

Algoritma & Pemrograman #4

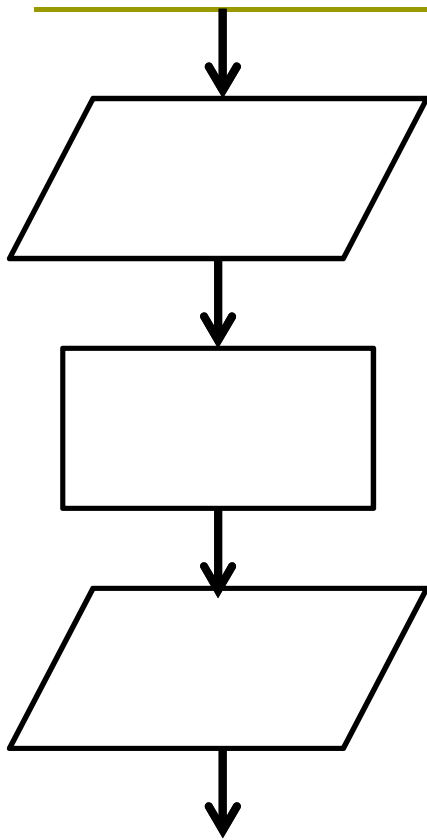
by antonius rachmat c, s.kom, m.cs

Tambahan

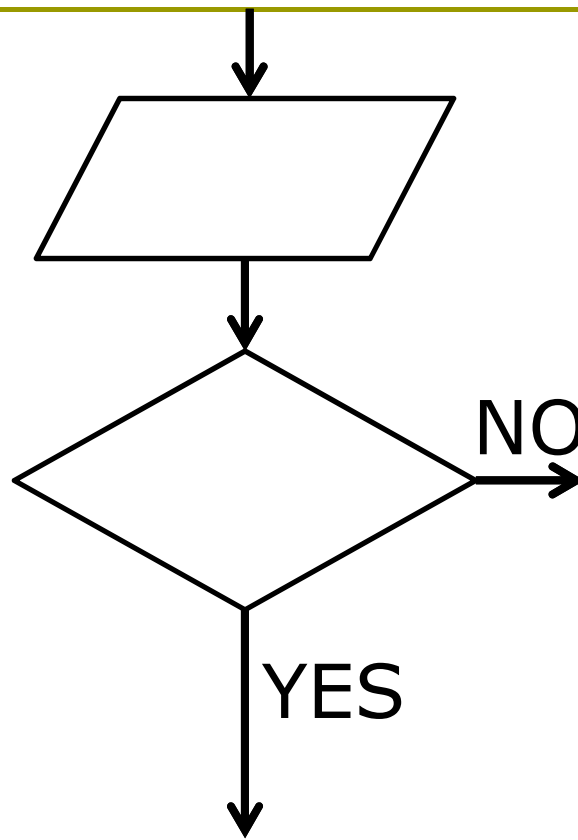
□ Jenis Operator

- Unary Operator -> membutuhkan satu operand
 - Operator Increment dan Decrement
 - -- dan ++
 - Prefiks = ++a, Postfiks = a++
- Binary Operator -> membutuhkan dua operand
 - $x + y$
 - b / d
 - $t - z$
 - $y \% n$
- **“Ternary” Operator (3 operand)**
 - Akan dibahas dibelakang

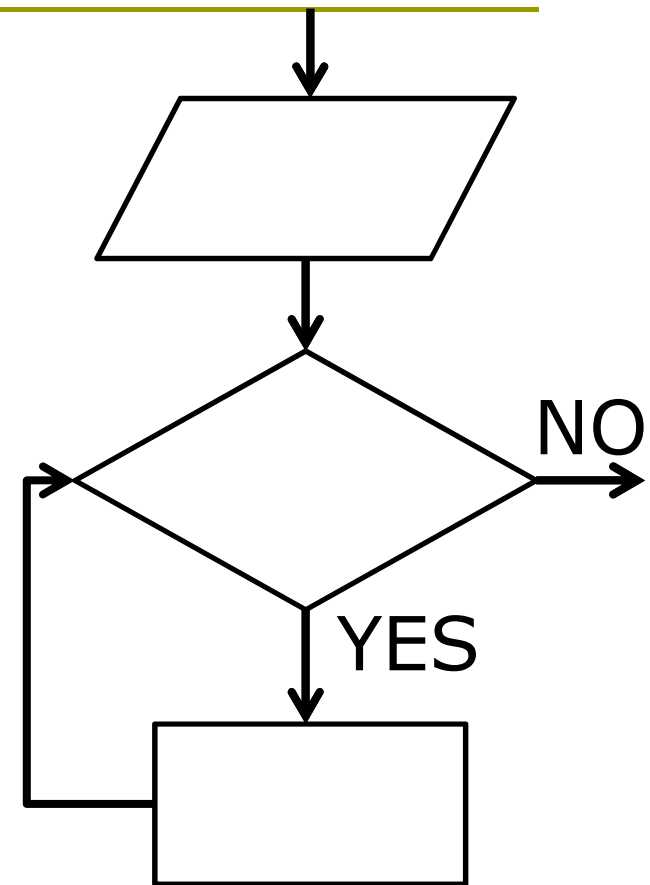
Remember: Struktur Kontrol



Sekuensial



Percabangan



Perulangan

Struktur Kendali

- Adalah statement yang memungkinkan **pemilihan** atas perintah yang akan dijalankan sesuai dengan **kondisi** tertentu.
- Ada tiga macam perintah percabangan dalam C:
 - `if (<kondisi>) { <statement;> ... }`
 - `if (<kondisi>) {
 <statement true; ... >
} else {
 <statement false; ... >
}`
 - `switch (<kondisi>) { case <pil> : <statement;>
 <break;> ... }`

Review jenis operator relasional

- Ada beberapa operator relasional :
 - $>$: lebih besar
 - $>=$: lebih besar atau sama dengan
 - $<$: lebih kecil
 - $<=$: lebih kecil atau sama dengan
 - $==$: sama dengan
 - $!=$: tidak sama dengan

Operator Logical

- Operator Logical AND

A	B	A && B
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

Operator Logical

- Operator Logical OR

A	B	A B
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

Operator Logical

- Operator Logical NOT

A	!A
TRUE	FALSE
FALSE	TRUE

Struktur Kendali

- ❑ Percabangan (**branch**) di dalam pemrograman digunakan oleh komputer untuk menentukan langkah kerja
- ❑ Percabangan menggunakan operator kondisional yang akan menghasilkan nilai **benar/true/1** atau **salah/false/0**.
- ❑ Jika nilai yang dihasilkan **benar**, maka akan perintah akan **dilaksanakan**, sedangkan jika **salah**, maka instruksi **tidak akan dilaksanakan**.

Contoh

□ Contoh perbandingan :

- $5 > 3$ → bernilai benar (TRUE)
- $6 < 2$ → bernilai salah (FALSE)
- $45 \neq 34$ → bernilai benar (TRUE)
- $(5 + 4) \leq (3 \times 3)$ → bernilai benar (TRUE)
- $6/7 \geq 12/17$ → bernilai benar (TRUE)
- $1 == 2$ → bernilai salah (FALSE)
- 1 → bernilai benar (TRUE)
- 0 → bernilai salah (FALSE)

Operator Relasional

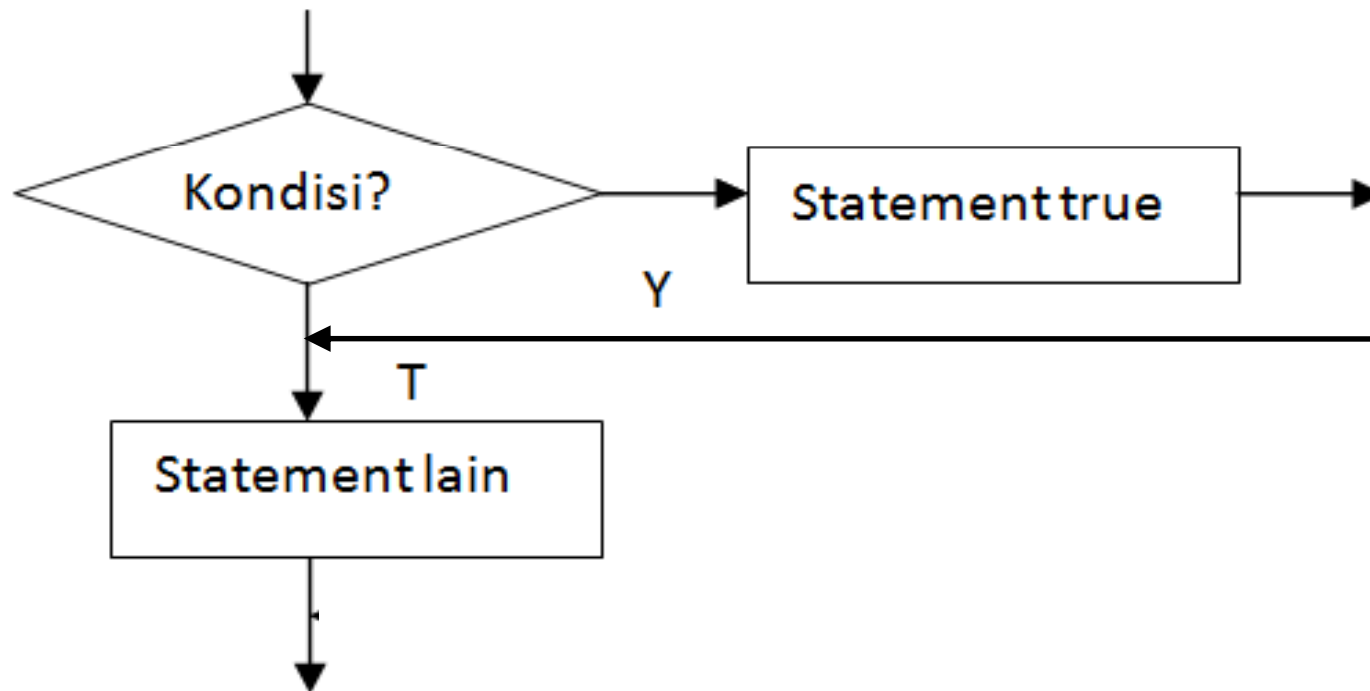
□ Contoh :

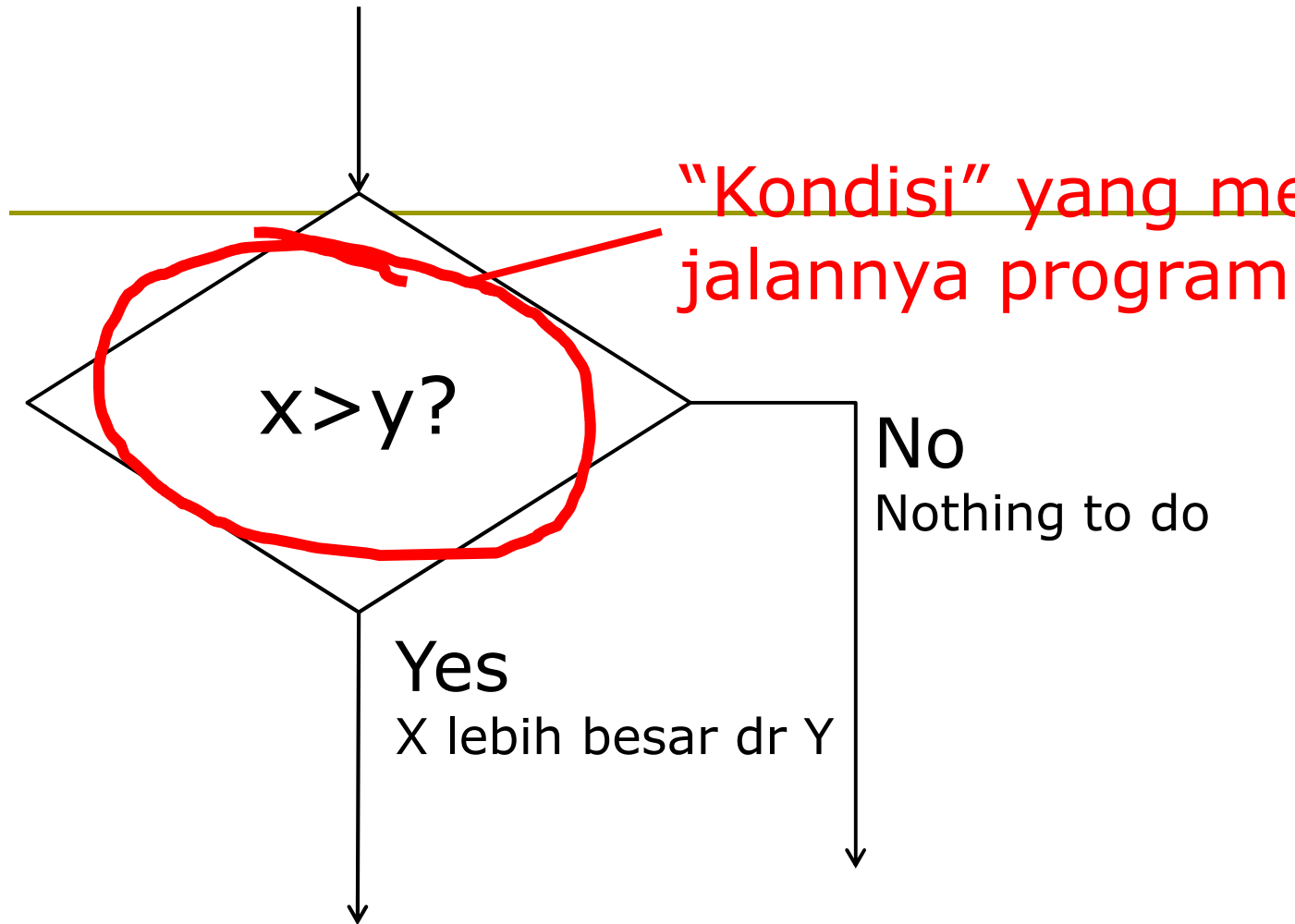
```
3 < 5 bernilai : 1
15 >= 3 * 5 bernilai : 1
8 == 7 bernilai : 0
11<=5+4 bernilai : 0
4!=5 bernilai : 1
```

```
#include <stdio.h>
#include <conio.h>
int main() {
    printf("3 < 5 bernilai : %d\n", 3 < 5);
    printf("15 >= 3 * 5 bernilai : %d\n", 15 >= 3 * 5);
    printf(" 8 == 7 bernilai : %d\n", 8==7);
    printf("11<=5+4 bernilai : %d\n", 11<=5+4);
    printf("4!=5 bernilai : %d\n", 4!=5);
    getch();
}
```

IF sederhana

- Sintaks: **IF (<kondisi>) <statement_jika_benar>;**
- Flowchart:





"Kondisi" yang menentukan jalannya program

$x > y?$

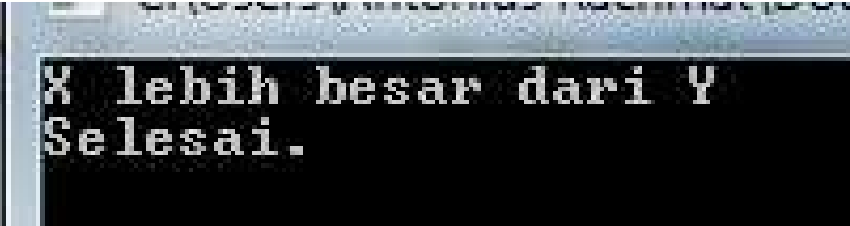
No
Nothing to do

Yes
X lebih besar dr Y

Conditional, Branching, Percabangan

Contoh if sederhana

```
#include <stdio.h>
#include <conio.h>
int main() {
    int X,Y;
    X = 100;
    Y = 10;
    if (X > Y)
    {
        printf("X lebih besar dari Y");
    }
    printf("\nSelesai.");
    getch();
}
```



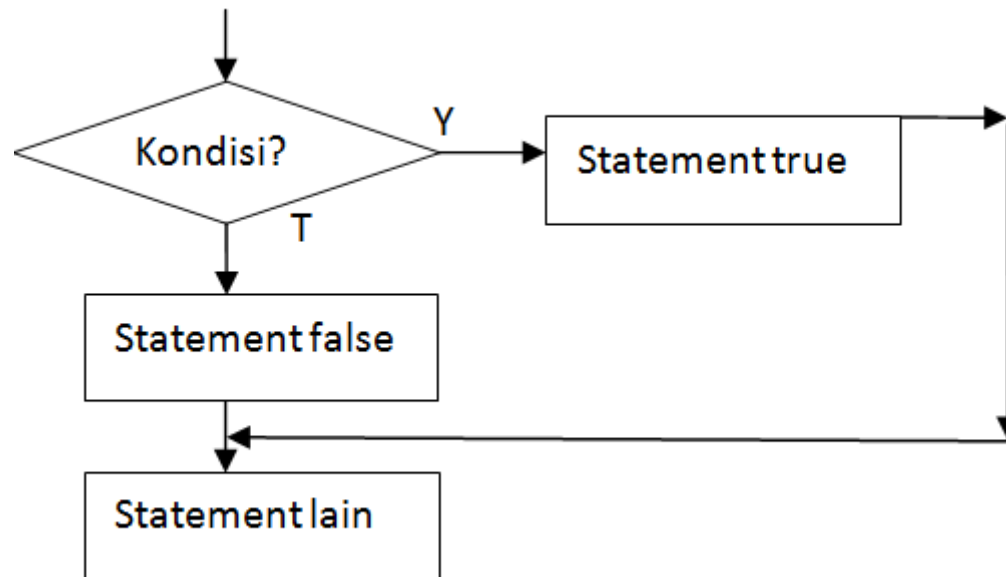
```
X lebih besar dari Y
Selesai.
```

IF ... ELSE ...

- Sintaks:

**IF (<kondisi>) <statement_jika_benar>;
ELSE <statement_jika_salah>;**

- Flowchart:



Contoh if else

```
#include <stdio.h>
#include <conio.h>
int main() {
    int X,Y;
    X = 100;
    Y = 10;
    if (X > Y)
        printf("X lebih besar dari Y");
    else
        printf ("%d tidak lebih besar dari %d",X,Y);
    printf ("\nSelesai.");
    getch();
}
```

```
X lebih besar dari Y
Selesai.
```


Bentuk IF lain (Ternary Operator)

- Sintaks :

<kondisi>?<statement jika benar>:<statement jika salah>;

- Contoh Program:

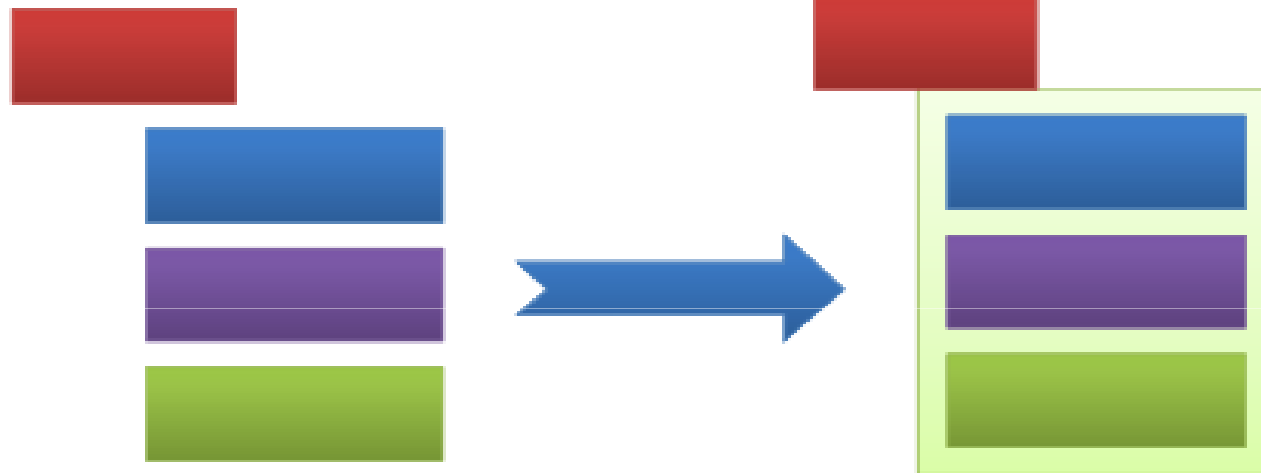
```
#include <stdio.h>
#include <conio.h>
int main() {
    int a;
    10 > 2 && 2 < 1?a=1:a=0;
    printf("%d", a);
    getch();
}
```

- Hasil : **0**

IF ELSE (2)

```
if (<kondisi>) {  
    <statement_benar-1>;  
    <statement_benar-2>;  
    <statement_benar-3>;  
    ...  
    <statement_benar-n>;  
} else {  
    <statement_salah-1>;  
    <statement_salah-2>;  
    <statement_salah-3>;  
    ...  
    <statement_salah-n>;  
}
```

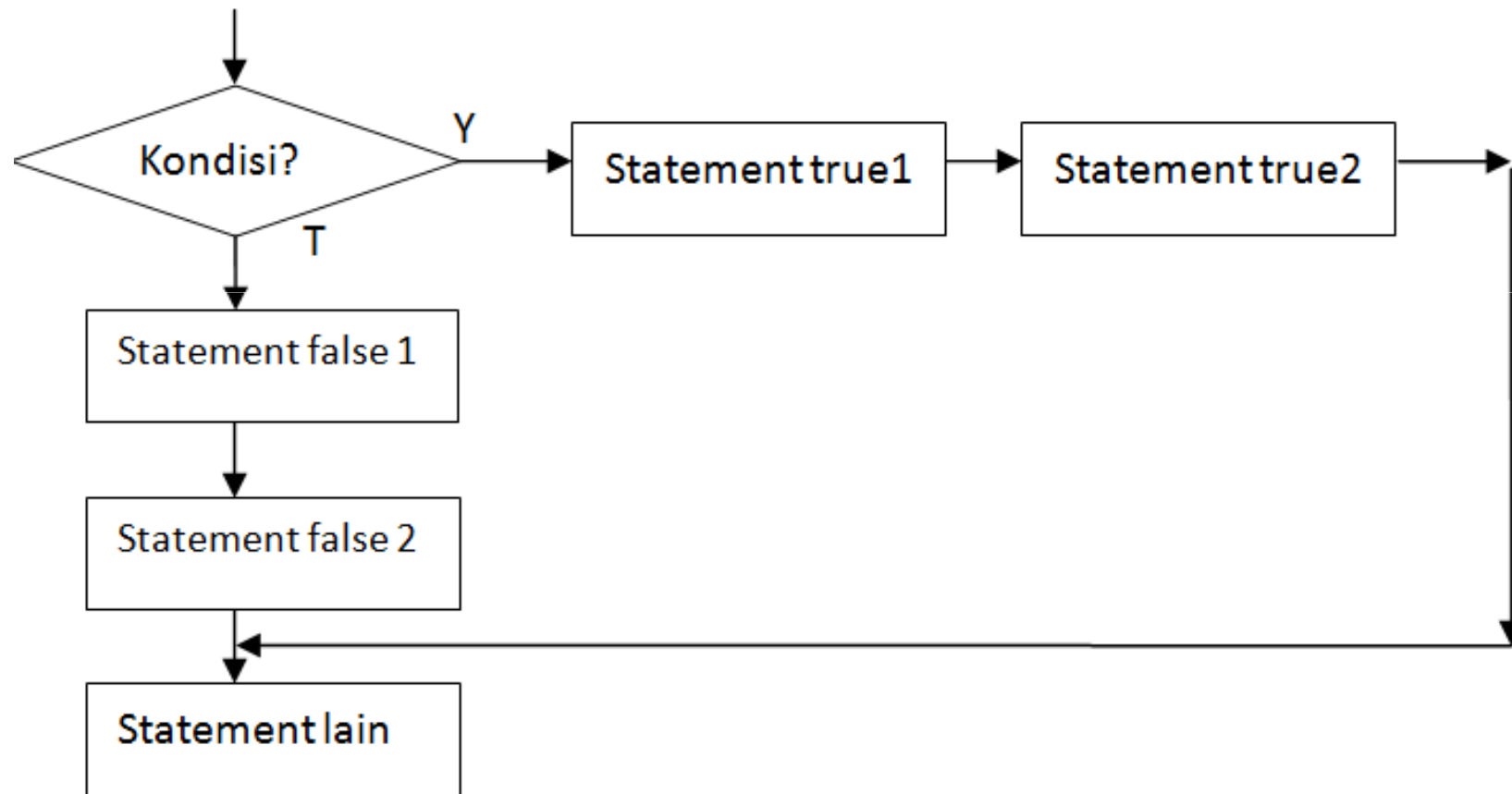
Curly bracket untuk pengelompokkan



```
if (A > B)
  printf("A lebih besar");
  printf("B lebih kecil");
  printf("A dan B berbeda");
```

```
if (A > B) {
  printf("A lebih besar");
  printf("B lebih kecil");
  printf("A dan B berbeda");
}
```

Flowchart IF ELSE banyak



Contoh kasus

```
int harga, diskon, bayar;  
printf("Harga barang: "); scanf("%d", &harga);  
printf("Diskon (%): "); scanf("%d", &diskon);
```

.....????

Diskon hanya untuk harga barang di atas 50.000. Jika harga barang lebih dari 100.000, mendapat tambahan diskon 10%

Harga barang: 40000

Diskon (%): 25%

Total: 40000

Tidak mendapat diskon karena harga \leq 50000

Harga barang: 60000

Diskon (%): 40%

Total: 36000

Diskon 40% karena harga $>$ 50000

Harga barang: 120000

Diskon (%): 25%

Total: 81000

Diskon 25% + 10% karena harga $>$ 100000

Tidak mendapat diskon
karena harga ≤ 50000

IF(harga < 50000)

.....

ELSE IF(harga ≥ 50000 && harga ≤ 100000)

.....

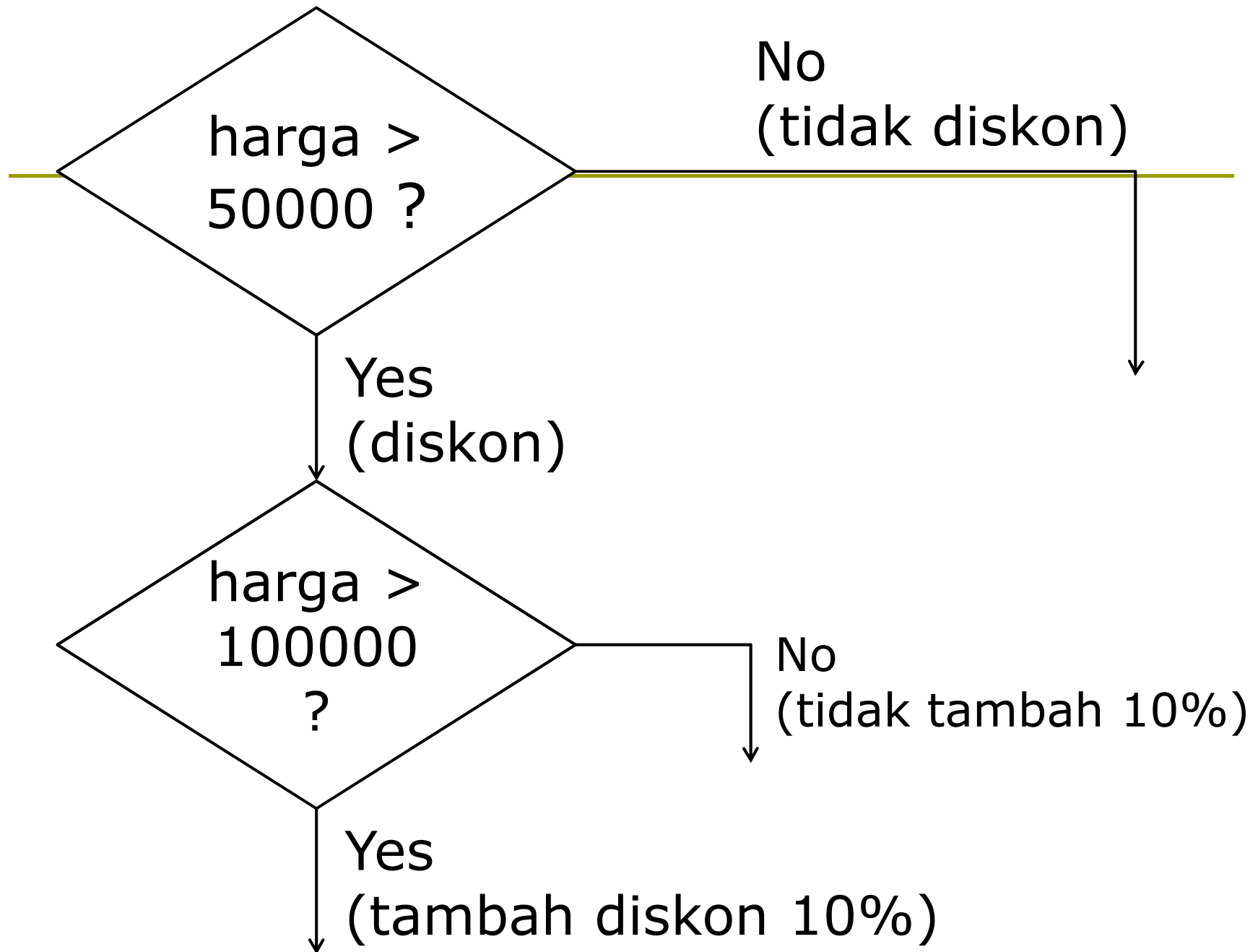
Tidak mendapat tambahan diskon
10% karena harga ≤ 100000

ELSE

.....

ENDIF

mendapat tambahan diskon 10%
karena harga > 100000



```
if(harga < 50000) {
    bayar = harga;
}
else if(harga >= 50000 && harga <= 100000) {
    bayar = harga - (harga * diskon/100.0);
}
else {
    bayar = harga - (harga * diskon/100.0);
    bayar = bayar * 0.9; //tambahan disc 10%
}
```

Sifat Percabangan Lainnya

- Pada bahasa C, kondisi yang bernilai **tidak sama** dengan **0** (nol) berarti bernilai **TRUE**, jika nilainya **0** maka bernilai **FALSE**.
- Contoh:
 - **if (4+5)** → bernilai TRUE

Kesimpulan

- IF dieksekusi jika bernilai **1**
- Jika **a** bernilai positif atau negatif maka:
 - if(a) berarti **1**, dieksekusi
- Jika **a** bernilai nol, maka:
 - if(a) berarti **0**, tidak dieksekusi
- Jika **a** bernilai **0**, maka:
 - if(a == 0) berarti **1**, dieksekusi
 - if(!a) berarti **1**, dieksekusi

Kesalahan dalam IF

Semua menghasilkan TRUE

- ❑ Kondisi pada statement if bukan merupakan nilai **integer**

```
int main()
{
    char k='a';
    if(k) printf("a "); else printf("bukan a ");
    float a = 2.0;
    if(a) printf("2.0 "); else printf("bukan 2.0 ");
    getch();
}
```

- ❑ Menggunakan = bukan ==

```
int main()
{
    char k='a';
    if(k='b') printf("a "); else printf("bukan a ");
    float a = 2.0;
    if(a=1) printf("1.0 "); else printf("bukan 1.0 ");
    getch();
}
```

- ❑ Menggunakan **elseif** bukan **else if -> error!**

Soal-soal

- ❑ Buatlah program untuk memeriksa bilangan genap atau ganjil
- ❑ Buatlah program untuk memeriksa bilangan terbesar dari 2 bilangan
- ❑ Buatlah program untuk memvalidasi umur (1-100 tahun)!
- ❑ Buatlah program untuk mengetahui bentuk suatu zat (beku atau tidak!)
- ❑ Buat program yang meminta tiga input bilangan bulat (int). Program akan menampilkan tulisan "BISA" jika bilangan 1 dan bilangan 2 dijumlahkan hasilnya sama dengan input bilangan ketiga!

Soal-soal

- ❑ Buatlah program untuk mengecek apakah suatu segitiga sama sisi atau bukan!
- ❑ Buatlah program untuk mengetahui **kuadran** dari inputan koordinat x dan y!
- ❑ Buatlah program untuk mencari bilangan terbesar dari 3 bilangan yang diinputkan dengan menggunakan IF!
- ❑ Buatlah program untuk memeriksa apakah suatu bilangan adalah kelipatan 11 atau bukan!
- ❑ Buat program untuk memvalidasi agar pembagian tidak dgn nol!

Percabangan Kompleks

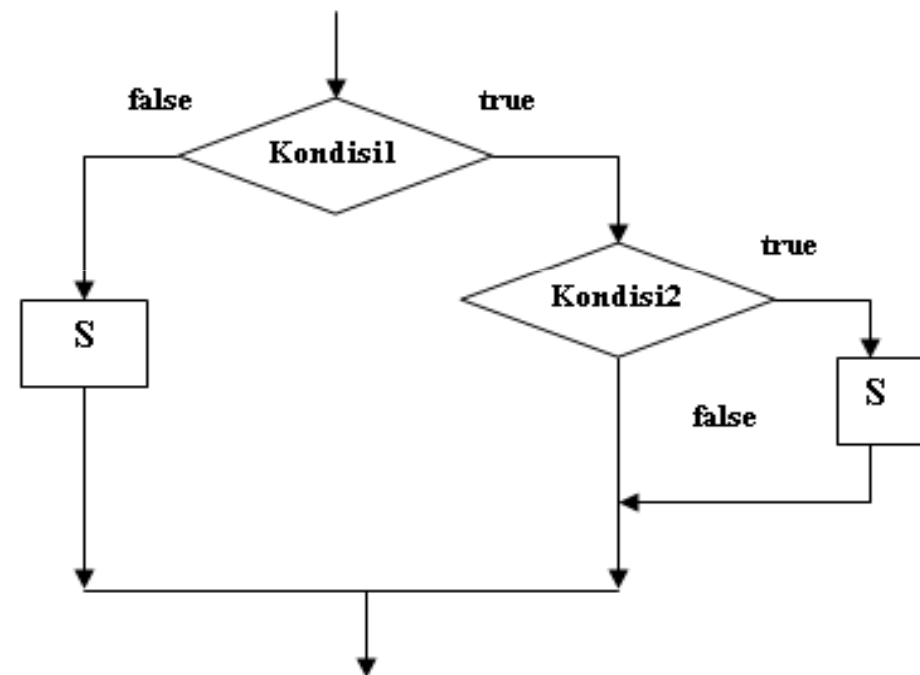
- Percabangan dimana kondisi pemilihan **tidak hanya satu** tetapi bisa terdiri dari banyak alternatif serta perintah-perintah yang dikerjakannya juga bisa lebih dari satu.

NESTED IF (IF BERSARANG)

Contoh 1 :

```
if (kondisi1)
{
    if (kondisi2)
    {
        S;
        S;
        ...
    }
}
else
{
    S;
    S;
    ...
}
```

Flowchart



Keterangan:

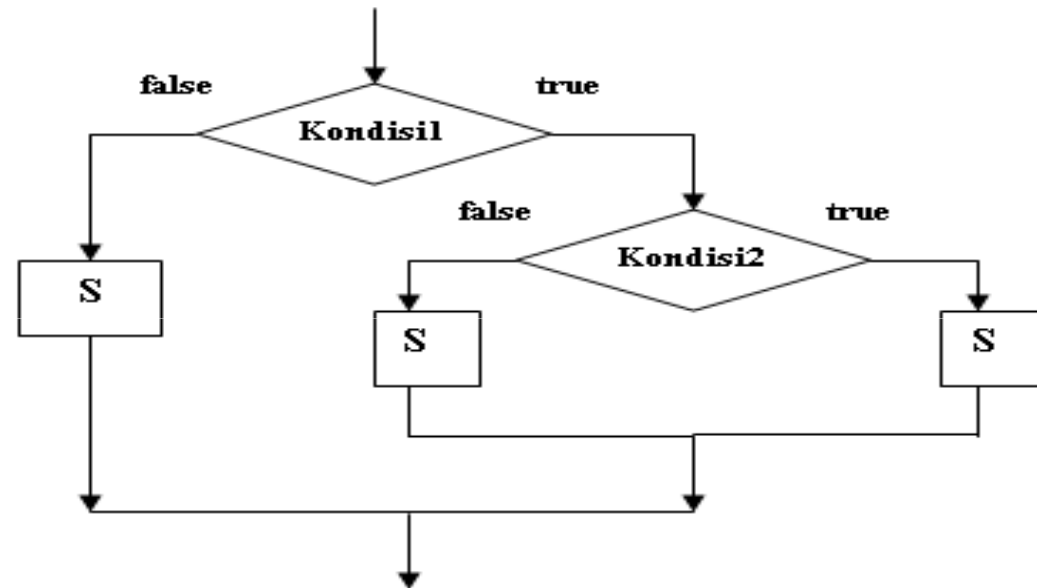
S adalah Statement / Program

Nested IF (2)

Contoh 2 :

```
if (kondisi1)
{
    if (kondisi2)
    {
        S;
        S;
        ...
    }
    else
    {
        S;
        S;
        ...
    }
}
else
{
    S;
    S;
    ...
}
```

Flowchart:

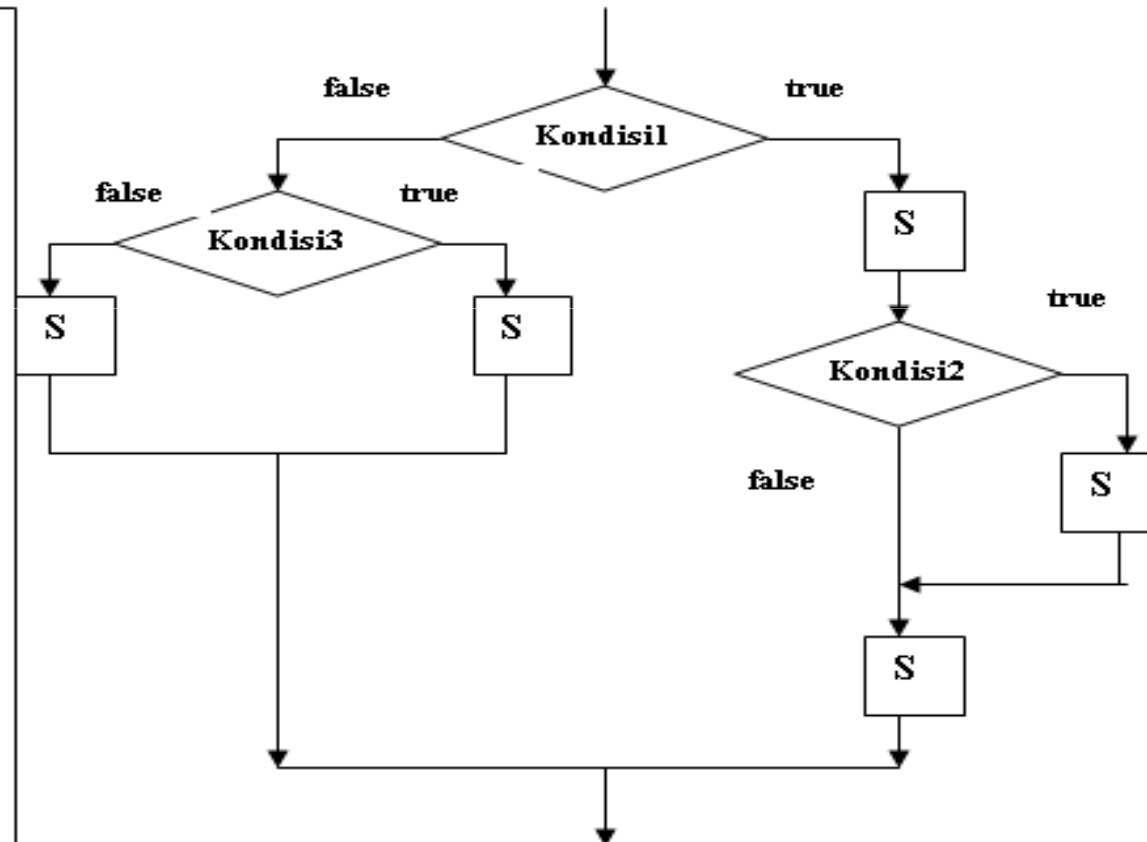


Nested IF (3)

Contoh 3:

```
if (kondisi1)
{
    S;
    if (kondisi2)
    {
        S;
    }
    S;
}
else
{
    if (kondisi3)
    {
        S;
    }
    else
    {
        S;
    }
}
```

Flowchart:

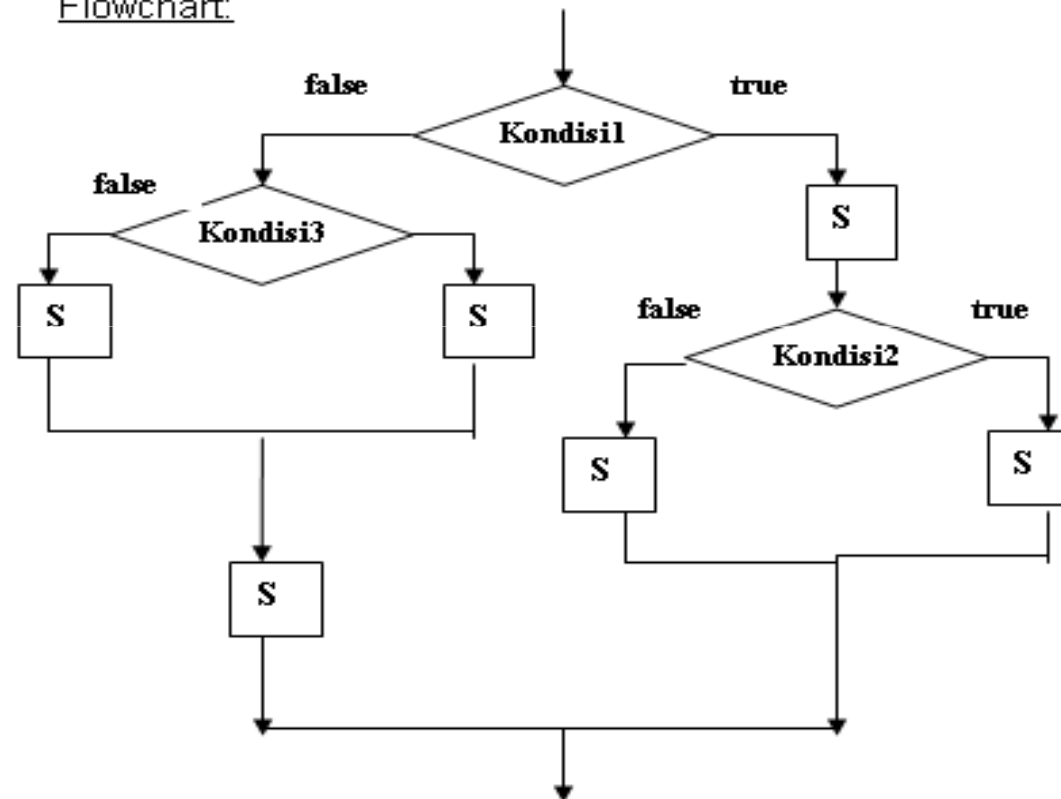


Nested IF (4)

Contoh 4:

```
if (kondisi1)
{
    S;
    if (kondisi2)
    {
        true
        S;
    }
    else
    {
        S;
    }
}
else
{
    if (kondisi3)
    {
        S;
    }
    else
    {
        S;
    }
}
S;
```

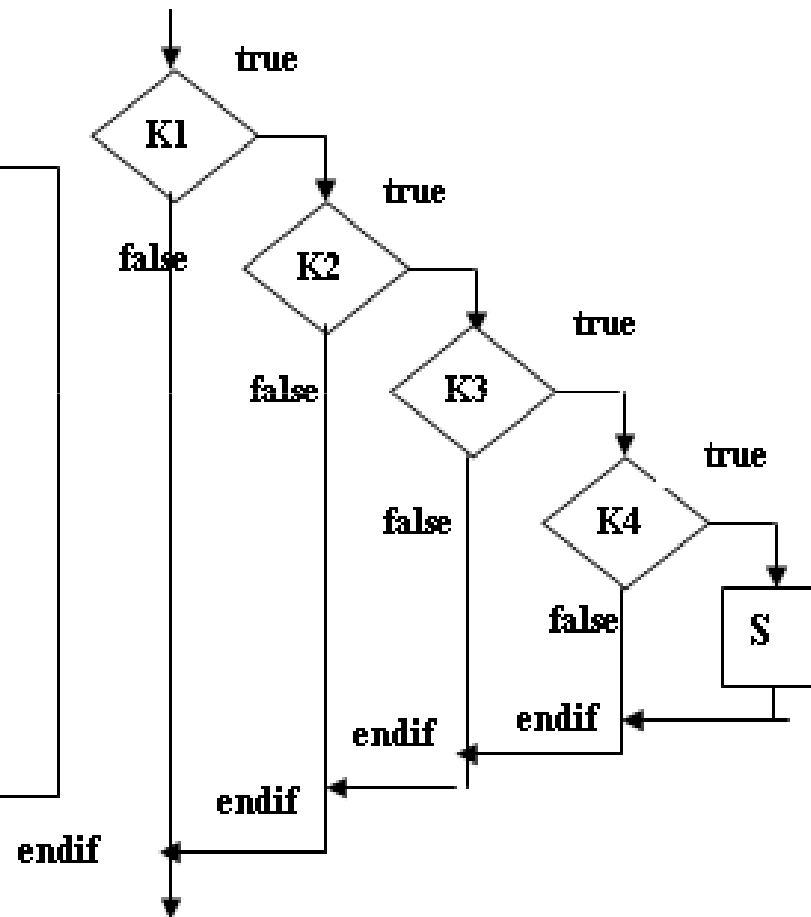
Flowchart:



Nested IF (5)

Contoh 5:

```
if (kondisi1)
{
    if (kondisi2)
    {
        if (kondisi3)
        {
            if (kondisi4) {
                S;
            }
        }
    }
}
```

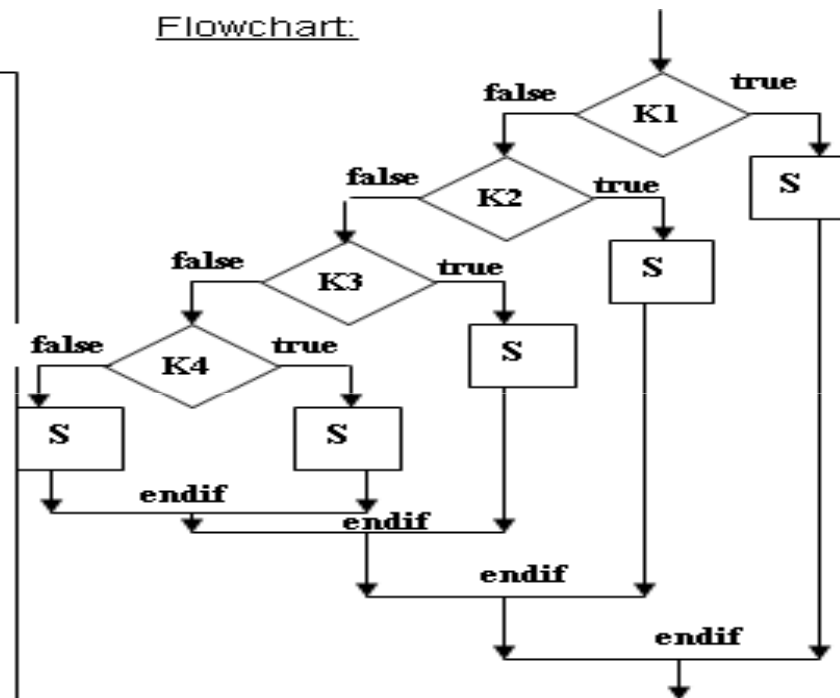


Nested IF (6)

Contoh 6:

```
if (kondisi1)
{
    S;
}
else
{
    if (kondisi2)
    {
        S;
    }
    else
    {
        if (kondisi3)
        {
            S;
        }
        else
        {
            if (kondisi4)
            {
                S;
            }
            else
            {
                S;
            }
        }
    }
}
}
```

Flowchart:



IF Bertingkat dengan ELSE

```
if (kondisi1){  
    instruksi1;  
}  
else if(kondisi2){  
    instruksi2;  
}  
else if(kondisi3){  
    instruksi3;  
}  
else if(kondisi4){  
    instruksi4;  
}
```

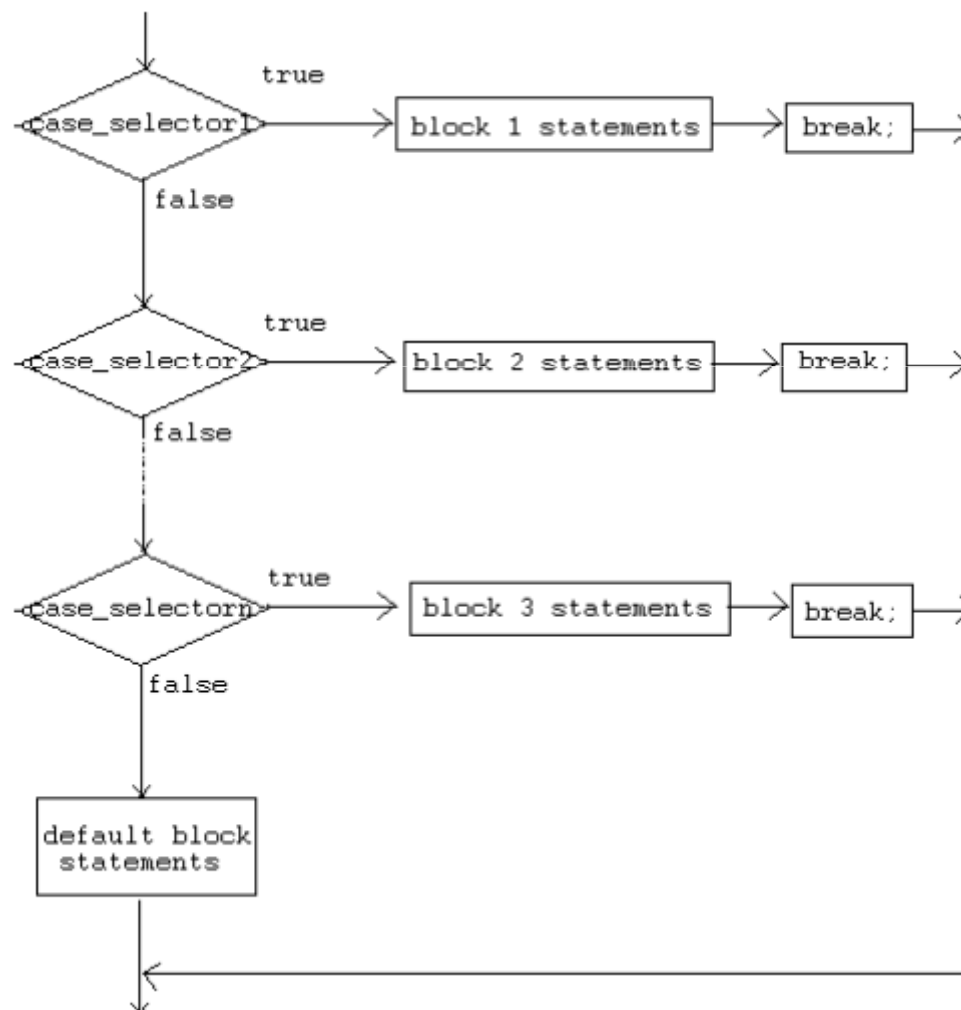
Bedakan dengan tanpa ELSE:

```
if (kondisi1){  
    instruksi1;  
}  
if(kondisi2){  
    instruksi2;  
}  
if(kondisi3){  
    instruksi3;  
}  
if(kondisi4){  
    instruksi4;  
}
```

Percabangan dengan Switch

- ❑ Perintah ini digunakan sebagai alternatif pengganti dari sintaks **if ... else** secara sederhana dimana alternatif pilihan bisa lebih dari satu.
- ❑ **Namun** switch tidak bisa digunakan untuk pilihan yang mengandung suatu kondisi tertentu.
- ❑ Dengan perintah ini program percabangan akan semakin mudah dibuat dan dipelajari

Flowchart Switch



Karakteristik Switch

- ❑ Perintah **switch** akan menyeleksi kondisi yang diberikan dan kemudian membandingkan hasilnya dengan **konstanta-konstanta** yang berada di *case*.
- ❑ Perbandingan akan dimulai dari konstanta ke-1 sampai konstanta terakhir.
- ❑ Jika hasil dari perbandingan kondisi sama dengan nilai konstanta tertentu, misalnya konstanta ke-1, maka pernyataan ke-1 akan dijalankan sampai ditemukan perintah **break**.
- ❑ Pernyataan *break* akan membawa proses **keluar** dari perintah *switch*.
- ❑ Jika hasil dari kondisi tidak ada yang sama dengan konstanta-konstanta yang diberikan, maka pernyataan pada **default** akan dijalankan

.....
printf("Angka: "); scanf("%d", &angka);
switch(angka) {
 case 10: printf("WOHOO!"); break;
 case 20: printf("YIHAA"); break;
 case 30: printf("AUOUO"); break;
 case 40: printf("BOOOO"); break;
 case 50: printf("HOHOHO"); break;
}

.....
angka = 20?40? 50? 60?

```
.....  
printf("Angka: "); scanf("%d", &angka);  
switch(angka) {  
    case 10: printf("WOHOO!"); break;  
    case 20: printf("YIHAA"); break;  
    case 30: printf("AUOUO");  
    case 40: printf("BOOOO");  
    case 50: printf("HOHOHO"); break;  
    default: printf("DEFAULT");  
}
```

```
.....  
angka = 20?30? 50? 60?
```

Program dengan Switch

```
#include <stdio.h>
#include <conio.h>

int main()
{
    int hari = 5;
    clrscr();
    printf( "Hari ini adalah hari " );
    switch(hari)
    {
        case 1:
            printf("Minggu");
            break;
        case 2:
            printf("Senin");
            break;
        case 3:
            printf("Selasa");
            break;
        case 4:
            printf("Rabu");
            break;
        case 5:
            printf("Kamis");
            break;
        case 6:
            printf("Jumat");
            break;
        case 7:
            printf("Sabtu");
            break;
        default:
            printf("Tidak ada hari lebih dari 7");
    }
    printf("\n");
}
```

Latihan

- Buatlah program untuk mengetahui bilangan “tengah” dari 3 bilangan:
 - Input: 23, 12, 9
 - Maka output: **12**

Latihan

- Buatlah program yang menerima inputan 3 buah panjang sisi segitiga, kemudian:
 - Jika segitiga tersebut sama sisi katakan SEGITIGA SAMA SISI
 - Jika segitiga tersebut sama kaki katakan SEGITIGA SAMA KAKI
 - Jika bukan keduanya katakan SEGITIGA SEMBARANG

Latihan

- Buatlah program diskon:
 - Inputan adalah jumlah total pembelian dan jumlah barang.
 - Jika total pembelian ≥ 500000 dan jumlah barang > 5 maka bonusnya Setrika
 - Jika total pembelian ≥ 100000 dan jumlah barang > 3 maka bonusnya Payung
 - Jika total pembelian ≥ 50000 atau jumlah barang > 2 maka bonusnya ballpoint
 - Selainnya itu tidak mendapat bonus.

Latihan

- Mencari bilangan terkecil dari 4 bilangan yang diinputkan!
- Mencari bilangan terbesar dari 4 bilangan yang diinputkan!
- Mengatahui apakah suatu bilangan 0, positif, atau negatif!

Latihan

- Buatlah konversi nilai huruf dari nilai yang diinputkan user!
 - A 80-100
 - B 65-79
 - C 50-64
 - D 35-49
 - E 0-34

Latihan

- Buat program untuk menghitung luas segitiga, luas lingkaran, dan luas persegi panjang dengan menggunakan sistem menu.
 - Misal:
 - Luas Segitiga
 - Luas Lingkaran
 - Luas Persegi Panjang
 - Pilihan anda: _

Soal-soal

- Membuat konversi dari bilangan yang diinputkan sebagai berikut:
 - Misal 101 : bilangan ratusan
 - Misal 1200 : bilangan ribuan
 - Misal 11 : bilangan puluhan, dan seterusnya!

NEXT

- Perulangan Sederhana