NAMA : MUH AKMAL

NIM : 180250502068

KLS : TI.B

1.Pengertian Algoritma

Algoritma adalah suatu urutan dari beberapa langkah logis dan sistematis yang digunakan untuk menyelesaikan masalah tertentu..Pendapat lain mengatakan definisi algoritma adalah proses atau serangkaian aturan yang harus diikuti dalam perhitungan atau operasi pemecahan masalah lainnya, terutama oleh komputer. Dengan kata lain, semua susunan logis yang diurutkan berdasarkan sistematika tertentu dan digunakan untuk memecahkan suatu masalah dapat disebut dengan algoritma.

2. Pentingnya Algoritma

Pentingnya algoritma adalah agar pengerjaan suatu program dapat dilakukan dengan runtut dan rapi. Agar ketika kita mengerjakan sebuah program dapat tersusun dan tidak bingung ketika mengerjakan sebuah program.

3. Konsep dasar Algoritma

Perbedaan Algoritma dan Program
Komputer hanyalah salah satu pemroses. Agar dapat dilaksanakan oleh komputer, algoritma harus ditulis dalam notasi bahasa pemrograman sehingga dinamakan program. Jadi program adalah perwujudan atau implementasi teknis Algoritma yang ditulis dalam bahasa pemrogaman tertentu sehingga dapat dilaksanakan oleh komputer.

Deskripsi Algoritma :
Aksi 1 : Tuangkan larutan dari bejana A ke dalam bejana B
Aksi 2 : Tuangkan larutan dari bejana B ke dalam bejana A.
· Algoritma TUKAR ISI BEJANA di atas tidak menghasilkan pertukaran yang benar. Langkah di atas
tidak logis, hasil pertukaran yang terjadi adalah percampuran kedua larutan tersebut.
· Untuk mempertukarkan isi duah bejana, diperlukan sebuah bejana tambahan sebagai tempat penampungan sementara, misalnya bejana C. Maka algoritma untuk menghasilkan pertukaran yang benar adalah sebagai berikut :

Deskripsi Algoritma 2:
Aksi 1 : Tuangkan larutan dari bejana A ke dalam bejana C.

Aksi 2 : Tuangkan larutan dari bejana B ke dalam bejana A.
Aksi 3 : Tuangkan larutan dari bejana C ke dalam bejana B.

Ciri penting algoritma:
· Algoritma harus berhenti setelah mengerjakan sejumlah langkah terbatas.
· Setiap langkah harus didefinisikan dengan tepat dan tidak berarti-dua (Ambiguitas).
· Algoritma memiliki nol atau lebih masukan (input).
· Algoritma memiliki nol atau lebih keluaran (output).
· Algoritma harus efektif (setiap langkah harus sederhana sehingga dapat dikerjakan dalam waktu yang efisien).

Struktur Dasar Algoritma
Langkah-langkah penyelesaian masalah bisa berupa :

a. Runtunan (sequence))
Sebuah runtunan terdiri dari satu atau lebih instruksi. Tiap instruksi dikerjakan berurutan sesuai aturan penulisannya. Urutan instruksi menentukan keadaan akhir algoritma, jika urutannya diubah maka hasil akhirnya mungkin akan berubah. Urutan instruksi menunjukkan cara berfikir penyusun algoritma dalam menyelesaikan masalah

Runtunan Instruksi:

Instruksi 1
Instruksi 2
Instruksi 3

b. Pemilihan (selection)

Adakalanya sebuah instruksi dikerjakan jika sebuah kondisi tertentu terpenuhi
Struktur umum :
If kondisi
then
Aksi
atau If kondisi then
Aksi 1
Else
Aksi 2

c. Pengulangan (repetition)
Komputer tidak pernah bosen dan lelah jika diminta untuk mengerjakan instruksi secara berulang-ulang.
Contoh :
· Menulis kalimat ”Saya harus lebih giat belajar” sebanyak 1000 kali
Ulangi :
– Tulis kalimat ” Saya harus lebih giat belajar”
Sampai jumlah_kalimat = 1000
· Mengupas 100 buah kentang

Selama kentang terkupas < 100 maka
– Kupas 1 kentang

4. Struktur penulisan algoritma

A. Penulisan dengan bahasa natural

Ketika menyajikan algoritma dalam bahasa natural, memang tidak ada aturan standar namun sebaiknya memperhatikan beberapa hal sebagai berikut:

1. Setiap Urutan langkah-langkah sebaiknya menggunakan penomoran dari 1,2 dan seterusnya.

2. Urutan langkah-langkah harus dimulai dengan kata **mulai** atau **Start** dan diakhiri dengan kata **selesai / stop.**

Atau anda juga bisa menggunakan istilah dengan bahasa lainnya yang serupa, Start dituliskan sebagai ‘inisialisasi’ atau ‘Mulai’, sedangkan End sendiri dituliskan sebagai ‘selesai’ dan dituliskan pada akhir algoritma.

3. Langkah-langkah penyelesaian masalah bisa ditulis secara berurutan dari awal sampai akhir.

4. Bisa menggunakan bahasa apapun yang mudah dipahami singkat jelas padat.

Contoh Penulisan Algoritma dengan bahasa natural dalam kehidupan sehari hari

Berikut adalah beberapa contoh penulisan algoritma bahasa natural untuk kasus-kasus dalam kehidupan sehari-hari:

a. Kasus 1:

Menukar isi gelas berisi kopi dan gelas berisi teh.

Untuk kasus ini kita misalkan gelas berisi kopi adalah gelas A, sedangkan gelas isi teh adalah gelas B

Penulisan algoritma bahasa natural:

1. Mulai
2. Sediakan satu gelas kosong misal namanya gelas C.
3. Masukan isi gelas A (gelas berisi kopi)kedalam gelas C (Gelas kosong)
4. Masukan isi gelas B (gelas berisi teh) kedalam gelas C(gelas kosong yang sebelumnya berisi kopi)
5. Masukan isi gelas C (gelas kosong yang sudah diisi kopi) kedalam gelas B (gelas kosong yang sebelumnya berisi teh)
6 Selesai.

b.Kasus 2:

Algoritma menyalakan motor

1. Mulai
2. Masukan kunci motor
3. Putar kunci motor hingga kontak aktif
4. Tekan tombol starter untuk menyalakan motor.
5. Jika motor tidak menyala gunakan cara manual.
5. Motor menyala
6. Selesai

c.Kasus 3:

Algoritma untuk kasus menanak nasi

1. Mulai
2. Cuci beras sampai bersih
3. Masukan beras kedalam mejic com
4. Colokan mejicom ke listrik
4. Tekan tombol menanak nasi dan tunggu hingga tombol mati
5. Nasi masak
6. Selesai.

B. Penulisan Algoritma Dengan Flowchart

Alogaritma pada awalnya dimodelkan dalam bentuk bangunan ruang oleh para ilmuan ketika computer mulai berkembang.Model penulisan alogaritma ini disebut sebagai bagan alir atau flowchart.Pedoman yang harus diikuti oleh perancang alogaritma ketika menggunakan metode ini adalah sebagai berikut:
1.Peletakkan symbol bagan alir(flowchart)sebaiknya dimulai dari atas kebawah dan dimulai dari sebelah kiri suatu halaman
2.Kegiatan didalam symbol bagan alir(flowchart)harus ditunjukkan dengan jelas nama kegiatan yang jelas
3.Harus dimulai dari symbol start(awal)dan diakhiri dengan symbol end(akhir)
4.Setiap kegiatan harus memiliki input dan menghasilkan output
5.Penjelasan dalam symbol flowchart sebaiknya menggunakan kata kerj,misalnya:
1. “Entry data siswa”
2. “Hitung A+B”

6.Setiap kegiatan dibagan alir(flowchart)harus memiliki alur data proses secra rinci dann jelas
7.Kegiatan yang terpotong dan akan disambung ditempat lain harus ditunjukkan dengan jelas menggunakan symbol penghubung
Lima jenis bagan alir(flowchart),adalah sebagai berikut:
1.Bagan alir system(system flowchart)yang menjelaskan urutan setiap proedur yag dapat pada system
2.Bagan alir dokuen(dokumen flowchart)yang menunjukkan arah aliran data laporan dan folmulir oada subprogram atau proses.
3.Bagan alir skematik(schematic flowchart)memiliki kemiripan dengan bagan alir system,yaitu untuk menggambarkan skema aliran data pada prosedur didalam system

4.Bagan alir program(program flowchart)berguna untuk melakukan analisis system dengan menggambarkan proses dalam suatu prosedur program
5.Bagan alir proses(process flowchart)merupakan bagan yang sering digunakan dalam aliran proses pada teknik industry.

C. Penulisan Dengan Peseudocode

Pseudocode adalah deskripsi dari algoritma pemrograman komputer yang menggunakan konvensi struktural dari suatu bahasa pemrograman, dan ditujukan agar dapat dibaca oleh manusia dan bukan oleh mesin. Pseudocode biasanya tidak menggunakan elemen cukup detail yang tidak perlu untuk kebutuhan pemahaman manusia dari suatu algoritma, seperti deklarasi variabel dan kode.

A.Tujuan pseudocode

adalah agar manusia dapat dengan mudah dalam pemahaman dibandingkan dengan menggunakan bahasa pemrograman yang umumnya digunakan, aspeknya yang relatif ringkas dan tidak bergantung pada suatu sistem tertentu yang merupakan prinsip utama dalam suatu algoritma.

B.Struktur Pseudocode

1.Judul

{Berisi Judul Algoritma}

2.Deskripsi

{Berisi Deklarasi Variabel atau Konstantan}

3.Implementasi

{Berisi Inti Algoritma}

Contoh Penulisan :

Menghitung penjumlahan

1.Mulai

2.Input a, b

c = a + b

Print "c"

3.Selesai

5. Mengenal Operator

Dalam bahasa pemrograman, kita sudah pernah mendengar apa yang di maksud dengan Operator. Operator dalam bahasa pemrograman itu sendiri, terbagi dalam beberapa jenis, diantaranya :

- Operator Aritmatika
- Operator Penugasan/assigment
- Operator Logika
- Operator String

1. Operator Aritmatika

   Operator aritmatika adalah operator yang digunakan hanya untuk melakukan operasi matematika, seperti : penjumblahan, pengurangan, perkalian, pembagian dan modulus (sisa hasil pembagian. Adapun simbol-simbol yang digunakan dalam operator ini antara lain :

- + : untuk penjumlahan
- - : untuk pengurangan
- * : untuk perkalian
- / : untuk pembagian
- % : modulus (sisa hasil bagi)

2. Operator Penugasan/assigment

   Operator Penugasan/assigment adalah operator yang berfungsi untuk mengisi nilai suatu variabel, yang ditandai berupa (=) sama dengan. Adapun contohnya sebagai berikut :
   B = 10
   Pada variabel B akan terisi dengan nilai 10
   A = 1 + B
   Pada variabel A akan terisi nilai 1 yang ditambahkan dengan nilai B yaitu 10

3. Operator Logika

   Operator Logika adalah operator yang hanya memiliki dua kondisi atau keadaan, yaitu true (1) dan false (0) yang biasanya digunakan untuk membandingkan hasil suatu keadaan.

4. Operator String

   Operator string adalah operator yang berfungsi untuk menggabungkan dua buah character dalam pemrograman. Adapun contohnya adalah sebagai berikut :
   A = “Belajar”
   B = “Bahasa C++”
   C = A + B
   Maka nilai C adalah “Belajar Bahasa C++”

6. Percabangan

Pengertian algoritma pemrograman seleksi kondisi, atau disebut juga algoritma percabangan (atau disebut juga dengan flow control dan algoritma pemilihan) adalah salah satu jenis perintah dalam algoritma yang digunakan sebagai cara untuk memberitahukan program tentang perintah apa yang harus dijalankan, dimana perintah tersebut disesuaikan dengan beberapa kondisi tertentu. Fungsi algoritma percabangan ini pada adalah untuk memproses keputusan yang tepat dan sesuai dengan yang keinginan pengguna sistem berdasarkan beberapa kondisi yang terjadi pada sistem yang digunakan tersebut.

Dalam sebuah program atau sistem, ada saatnya sebuah instruksi atau perintah hanya bisa dilakukan jika memenuhi suatu kondisi atau persyaratan tertentu. Itu mengapa, algoritma

percabangan ini bisa disebut juga dengan algoritma seleksi kondisi. Agar Anda paham maksudnya, kami berikan sebuah contoh. Misalkan, kita hendak menentukan apakah suatu bilangan termasuk bilangan genap atau bilangan ganjil. Nah, algoritmanya dapat kita jelaskan sebagai berikut:

1. Mulai
2. Masukkan suatu bilangan, misalkan bilangan X)
3. Jika bilangan X habis dibagi dua, maka lanjut ke perintah keempat. Jika tidak lanjut ke perintah kelima.
4. Tuliskan "X adalah bilangan genap". Lanjut ke perintah keenam.
5. Tuliskan "X adalah bilangan ganjil"
6. Selesai

Dari algoritma di atas, kita bisa lihat bahwa ada dua kemungkinan perintah yang akan dikerjakan setelah perintah ketiga dikerjakan. Perintah pertama, jika bilangan X habis dibagi dua maka selanjutnya perintah keempat yang dikerjakan, kemudian lompat ke perintah keenam dan perintah kelima tidak dikerjakan. Perintah kedua, jika bilangan X tidak habis dibagi dua maka melompat ke perintah kelima dan perintah keempat tidak dikerjakan. Kedua perintah tersebut sama-sama berakhir pada perintah keenam, yang menyatakan bahwa proses algoritma telah selesai.

Bagaimana, sudah paham mengenai maksud dari algoritma percabangan ini? Nah, ternyata algoritma percabangan ini banyak macamnya. Namun, inti dari algoritma ini sama, yaitu suatu program atau sistem akan mengerjakan sebuah perintah yang disesuaikan dengan kondisi atau syarat tertentu. Apa saja macam-macam algoritma percabangan? Artikel kali ini akan mengulasnya untuk Anda. Berikut ini pembahasannya:

1. Percabangan untuk 1 kondisi

Pada percabangan jenis ini, hanya ada satu kondisi yang menjadi syarat untuk melakukan satu buah atau satu blok instruksi. Format umum dari algoritma percabangan dengan satu kondisi adalah sebagai berikut:

```
IF kondisi THEN
instruksi
ENDIF
```

Arti dari format di atas, jika "kondisi" bernilai benar atau tercapai, maka aksi dikerjakan. Sedangkan jika bernilai salah, maka instruksi tidak dikerjakan dan proses langsung keluar dari percabangan dan kembali lagi ke kondisi awal.

Contoh dari penggunaan algoritma percabangan untuk satu kondisi adalah sebagai berikut:

```
if A > B then
write (A)
end if
```

Instruksi di atas artinya instruksi akan menampilkan nilai A hanya jika kondisi "A lebih besar daripada B" bernilai benar. Jika bernilai salah, maka tidak ada aksi yang akan dilakukan atau proses langsung keluar dari percabangan (end if).

Berikut ini kami berikan contoh beberapa contoh program algoritma percabangan untuk satu kondisi menggunakan macam-macam bahasa pemrograman. Berikut ini adalah contoh untuk program menggunakan bahasa Pascal adalah sebagai berikut:

```
uses crt;
var
jeniskelamin:char;
begin
clrscr;
writeln('Jenis Kelamin : ');
writeln('L unutk laki-laki, P untuk perempuan');
writeln('Jenis kelamin anda: ');readln(jeniskelamin);
if(jeniskelamin = 'l') then writeln('Laki-laki');
if(jeniskelamin = 'p') then writeln('Perempuan');
readkey;
end
```

Contoh lainnya dari program percabangan untuk satu kondisi pada suatu program menggunakan bahasa C++ adalah sebagai berikut:

```
#include <iostream.h>
int main (){
int nilai;
char a;
cout<<"Masukkan Nilai Anda:";
cin>>nilai;
if (nilai>60){
cout<<"Selamat Anda Lulus!!";}
cin>>a;
return 0;}
```

2. Percabangan untuk 2 kondisi

Pada percabangan jenis ini, ada dua kondisi yang menjadi syarat untuk dikerjakannya salah satu dari dua instruksi. Kondisi ini bisa bernilai benar atau salah. Bentuk umum dari percabangan dengan dua kondisi adalah sebagai berikut:

```
IF kondisi THEN
instruksi 1
ELSE
instruksi 2
ENDIF
```

Arti dari format di atas, jika "kondisi" bernilai benar maka instruksi 1 yang akan dikerjakan. Sedangkan jika bernilai salah), maka instruksi 2 yang akan dikerjakan. Perbedaannya dengan percabangan untuk satu kondisi terletak pada adanya dua instruksi untuk dua kondisi, yaitu kondisi bernilai benar dan kondisi bernilai salah.

3. Percabangan untuk 3 kondisi atau lebih

Algoritma percabangan untuk tiga kondisi atau lebih adalah bentuk pengembangan dari dua macam algoritma percabangan yang telah dibahas sebelumnya. Karena itu, percabangan jenis

ini akan memiliki banyak variasi. Secara umum, format percabangannya dapat dituliskan sebagai berikut :

```
IF kondisi THEN
instruksi 1
ELSE IF kondisi 2 THEN
instruksi 2
ELSE
instruksi 3
ENDIF
```

Maksud dari algoritma di atas, instruksi 1 akan dikerjakan jika “kondisi 1” bernilai benar. Jika bernilai salah, pemeriksan dilanjutkan ke “kondisi 2”. Jika “kondisi 2” bernilai benar, maka instruksi 2 dikerjakan. Jika tidak, pemeriksaan dilanjutkan pada kondisi-kondisi lainnya. Pemeriksaan ini akan terus dilakukan terhadap semua kondisi yang ada. Jika tidak ada satu pun kondisi yang bernilai benar maka pernyataan yang dikerjakan adalah instruksi 3 atau instruksi (n+1) pada percabangan lebih dari 3 kondisi.

4. Percabangan “Case of….”

Selain menggunakan format yang dijelaskan pada poin 3, percabangan 3 kondisi atau lebih bisa juga menggunakan format “Case Of”. Format ini memiliki kegunaan yang sama, tetapi format ini digunakan untuk memeriksa data yang bertipe karakter atau integer. Secara umum format penulisannya adalah sebagai berikut:

```
switch (ekspresi) {
case kontanta-1:
instruksi 1 break;
case konstanta-2:
instruksi 2 break;
default:
instruksi 3}
```

Contoh penerapan percabangan Case Of dalam sebuah program menggunakan bahasa Pascal adalah sebagai berikut:

```
uses wincrt;
var x : integer;
begin
write (‘Masukkan sebuah nilai [0…3] : ‘);
readln (x);
Case (x) of
0 : Writeln(‘X bernilai 0’);
1 : Writeln(‘x bernilai 1’);
2 : Writeln(‘X bernilai 2’);
3 : Writeln(‘X bernilai 3’);
else
Writeln(‘X tidak bernilai 0, 1, 2, ataupun 3’);
end;
end.
```

Contoh program percabangan Case Of menggunakan bahasa C++ :

```
void main() {
int nHari;
cout << “Masukkan No Hari [1..7] : “;
cin >> nHari;
cout << “Ini adalah hari “;
switch (nHari) {
case 1:
cout << “Ahad”;
break;
case 2:
cout << “Senin”;
break;
case 3:
cout << “Selasa”;
break;
case 4:
cout << “Rabu”;
break;
case 5:
cout << “Kamis”;
break;
default:
cout << “Jumat”;}
getch();}
```

5. Percabangan bersarang

Percabangan bersarang adalah instruksi yang terdiri dari adanya percabangan yang lain di dalam percabangan, atau di dalam percabangan ada percabangan lagi. Format penulisan untuk percabangan bersarang adalah sebagai berikut:

```
If <kondisi1> then
if <kondisi2> then
Instruksi1
Else
Instruksi2
Else
If <kondisi3>
Instruksi3
Else
Instruksi4
EndIf
```

Jika kondisi berjumlah lebih dari 3 kondisi, polanya tetap sama. Untuk kondisi ke 2 dan seterusnya, penulisannya menggunakan “ELSE IF kondisi THEN”, sedangkan untuk kondisi terakhir cukup menggunakan ELSE saja.

Mulanya, “kondisi1” dicek nilai kebenarannya. Jika benar, maka dicek nilai kebenaran “kondisi2”. Jika “kondisi2” benar, maka dikerjakan Instruksi1. Jika tidak, dikerjakan Instruksi2.

Sedangkan jika “kondisi1” tidak benar, maka akan dicek nilai kebenarannya. Jika “kondisi3” bernilai benar, maka dikerjakan Instruksi3. Jika tidak, maka akan dikerjakan Instruksi4.

Inilah salah satu contoh program percabangan bersarang (Nested If) menggunakan bahasa Pascal:

```
uses wincrt;
var x, y, z : real;
begin
write (‘Masukkan bilangan pertama : ‘);
readln (x);
write (‘Masukkan bilangan kedua : ‘);
readln (y);
write (‘Masukkan bilangan ketiga : ‘);
readln (z);
if x > y then
if x > z then
write (‘Bilangan terbesar : ‘,x:5:2)
else
write (‘Bilangan terbesar : ‘,z:5:2)
else
if y > z then
write (‘Bilangan terbesar :’,y:5:2)
else
write (‘Bilangan terbesar : ‘,z:5:2);
end.
```

Dan di bawah ini adalah satu contoh program percabangan bersarang lainnya menggunakan bahasa C++ :

```
#include <iostream.h>
void main() {
int A, B, C;
cout << “masukan angka 1 =  “;
cin >> A;
cout << “masukan angka 2 =  “;
cin >> B;
cout << “masukan angka 3 =  “;
cin >> C;
if(A<B){
if(A<C)
cout<< “angka terkecil adalah : ” << A;
else
cout<< “angka terkecil adalah : ” << C;
}
else if(B<C)
cout<< “angka terkecil adalah : ” << B;
else
cout<< “angka terkecil adalah : ” << C;
}
```

Setelah Anda mempelajari algoritma percabangan dalam artikel ini, Anda selanjutnya dapat mengembangkannya ke dalam bentuk flowchart. Apa fungsi flowchart dalam pemrograman? Simak artikelnya di website kami ini. Sekian artikel kami kali ini mengenai algoritma percabangan.
Eitss, Anda juga harus mengerti dahulu pengertian algoritma, flowchart, dan pseudocode dalam mempelajari pemrograman. Anda juga bisa mengetahui semuanya di website kami ini. Semoga artikel kami ini bermanfaat bagi Anda untuk mengenal algoritma percabangan.

7. Perulangan

Pengulangan atau disebut sebagai looping adalah instruksi khusus dalam bahasa pemrograman dan algoritma yang digunakan untuk mengulang beberapa perintah sesuai dengan jumlah yang telah ditentukan. tujuannya adalah untuk mempermudah pengerjaan program dan untuk mempersingkat instruksi program. dengan pengulangan instruksi program yang seharunya ditulis dengan jumlah baris yang banyak bisa dipersingkat.

1.Intruksi Perulangan dalam algoritma
Ada 3 jenis bentuk instuksi format pengulangan di dalam algoritma yitu sebagai berikut:
Pengulangan for disebut juga sebagai pengulangan di awal format instruksinya adalah sebagai berikut:

For i ← nilai_awal to nilai_akhir do

Statement

Endfor

Contoh:

Buatlah algoritma untuk mencetak tulisan "Algoritma Menyenangkan" sebanyak 100 baris maka instruknya adalah:

Jawab:

program looping_for

DEKLARASI

i:integer

ALGORITMA:

for i ← 1 to 100 do

writeln('Algoritma Menyenangkan')

endfor

2. Penggulangan menggunakan Instruksi While DO

Format:

while kondisi do

pernyataan

endwhile

Contoh Kasus:

Buatlah algoritma untuk mencetak tuliasan angka 1 sampai 100

Jawaban:

promgram looping

DEKLARASI

var i:integer

ALGORITMA:

i ← 0

while i <100 do

writeln ('angka ke', i)

i ← i+1 {pencacah naik}

endwhile

3. Pengulangan dengan Menggungakan Repeat Until

Format:

repeat

statement

pencacah naik atau pencacah turun until kondisi

contoh kasus:

Buatlah algoritma untuk mencetak tulisan Hello World sebanyak 1000 baris.

Jawab:

program cetak

DEKLARASI

i:integer

ALGORITMA:

i ← 1 {isi nilai awal variable i dengan angka 1)

repeat write ('Hello World') i ← i+1

until i<=1000

A Kapan Harus menggunakan Instruksi pengolangan di dalam algoritma?

Sebenarnya untuk memecahkan masalah kasus pemrograman bisa dipecahkan dengan banyak cara tertangung logika si programmer, seperti halnya banyak jalan menuju kota jakarta, tapi tujuannya tetap saja, tapi yang terbaik adalah bagaimana membuat program dengan instruksi sedikit dan proses sangat cepat.

Programmer yang pintar akan sangat mudah sekali mencari cara yang terbaik untuk membuat program dengan instruksi yang singkat namun prosesnya cepat. salah satu instruksi yang bisa digunakan adalah pengulangan, ketika sebuah kasus memungkinkan untuk menggunakan pengulangan maka harus menggunakan pengulangan.

B.Kapan instruksi pengulangan harus digunakan?

Instruksi pengulangan digunakan manakala program atau bagian program terindikasi bisa menggunakan proses pengulangan.

Sebagai contoh sederhana. misalkan untuk kasus program untuk menampilkan angka 1 sampai 1000, atau program untuk mencetak tulisan tertentu dalam jumlah tertentu.

Sebenarnya bisa saja tidak menggunakan pengulangan, namun kurang efektif walaupun hasil outputnya bisa saja sama.

2.Pengulangan dengan Pencacah Naik

Pengulangan pencacah naik yaitu kondisi pengulangan yang dimulai dengan kondisi pencacah kecil ke besar naik sampai jumlah pengulangan yang diinginkan.

Contoh: buat algoritma untuk mencetak tulisan "Teknologi Modern" sebanyak 1000 baris.

Jika menggunakan pencacah naik instruksi algoritmanya adalah sebagai berikut:
Jawab:

algoritma pencacah_naik

DEKLARASI

i:integer

ALGORITMA:

for i ← 1 to 1000 do

writeln ('Teknologi Modern');

endfor

Pengulangan yang digunakan di algorirma di atas disebut pengulangan pencacah naik karena dimulai dari angka 1 terus naik sampai angka 1.000. bisa juga menggunakan Repeat Until atau While DO.

3.Pengulangan dengan Pencacah Turun

Pengulangan pencacah turun yaitu kodisi pengulangan yang dimulai dengan kondisi nilai pencacah dari besar ke kecil. sesuai dengan jumlah yang diinginkan.

Contoh: buat algoritma untuk mencetak tulisan "Teknologi HP Modern" sebanyak 1000 baris.

Jika menggunakan pencacah turun instruksi algoritmanya adalah sebagai berikut, misal menggunakan intruksi repeat until:

Jawab:

algoritma cacah_turun

DEKLARASI

i:integer

ALGORITMA:

i ← 1000 {nilai pencacah awal 1000 dimasukan ke variable i}

repeat writeln ('teknologi HP Modern')

8. Latihan

1. Algoritma Narasi

Contoh: Algoritma Kelulusan_mhs

Persoalan: Diberikan data berupa nama dan nilai mahasiswa. Jika nilai mahasiswa lebih besar atau sama dengan 60 maka mahasiswa tersebut dinyatakan lulus. Sedangkan jika nilainya lebih kecil dari 60, maka mahasiswa tersebut dinyatakan tidak lulus.

Algoritmanya akan seperti berikut:
baca nama dan nilai mahasiswa.
jika nilai >= 60 maka
keterangan = lulus
tetapi jika
keterangan = tidak lulus.
tulis nama dan keterangan

2. Algoritma Pseudo Code

Contoh; Algoritma Kelulusan_mhs

Persoalan: Diberikan data berupa nama dan nilai mahasiswa. Jika nilai mahasiswa lebih besar atau sama dengan 60 maka mahasiswa tersebut dinyatakan lulus. Sedangkan jika nilainya lebih kecil dari 60, maka mahasiswa tersebut dinyatakan tidak lulus.

Deklarasi dari tipe datanya akan seperti berikut:
Nama = string
Nilai = integer
Keterangan = string
Algoritmanya akan seperti berikut:
read (nama, nilai)
if nilai >= 60 then

keterangan = ‘lulus’
else
keterangan = ‘tidak lulus’
write(nama, keterangan)

3. Algoritma flowchart

Jumlah barang memiliki sifat yang dinamik sesuai dengan masukan dari pengguna. Intinya, jika total pembeliannya kurang dari 1500000, maka pembeli tidak akan mendapatkan diskon.

- Flowchart menghitung luas lingkaran

  Algoritma flowchart di bawah ini adalah contoh flowchart untuk menghitung luas lingkaran dengan rumus L = πr2:

  Algoritma:

- Program dimulai
- Tentukan nilai phi dan r
- Hitung L = phi x r2
- Cetak Hasil L
- Program Selesai

  Flowchart: