

**PEMILIHAN**

# **PEMILIHAN**

## **Menganalisis Kasus**

Analisis kasus adalah konstruktor terpenting dalam algoritma pemrograman sehingga membuat teks algoritma yang sama dapat menghasilkan eksekusi yang berbeda-beda. Setiap statement dalam pascal normalnya dijalankan secara sekuensial (beruntun), dari atas kebawah seperti yang tertulis pada *source code* programnya. Analisis kasus dapat merubah urutan eksekusi.

Jika kita mendefinisikan analisis kasus berarti juga mendefinisikan :

**Kondisi** : yang berupa suatu ekpresi Boolean yang menghasilkan *true* atau *false*.

**Aksi** : yang akan dilaksanakan jika kondisi yang dipasangkan dengan aksi yang bersangkutan dipenuhi.

Suatu aksi hanya akan dilaksanakan jika kondisi yang dipasangkan dengan aksi yang bersangkutan terpenuhi (bernilai true). Penentuan kondisi Boolean dan aksi yang dilakukan bergantung pada jumlah kasus yang terdapat pada masalah tersebut. Ungkapan kondisi dapat dihasilkan dengan operator perbandingan dan operator logika. ungkapan kondisi dapat dibentuk dengan mengkombinasikan operand yang bertipe sama dengan salah satu operator.

## **Satu Kasus dan Contoh Masalahnya**



Sebuah program tidak selamanya akan berjalan dengan mengikuti struktur berurutan, kadang kadang kita perlu merubah urutan pelaksanaan program dan menghendaki agar pelaksanaan program meloncat ke baris tertentu. Peristiwa ini kadang disebut sebagai percabangan/pemilihan atau keputusan. Algoritma percabangan pada pemrograman umumnya menggunakan kata kunci IF (jika), THEN (maka), dan ELSE (selainnya).

Algoritma percabangan dengan 1 kondisi memiliki format dasar seperti berikut :

**IF <kondisi>**

**THEN <pernyataan>**

Pada format diatas , jika <kondisi> bernilai benar maka <pernyataan> dikerjakan, sedangkan jika bernilai salah, maka <pernyataan> tidak dikerjakan dan proses langsung keluar dari percabangan.

**Contoh Kasus :** Nilai siswa akan gagal bila bernilai ≤ 60

Satu kondisi (if-then) : artinya hanya ada satu kondisi yang menjadi syarat untuk melakukan satu atau satu blok (sekelompok) pernyataan. Bentuk umum algoritma teks standar percabangan dengan satu kondisi :

if <kondisi> then

*pernyataan*

Jika <kondisi> bernilai benar maka pernyataan dikerjakan, sedangkan jika tidak, maka pernyataan tidak dikerjakan dan proses langsung keluar daripercabangan (begin).

if A>B then

write (A)

Ekspresi di atas menunjukkan bahwa perintah menulis/menampilkan A dikerjakan hanya jika kondisi A>B bernilai benar. Jika yang terjadi adalah sebaliknya, tidak ada pernyataan yang dilakukan atau proses langsung keluar dari percabangan (endif ).Secara flowchart ekspresi itudapat ditulis seperti berikut.

Perhatikan bahwa pada kotak belah ketupat memiliki dua cabangarus data, yang satu untuk kondisi bernilai benar (y, artinya ya), sedang yanglain untuk kondisi bernilai salah (t, artinya tidak).Jika kondisi bernilai benar(y) maka perintah yang dikerjakan adalah :

write(A)

Jika kondisi salah maka arus data langsung menuju ke bawah tanpa mengerjakan pernyataan apapun.

## **Dua Kasus dan Contoh Masalahnya**



Percabangan If ... Else artinya ada dua kondisi yang menjadi syarat untuk dikerjakannya dua jenis pernyataan. Bentuk umum percabangan dengan dua kondisi :

**IF <KONDISI> THEN**

**PERNYATAAN1**

**ELSE**

**PERNYATAAN2**

Jika <kondisi> bernilai benar maka pernyataan1 dikerjakan.Sedangkan jika tidak (<kondisi> bernilai salah), maka pernyataan yangdikerjakan adalah pernyataan2. Berbeda dengan percabangan satu kondisi,pada percabangan dua kondisi ada dua pernyataan untuk dua keadaan kondisi, yaitu untuk <kondisi> yang bernilai benar dan <kondisi> yang bernilai salah.Contoh algoritma percabangan dua kondisi :

if A>B then

write (A)

else

write (B)

Ekspresi di atas sedikit berbeda dengan sebelumnya. Perintah menulis/menampilkan A dikerjakan hanya jika kondisi A>B bernilai benar,sedangkan jika yang terjadi adalah sebaliknya maka pernyataan yangdilakukan adalah menulis B.Secara flowchart pernyataan di atas dapatditulis sebagai berikut.

Pada format di atas, Jika <kondisi> bernilai benar maka pernyataan 1 dikerjakan. Sedangkan jika tidak (<kondisi>bernilai salah), maka pernyataan yang dikerjakan adalah pernyataan 2. Berbeda dengan percabangan 1 kondisi, pada percabangan dua kondisi ada dua pernyataan untuk kedua kondisi,yaitu untuk <kondisi> yang bernilai benar dan <kondisi> yang bernilai salah.

## **Tiga Kasus atau Lebih dan Contoh Masalahnya**



Percabangan dengan tiga kondisi atau lebih adalah bentuk pengembangan dari dua bentuk percabangan percabangan yang telah kita bahas sebelumnya. Akan ada banyak sekali variasinya tetapi secara umum ekspresi percabangannya dapat kita tuliskan sebagai berikut :

**IF. <KONDISI1>**

**THEN**

**PERNYATAAN1**

**ELSE IF <KONDISI2> THEN**

**PERNYATAAN2**

**ELSE IF <KONDISI(N-1)> THEN**

**PERNYATAAN(N)**

**ELSE**

**PERNYATAAN(N)**

Jika di telusuri algoritma nya :

Mula-mula <kondisi1> dicek nilai kebenarannya. Jika benar, maka dikerjakan pernyataan1. Jika salah, maka dicek nilai kebenaran <kondisi2>. Jika <kondisi2> benar,maka dikerjakan pernyataan2.

Jika tidak algoritma akan mengecek ke kondisi berikutnyadengan cara yang sama dengan yang sebelumnya. Terakhir, jika semua kondisi bernilaisalah, maka pernyataan yang dikerjakan adalah Pernyataan (n+1).

Pada algoritma di atas pernyataan1 akan dikerjakan jika <kondisi1> bernilai benar, jika tidak pemeriksan dilanjutkan ke <kondisi2>. Jika <kondisi2> bernilai benar maka pernyataan2 dikerjakan. Jika tidak, pemeriksaan dilanjutkan pada kondisi-kondisi berikutnya. Pemeriksaan ini terus terhadap semua kondisi yang ada. Jika tidak ada kondisi yang benar maka pernyataan yang dikerjakan adalah pernyataan(n+1).

## **Strukture CASE**

Percabangan lebih dari 2 kondisi selain menggunakan format If ... Elseif ... Then ada juga menggunakan format Case Of. Pernyataan switch – case memiliki kegunaan yang sama seperti if – else bertingkat, tetapi untuk memeriksa data yang bertipe karakter atau integer. Secara umum bentuk penulisannya adalah sebagai berikut:

**switch** (ekspresi) {

**case** kontanta-1:

 // pernyataan 1 break;

**case** konstanta-2:

 // perintah 2 break;

**default:**

// perintah

 }

## **Implementasi**

1. **Implementasi kedalam bentuk source code, kasus yang dibahas**
* Satu kasus / if ... then

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.**

* Dua kasus / if ... else

uses wincrt;

**var** suhu: integer;

**begin**

 write ('masukan nilai suhunya : ');

 read (suhu);

 if suhu > 37 then

 writeln ('panas')

 else

 writeln ('tidak panas');

 write ('selesai')

**end.**

* Tiga kasus / if ... else if ... then

uses crt;

**var**

 nm : string [20];

 nilai : integer;

**begin**

 clrscr;

 write('masukkan nama anda :');readln(nm);

 write('masukkan nilai anda :');readln(nilai);

 if nilai > 90 then

 writeln('selamat nilai anda sangat baik = ')

 else if (nilai>=60) and (nilai <= 90) then

 writeln('selamat baik);

 else

 writeln('maaf anda gagal dalam ujian ini');

 readln;

**end.**

1. **Implementasi pengunaan Case dalam permogramana Turbo Pascal**

**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.**



****