

[Home](#) / [Tutorial](#) / [C](#) / [Pernyataan if](#)

Pernyataan if

Posted on Januari 29, 2018 | Last Modified Januari 30, 2018

Sebuah program akan menjalankan sebuah tugas tertentu jika memenuhi sebuah kondisi. Contoh sederhana: jika sebuah nilai merupakan bilangan genap, maka program akan mencetak ke layar “bilangan genap” dan begitu juga jika nilai tersebut merupakan bilangan ganjil, maka program tersebut akan mencetak ke layar “bilangan ganjil”. Contoh tersebut merupakan contoh pengambilan keputusan dimana program akan melakukan sesuatu sesuai dengan syarat/kondisi yang sudah ditetapkan. Komputer tidak akan dapat mengambil keputusan secara otomatis, maka diperlukan sebuah pernyataan, salah satunya pernyataan `if`.

Pernyataan `if` merupakan salah satu pernyataan yang digunakan untuk memberikan pilihan pada komputer untuk mengambil keputusan sesuai dengan kondisi/syarat yang sudah ditentukan.

Berikut struktur pernyataan `if` :

```
1 if(ekspresi)
2     //tuliskan perintah di sini
```

atau

```
1 if(ekspresi) {
2     //tuliskan perintah di sini
3 }
```

Jika perintah yang akan dijalankan dalam pernyataan `if` hanya **SATU BARIS**, maka buka/tutup kurung kurawal pada pernyataan `if` dapat **DIABAIKAN** dan apabila perintah yang akan dijalankan dalam pernyataan `if` lebih dari satu baris, maka harus menggunakan buka/tutup kurung kurawal untuk menyatakan perintah-perintah tersebut adalah block pernyataan `if`. Ekspresi yang ada dalam pernyataan `if` adalah sebuah syarat yang harus dipenuhi untuk menjalankan perintah-perintah yang ada di dalam pernyataan `if`. Ekspresi ini hanya menghasilkan dua nilai, yaitu **true** (benar) atau **false** (salah).

contoh:

`1 < 5`

Ekspresi ini akan menghasilkan true, karena ekspresinya benar bahwa 1 lebih kecil dari 5

`1 < 0`

Ekspresi ini akan menghasilkan false karena ekspresinya salah bahwa 1 tidaklah lebih kecil dari 0

true dan false dalam bahasa C dinyatakan dalam angka, dimana true itu adalah angka **BUKAN NOL** seperti 1, 2, 3, -1 dan sebagainya sedangkan false adalah angka **NOL**. Jadi, dari contoh sebelumnya, maka `1 < 5` akan menghasilkan 1 dan `1 < 0` akan menghasilkan 0.

Untuk lebih memahami pernyataan `if` mari lihat contoh di bawah.

```
1 #include <stdio.h>
2
3 int main() {
4     int nilai;
5
6     nilai = 10;
7
8     if(nilai % 2 == 0)
9         printf("bilangan genap\n");
10
11    if(nilai % 2 == 1)
12        printf("bilangan ganjil\n");
13
14    return 0;
15 }
```

Penjelasan:

Apa yang dilakukan pada kode di atas adalah mencetak ke layar tulisan `bilangan genap` jika variabel `nilai` berisi bilangan genap dan mencetak ke layar `bilangan ganjil` jika variabel `nilai` berisi bilangan ganjil. Kondisi yang ada dalam pernyataan `if` adalah ekspresi `nilai % 2 == 0`, dimana variabel `nilai` dimodulo dengan 2. Jika hasil variabel `nilai` modulo 2 sama dengan 0, maka ekspresi akan menghasilkan 1 (true) dan sebaliknya jika hasilnya 1, maka ekspresi akan menghasilkan 0 (false). Operator `%` adalah operator modulo, untuk mencari hasil sisa bagi. Jika ekspresi tersebut menghasilkan 1 (true), maka perintah yang ada di dalam `if` akan dijalankan dan sebaliknya. Dalam kasus source di atas, program akan mencetak `bilangan genap` karena variabel `nilai` berisi bilangan genap, yaitu 10 dan sisa bagi dari 10 bagi 2 adalah 0 dan menghasilkan 1 (true).

if-else

else adalah kata kunci yang digunakan untuk menjalankan sebaris atau blok perintah jika ekspresi yang ada dalam pernyataan **if** tidak memenuhi syarat.

Berikut bentuk pernyataan **if-else** .

```
1 if(ekspresi) {  
2     //tuliskan perintah di sini  
3 } else {  
4     //tuliskan perintah di sini  
5 }
```

Sama seperti aturan sebelumnya, jika perintah yang ada di dalam **else** hanya satu baris, buka/tutup kurung kuwaral dapat diabaikan.

Contoh:

```
1 #include <stdio.h>  
2  
3 int main() {  
4     int nilai = 10;  
5  
6     if(nilai % 2 == 0)  
7         printf("bilangan genap");  
8     else  
9         printf("bilangan ganjil");  
10  
11     return 0;  
12 }
```

Penjelasan:

Jika ekspresi pernyataan **if** pada baris ke-6 menghasilkan 0 (false), maka perintah pada baris ke-7 tidak akan dijalankan dan perintah pada baris ke-9 lah yang akan dijalankan.

Dengan bantuan **else** , pernyataan **if** dapat dibuat bertingkat seperti di bawah.

```
1 if(ekspresi) {  
2  
3 } else if(ekspresi) {  
4  
5 } else if(ekspresi) {  
6  
7 } else {  
8  
9 }
```

Contoh:

```
1 #include <stdio.h>
2
3 int main() {
4     int nilai = 65;
5
6     if(nilai >= 90) {
7         printf("Nilai huruf: A");
8     } else if(nilai >= 75) {
9         printf("Nilai huruf: B");
10    } else if(nilai >= 60) {
11        printf("Nilai huruf: C");
12    } else if(nilai >= 40) {
13        printf("Nilai huruf: D");
14    } else {
15        printf("Nilai huruf: E");
16    }
17
18    return 0;
19 }
```

statement **if-else** juga dapat dibuat bersarang seperti di bawah.

```
1 if(ekspresi) {
2     if(ekspresi) {
3
4     }
5 } else if(ekspresi) {
6     if(ekspresi) {
7         if(ekspresi) {
8
9         }
10    }
11 } else {
12     if(ekspresi) {
13
14     } else {
15
16     }
17 }
```

Ternary Operator

Ternary operator (?:) adalah operator yang digunakan untuk mempersingkat pernyataan **if-else**.

Format ternary operator:

(ekspresi)? (A): (B)

keterangan:

A adalah nilai yang akan dikembalikan jika ekspresi true.

B adalah nilai yang akan dikembalikan jika ekspresi false.

dimana A dan B **HARUS** memiliki **TIPE DATA** yang **SAMA**.

Perhatikan pernyataan **if-else** di bawah.

```
1 if(nilai % 2 == 0) {
2     printf("bilangan genap");
3 } else {
4     printf("bilangan ganjil");
5 }
```

Pernyataan **if-else** di atas dapat disingkat dengan ternary operator menjadi.

```
1 printf("%s", ((nilai % 2 == 0)? ("bilangan genap"): ("bilangan ganjil"))
```

Ternary operator juga dapat dibuat bersarang seperti **if-else**.

(ekspresi 1)?((ekspresi 2)? (A): (B)): (C)

f Share

🐦 Tweet

G+ Share

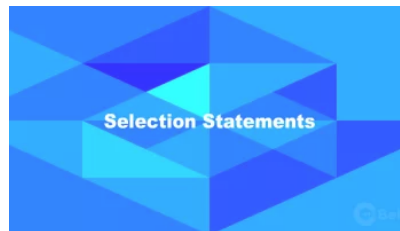
📌 Pin

in Share

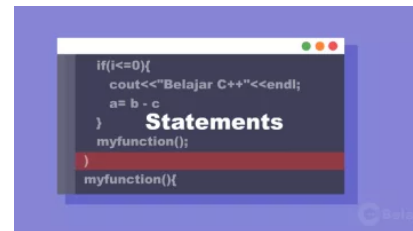
Terkait



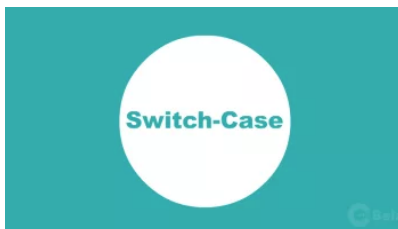
Pernyataan IF
Januari 23, 2018
dalam "C++"



Pernyataan Penyeleksian
Januari 23, 2018
dalam "C++"



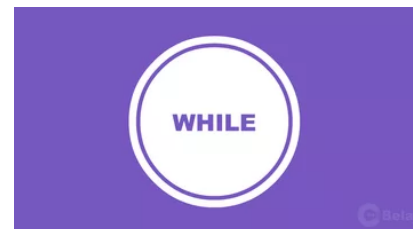
Pernyataan dan Kontrol Aliran
Januari 23, 2018
dalam "C++"



Pernyataan Switch
Januari 23, 2018
dalam "C++"



Pernyataan Do-while
Januari 23, 2018
dalam "C++"



Pernyataan While
Januari 23, 2018
dalam "C++"