

{kondisi berhenti : $k > 10$ }.

Sekian post tentang soal dan jawaban algoritma dari saya .

Maaf bila ada tulisan yang tidak baik.

Wassalamualaikum Wr. Wb

Never miss our latest news, subscribe here for free