

Struktur Data & Algoritma

PERTEMUAN 3

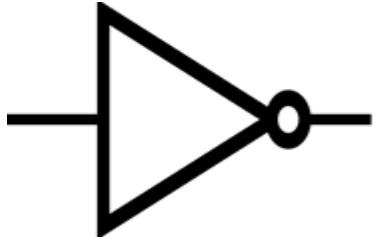
CATUR NUGROHO, M.KOM

PEMILIHAN/PERCABANGAN



Konsep Percabangan

- Percabangan adalah adalah suatu kondisi dimana sebuah instruksi tidak dijalankan secara sekuensial, tetapi lebih berdasarkan pada kondisi.
- Pernyataan percabangan memungkinkan suatu pernyataan dieksekusi hanya **jika suatu kondisi terpenuhi atau tidak terpenuhi.**
- **Program hanya akan menjalankan statement yang ada di dalam percabangan jika kondisi bernilai benar atau true.**



Kondisi pada percabangan harus berupa operator Boolean. **Contoh:**

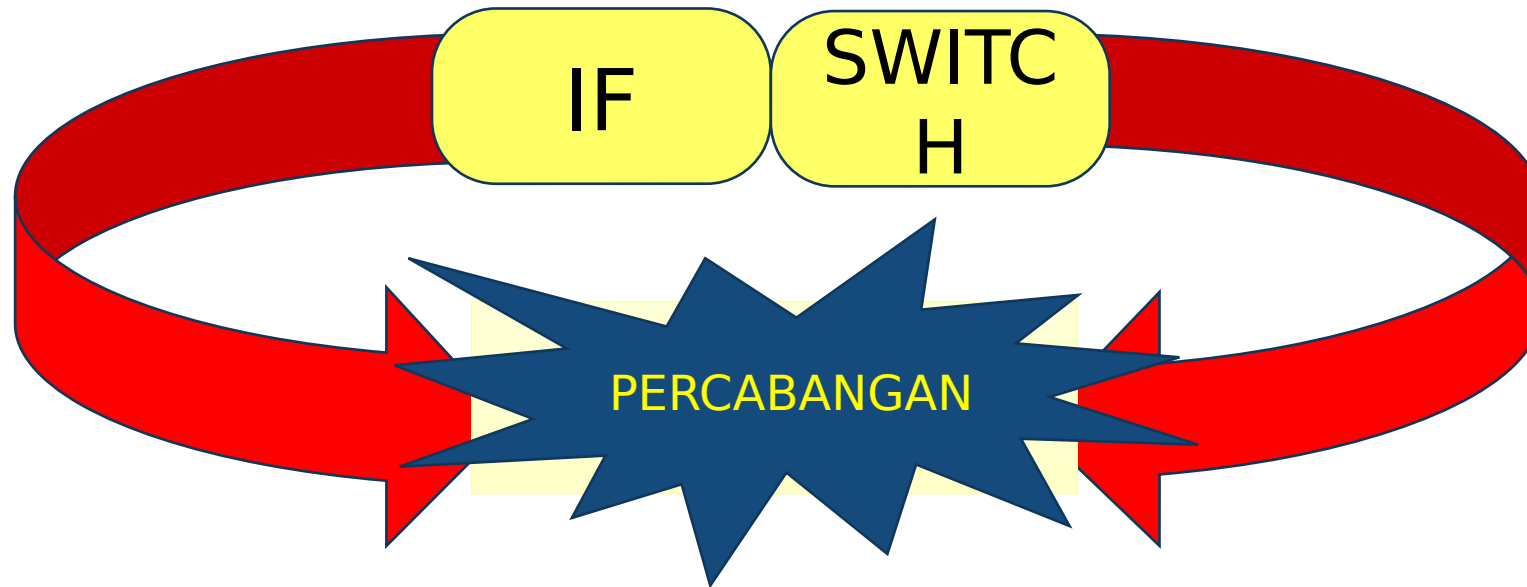
`int a = 5 // statement bukan merupakan kondisi`

`int a == 5 // statement yang berupa kondisi`



Kondisi menggunakan **operator – operator Boolean** yang hanya memberikan hasil true atau false.

seperti : `==, >, <, >=, <=, !=, &, &&, |, ||`



IF.. ELSE (1)

Digunakan untuk menguji sebuah kondisi.
Bila kondisi yang diuji terpenuhi, program akan menjalankan pernyataan - pernyataan tertentu Bila kondisi yg diuji salah, program akan menjalankan pernyataan lain

Bentuk umumnya adalah :

IF kondisi **then**

{pernyataan yg dijalankan jika kondisi terpenuhi}

else

{pernyataan yg dijalankan jika kondisi tidak terpenuhi}

JENIS PERCABANGAN IF

Proses percabangan IF memiliki tiga kasus, antara lain :

Percabangan Tunggal

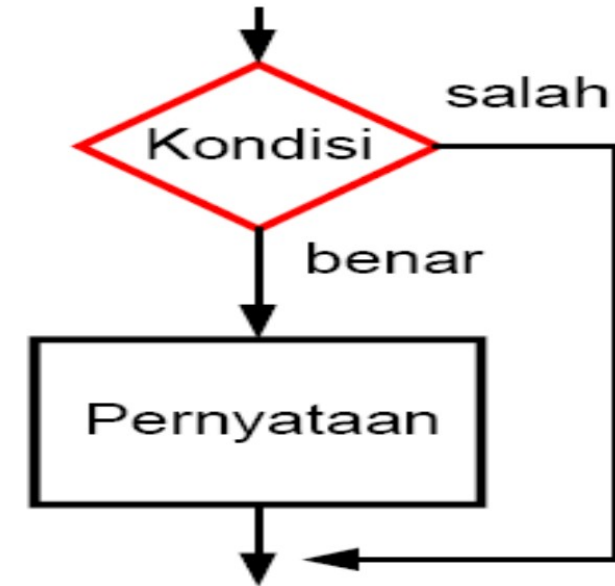
- Percabangan tunggal bekerja apabila hanya ada SATU alternatif instruksi yang dijalankan.
- Logika ini memungkinkan kompiler menjalankan sebuah instruksi atau tidak sama sekali.
- Kompiler baru akan menjalankan instruksi di dalam notasi percabangan apabila kondisi yang diinginkan terpenuhi atau memberikan nilai akhir true, Apabila kondisi false maka tidak akan ada instruksi yang dijalankan.

JENIS PERCABANGAN IF (Lanjutan)

**If kondisi Then
Pernyataan;**

```
//jika hanya terdiri satu statement  
if (kondisi) statemen  
//jika terdapat lebih dari satu statement  
if (kondisi) {  
  statemen1;  
  statemen2;  
  .... }  

```



PERCABANGAN GANDA

- Percabangan ganda apabila terdapat DUA alternatif instruksi yang dijalankan. Logika ini memungkinkan kompiler menjalankan salah satu dari DUA alternatif instruksi yang ada, dan salah satu instruksi pasti dijalankan.
- Struktur percabangan sedikit lebih kompleks bila dibandingkan dengan struktur yang hanya memiliki satu buah kondisi.

```
{ statemen_jika_kondisi_terpenuhi; }  
else  
{ statemen_jika_kondisi_tidak_terpenuhi; }
```

PERCABANGAN GANDA

Kondisi/syarat selalu digunakan operator relasional

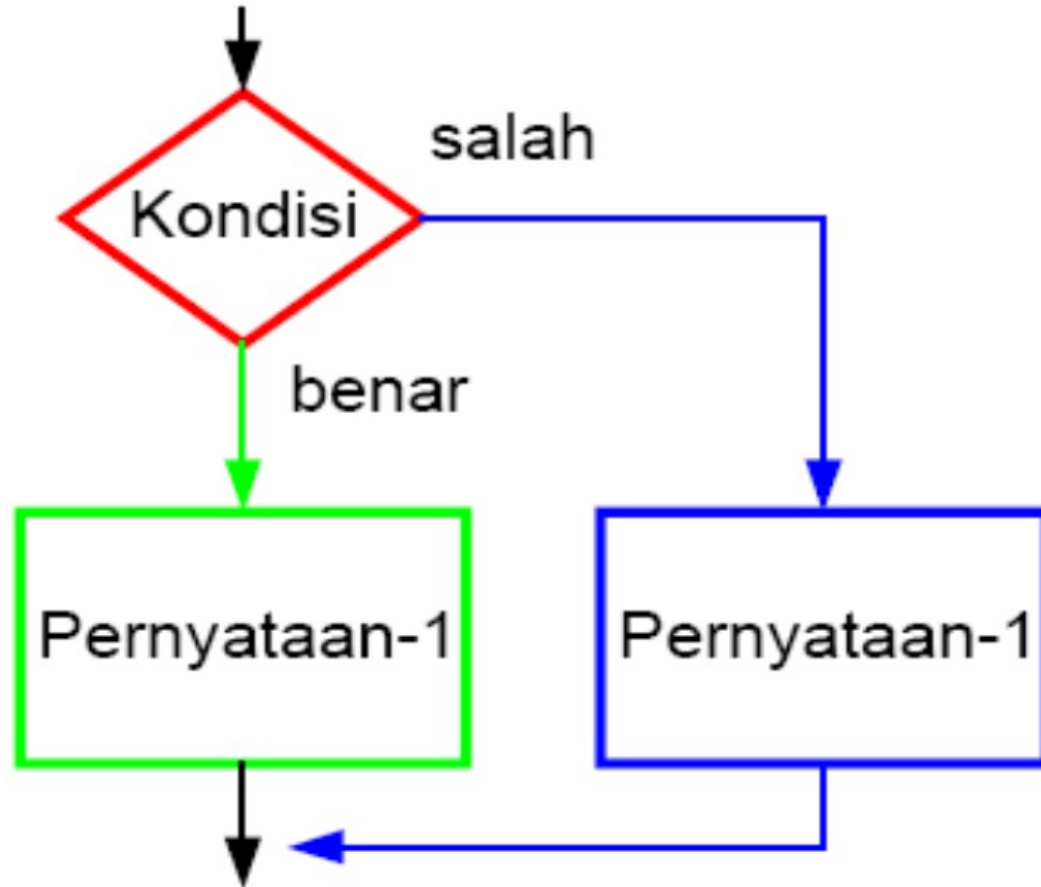
Operator	Arti
>	Lebih besar
<	Lebih kecil
=	Sama dengan
≥	Lebih atau sama
≤	Kurang atau sama
≠	Tidak sama dengan

PERCABANGAN LEBIH DARI DUA

Pada dasarnya hanya terdapat DUA jenis seleksi dalam struktur algoritma pemrograman, Struktur seleksi dapat dikembangkan menjadi bentuk yang tidak terbatas dan dapat dikombinasikan kedalam bentuk perulangan selama notasi penulisannya tidak terdapat kesalahan.

```
if (kondisi_1)
{statemen_jika_kondisi_1_terpenuhi;}
else if (kondisi_2)
{statemen_jika_kondisi_2_terpenuhi;}
else if (kondisi_3)
{statemen_jika_kondisi_3_terpenuhi;}
else
{statemen_jika_semua_kondisi_diatas_tdk_terpenuhi;}
```

PERCABANGAN LEBIH DARI DUA



NESTED IF

Nested if atau struktur if bersyarang atau struktur if dalam if digunakan untuk menyelesaikan masalah-masalah yang memiliki lebih dari dua cabang. Seperti halnya struktur **If-Then** ataupun **If Then-Else**, alternatif-alternatif dalam nested if bisa berupa alternatif hanya terdiri dari satu proses atau aksi (tunggal) atau alternatif yang terdiri dari sejumlah proses atau aksi (jamak) ataupun kombinasi antar keduanya.

SWITCH (1)

- ✓ Selain menggunakan struktur if, percabangan menggunakan switch.
- ✓ struktur percabangan IF, lebih cocok digunakan jika kondisi yang diperiksa sangat banyak
- ✓ Digunakan untuk menyederhanakan **if..else if..**
- ✓ Mempunyai banyak kondisi.
- ✓ Kondisi yang diperiksa harus berupa data ordinal (bertipe integer atau char), dan tidak boleh bertipe real

SWITCH

```
switch (ekspresi) {  
  case nilai_konstanta1 :  
    statemen(pernyataan);  
  break;  
  case nilai_konstanta2 :  
    statemen(pernyataan);  
  break;  
  .....  
  default :  
    statemen(pernyataan)_alternatif;  
}
```



PERBEDAAN IF DAN CASE

- Unggul yang mana antara struktur if dengan struktur case.?
- Sebenarnya kalau dikatakan unggul, keduanya sama-sama unggul. Karena struktur case memiliki kelebihan dari pada struktur if, dan struktur if pun memiliki kelebihan dari pada struktur case.
- Kelebihan statemen case terletak pada struktur script atau codenya yang lebih ringkas dibandingkan dengan struktur pada if.
- Sedangkan kelebihan statemen if yaitu struktur if dapat menyelesaikan hampir seluruh permasalahan, sedang case tidak semua permasalahan dapat diselesaikan dengan ini.

PERBEDAAN IF DAN CASE (Lanjutan)

- **Mengapa statemen if dapat menyelesaikan hampir semua masalah, sedangkan statemen case tidak bisa?**

"Karena pada statemen if dapat menjalankan suatu percabangan di dalam percabangan yang disebut juga dengan istilah if bersarang, dan pada statemen case tidak ada hal yang seperti itu."

Terima kasih