



UNIVERSITAS
ATMA JAYA YOGYAKARTA
serviens in lumine veritatis




Dasar Pemrograman (INFT06204)

Minggu 5
Pemilihan





Operator Precedence

Operator	Precedence
function calls	highest
! + - & (unary operators)	
* / %	
+ -	
< <= >= >	
== !=	
&&	
=	
	lowest



x	y	z	flag
3.0	4.0	2.0	0

1. !flag	/* !0 is 1 (true) */
2. x + y / z <= 3.5	/* 5.0 <= 3.5 is 0 (false) */
3. !flag (y + z >= x - z)	/* 1 1 is 1 (true) */
4. !(flag (y + z >= x - z))	/* !(0 1) is 0 (false) */



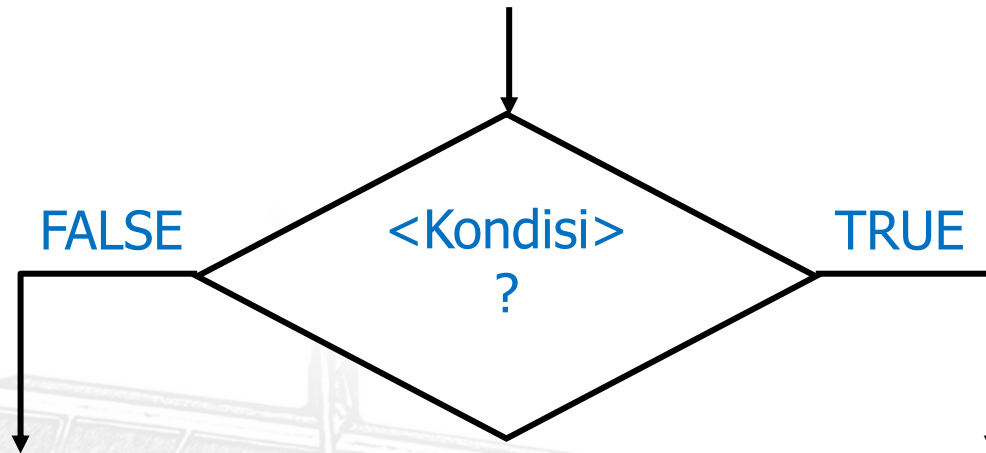
PEMILIHAN

- Di dalam program ada kalanya suatu aksi hanya akan dilakukan hanya jika memenuhi persyaratan atau kondisi tertentu
 - Jadi kita harus menganalisis kasus-kasus yang mungkin ada
- Mendefinisikan analisis kasus adalah mendefinisikan:
 - **Kondisi**, yang berupa ekspresi yang menghasilkan true atau false
 - **Aksi**, yang akan dilaksanakan jika kondisi yang dipasangkan dengan aksi ybs terpenuhi



PEMILIHAN

- Simbol :



- Kondisi berupa ekspresi relasional
(operator : ==, <, >, <=, >=, !=, &&, ||, !)



PEMILIHAN

- Operator **!** digunakan untuk membuat komplemen (hal yang berkebalikan) dari suatu kondisi *boolean*

if (! <ekspresi>)

<aksi>

→ aksi dilakukan jika ekspresi menghasilkan nilai *boolean* false



PEMILIHAN

- Operator **&& (AND)** digunakan untuk menghubungkan suatu kondisi *boolean* yang terdiri dari beberapa ekspresi relasional, yang akan bernilai true jika SEMUA ekspresi menghasilkan nilai TRUE

if (<ekspresi1> **&&** <ekspresi2>)
 <aksi>

→ aksi dilakukan jika ekspresi1 bernilai true dan ekspresi2 bernilai true



PEMILIHAN

- Operator **|| (OR)** digunakan untuk menghubungkan suatu kondisi *boolean* yang terdiri dari beberapa ekspresi relasional, yang akan bernilai true jika SALAH SATU/LEBIH ekspresi menghasilkan nilai TRUE

if (<ekspresi1> **||** <ekspresi2>)
 <aksi>

→ aksi dilakukan jika salah satu atau kedua ekspresi bernilai true



PEMILIHAN

- Penentuan kondisi boolean dan aksi yang dilakukan bergantung pada jumlah kasus yang terdapat pada masalah tersebut :
 - Satu kasus
 - Dua kasus
 - Tiga kasus atau lebih



Satu Kasus

```
if (<kondisi>)  
    <aksi>
```

- *Contoh:*

```
if (nilai < 0)
```

```
    printf("Nilai masukan harus lebih besar dari nol.");
```



Dua Kasus

```
if (<kondisi>
    <aksi1>
else
    <aksi2>
```

- *Contoh:*

```
if (nilai % 2 == 0)
    printf("Bilangan genap");
else
    printf("Bilangan ganjil");
```



Tiga atau Lebih Kasus

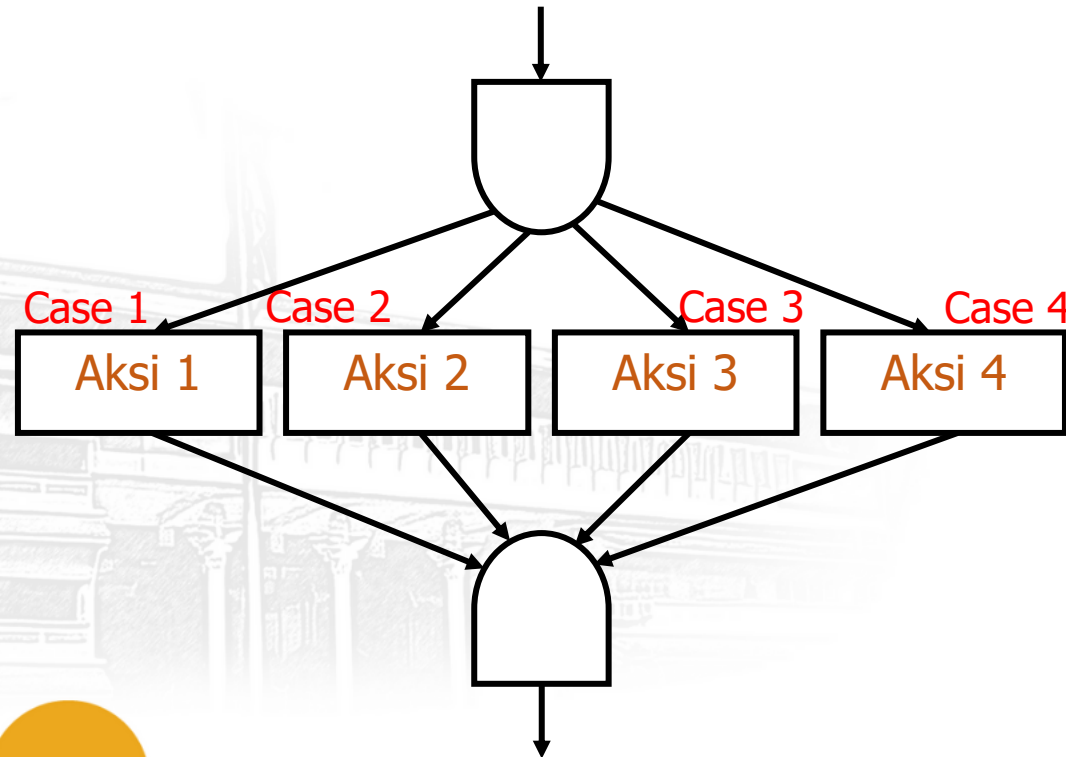
```
if (<kondisi1>
    <aksi1>
else
    if (<kondisi2>)
        <aksi2>
    else
        <aksi3>
```

- *Contoh:*
if (nilai % 2 == 0)
 printf("Bilangan genap");
else
 printf("Bilangan ganjil");



Struktur switch

- Untuk masalah dengan tiga kasus atau lebih, struktur SWITCH dapat menyederhanakan penulisan IF-ELSE yang bertingkat-tingkat.
- Simbol :





Struktur switch

- Struktur :

```
switch(nama)
{
    case <kondisi1> : <aksi1>
                    break;
    case <kondisi2> : <aksi2>
                    break;
    default          : <aksiN>
}
```



Struktur switch

- Struktur switch dalam C hanya dapat digunakan jika kondisi berupa ekspresi yang operatornya adalah “==” dan nilai yang ada di ruas kanan adalah **integer** atau **character**.





Struktur switch

Contoh:

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

```
void main()  
{
```

```
    int No_Hari;
```

```
    printf("Masukkan nomor hari (1-7) : "); scanf("%d", &No_Hari);
```

```
    switch (No_Hari)
```

```
    {
```

```
        case 1 : printf("Senin \n"); break;
```

```
        case 2 : printf("Selasa \n"); break;
```

```
        case 3 : printf("Rabu \n"); break;
```

```
        case 4 : printf("Kamis \n"); break;
```

```
        case 5 : printf("Jumat \n"); break;
```

```
        case 6 : printf("Sabtu \n"); break;
```

```
        case 7 : printf("Minggu \n"); break;
```

```
        default : printf("Bukan hari yang benar \n");
```

```
    }
```

```
}
```



Question and Answer



Terima kasih



uajy



Universitas Atma Jaya Yogyakarta



www.uajy.ac.id