

The logo for Oracle Academy is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by a thin black border, with dark gray horizontal bars at the top and bottom.

# ORACLE

## Academy

# Yayasan Jawa

5-2

## Memahami Eksekusi Bersyarat

**ORACLE**  
Academy



Hak Cipta © 2022, Oracle dan/atau afiliasinya. Oracle, Java, dan MySQL adalah merek dagang terdaftar dari Oracle dan/atau afiliasinya. Nama lain mungkin merupakan merek dagang dari pemiliknya masing-masing.

# Tujuan

- Pelajaran ini mencakup tujuan-tujuan berikut:
  - Jelaskan eksekusi bersyarat
  - Jelaskan operator logika
  - Memahami evaluasi "korsleting" operator logika
  - Build dirantai jika konstruksi



## Ketika Beberapa Ketentuan Berlaku

- Bagaimana jika tindakan tertentu diambil hanya jika beberapa kondisi benar?
- Pertimbangkan skenario di mana seorang siswa memenuhi syarat untuk mendapatkan beasiswa jika dua syarat berikut terpenuhi:
  - Grade harus  $\geq 88$
  - Jumlah hari absen = 0

## Menangani Beberapa Kondisi

- Operator relasional baik-baik saja ketika Anda hanya memeriksa satu kondisi
- Anda dapat menggunakan urutan pernyataan if untuk menguji lebih banyak dari satu syarat

```
if (nilai >= 88) {  
    if (jumlahHariAbsen == 0) {  
        System.out.println("Anda memenuhi syarat untuk beasiswa.");  
    }  
}  
//berakhir jika
```

# Menangani Berbagai Kondisi: Contoh

- Seperti yang ditunjukkan dalam contoh:
  - Urutan pernyataan if sulit untuk ditulis, lebih sulit dibaca, dan menjadi lebih sulit saat Anda menambahkan lebih banyak kondisi
  - Untungnya, Java memiliki cara mudah untuk menangani berbagai kondisi: operator logika

# Operator Logika Java

- Anda dapat menggunakan tiga operator logika Java untuk menggabungkan beberapa ekspresi boolean menjadi satu ekspresi boolean

Operator Logika	Arti
<b>&amp;&amp;</b>	DAN
<b>  </b>	ATAU
<b>!</b>	BUKAN

## Tiga Operator Logika

Operasi	Operator	Contoh
Jika satu kondisi DAN kondisi lain	<b>&amp;&amp;</b>	<code>int i = 2; int j = 8; ((i &lt; 1) &amp;&amp; (j &gt; 6))</code>
Jika salah satu syarat ATAU kedua kondisi	<b>  </b>	<code>int i = 2; int j = 8; ((i &lt; 1)    (j &gt; 10))</code>
BUKAN	<b>!</b>	<code>int i = 2; (! (aku &lt; 3))</code>

Tabel di slide mencantumkan operator logika dalam bahasa pemrograman Java. Semua contoh menghasilkan hasil boolean false.



## Menerapkan Operator Logika

- Anda dapat menulis contoh sebelumnya dengan menggunakan logika DAN operator sebagai:

```
nilai >= 88 && angka Hari Absen == 0
```

**Boolean**  
**Ekspresi 1**

**Logis**  
**Operator**

**Boolean**  
**Ekspresi 2**

- Operator logika memungkinkan Anda menguji berbagai kondisi dengan lebih mudah, dan kode lebih mudah dibaca

**ORACLE**  
Academy

JFo 5 -2  
Memahami Eksekusi Bersyarat

Hak Cipta © 2022, Oracle dan/atau afiliasinya. Oracle, Java, dan MySQL adalah merek dagang terdaftar dari Oracle dan/atau afiliasinya. Nama lain mungkin merupakan merek dagang dari pemiliknya masing-masing.

9

Dalam contoh ini, Anda menggunakan operator logika AND karena kedua ekspresi boolean harus benar agar siswa memenuhi syarat untuk mendapatkan beasiswa.

Operator Logical AND:

- Kondisi gabungan benar jika dan hanya jika kedua ekspresi boolean benar.
- Kondisi gabungan salah jika salah satu atau kedua ekspresi boolean salah.

# Logis DAN Operator: Contoh

```
public static void main(String[] args) { int
    numberDaysAbsent = 0; int nilai = 95; ekspresi
benar if (grade >= 88 && numberDaysAbsent == 0) { System.out.println("Anda
    memenuhi syarat untuk beasiswa.");

} else
    { System.out.println("Anda tidak memenuhi syarat untuk beasiswa
      " + ".");

} //endif //
akhir metode utama
```

Mengevaluasi menjadi benar jika keduanya boolean

Contoh ini mengilustrasikan operator logika AND. Agar output ditampilkan sebagai "Anda memenuhi syarat untuk mendapatkan beasiswa", kedua syarat tersebut harus benar. Artinya, nilainya harus lebih besar atau sama dengan 88, dan jumlah hari absen harus sama dengan nol.

# Logis OR Operator

- Pertimbangkan skenario di mana seorang siswa memenuhi syarat untuk tim olahraga jika salah satu dari dua syarat berikut terpenuhi:

•Tingkat  $\geq 70$

•Jumlah hari absen  $< 5$

- Dalam hal ini, Anda dapat menggunakan operator logika OR untuk menghubungkan beberapa ekspresi boolean

```
nilai >=70 || angkaHariAbsen < 5
```

Boolean

Ekspresi 1

Logis

Operator

Boolean

Ekspresi 2

Kondisi gabungan benar jika salah satu atau kedua ekspresi boolean benar.

Kondisi gabungan salah jika kedua ekspresi boolean salah.

# Logis ATAU Operator: Contoh

```
public static void main(String[] args) { int
    numberDaysAbsent = 3; int nilai = 85; if (grade >=
    70 || numberDaysAbsent < 5)
    { System.out.println("Anda memenuhi syarat untuk tim
    olahraga");

    } else
        { System.out.println("Anda tidak memenuhi syarat untuk tim " "
        + olahraga");

    } //endif } //
akhir metode utama
```

Bernilai true jika salah satu dari ekspresi boolean bernilai true

Contoh ini mengilustrasikan menggunakan operator OR logis. Dalam contoh ini, "Anda memenuhi syarat untuk tim olahraga" ditampilkan meskipun salah satu ketentuannya benar. Artinya, nilai harus  $\geq 70$  atau jumlah hari absen harus kurang dari lima.

## Operator NOT Logis

- Pertimbangkan skenario di mana seorang siswa memenuhi syarat untuk les gratis jika dua syarat berikut terpenuhi:
  - Tingkat < 88
  - Jumlah hari absen < 3
- Gunakan ! operator logis

```
!madeFreeTutor && numberDaysAbsent < 3
```

Logis  
Operator

Boolean  
Ekspresi 1

Boolean  
Ekspresi 2

# Operator NOT Logis

```
public static void main(String args[]) {  
    int jumlahHariAbsen = 2;  
    int nilai = 65;  
    boolean madeFreeTutor = nilai >= 88;  
    if (!madeFreeTutor && numberDaysAbsent < 3) {  
        System.out.println("Anda memenuhi syarat untuk bimbingan gratis "  
            + " bantuan");  
    }  
}  
akhir metode utama
```

Contoh ini mengilustrasikan logika ! operator. Karena nilainya sama dengan 65, !madeFreeTutor benar karena madeFreeTutor salah.

Ekspresi gabungan dievaluasi menjadi true dan keluaran berikut ditampilkan: "Anda memenuhi syarat untuk mendapatkan bantuan bimbingan belajar gratis."

## Latihan 1

- Buat proyek baru dan tambahkan file WatchMovie.java ke proyek
- Ubah

WatchMovie.java untuk menonton film yang memenuhi dua ketentuan berikut:

- Harga film lebih dari atau sama dengan \$12
  - Peringkat film sama dengan 5
- Menampilkan output sebagai **"Saya tertarik menonton film"**

Else tampilkan output sebagai **"Saya tidak tertarik menonton film"**

## Melewati Ujian DAN Kedua

- The && dan || operator adalah operator hubung singkat • Jika ekspresi ke-1 (di sebelah kiri) salah, tidak perlu mengevaluasi ekspresi ke-2 (di sebelah kanan)

```
b = (x != 0) && ((y / x) > 2);
```

Kiri  
Ekspresi

Benar  
Ekspresi

Evaluasi dilakukan dalam urutan kiri ke kanan dan berhenti segera setelah hasilnya diketahui. Ini berarti ekspresi di sisi kanan tidak akan dievaluasi jika tidak diperlukan.



## Melewati Ujian DAN Kedua

```
b = (x != 0) && ((y / x) > 2);
```

Kiri  
Ekspresi

Benar  
Ekspresi

- Jika x adalah 0 maka  $(x \neq 0)$  salah
- Untuk operator  $\&\&$ , karena tidak masalah apakah  $((y/x) > 2)$  benar atau salah, hasil dari pernyataan ini adalah salah
- Jadi Java tidak melanjutkan evaluasi  $((y/x) > 2)$

## Melewati Tes OR Kedua

- Jika ekspresi ke-1 (di sebelah kiri) benar, tidak perlu mengevaluasi ekspresi ke-2 (di sebelah kanan)
- Pertimbangkan contoh ini:

```
boolean b = (x <= 10) || (x > 20);
```



- Jika  $(x \leq 10)$  benar, maka  $(x > 20)$  tidak dievaluasi karena tidak masalah jika  $(x > 20)$  benar atau salah
- Hasil ungkapan ini adalah benar

# Apa itu Operator Bersyarat Terner?

Operasi	Contoh Operator
<p>Jika kondisi benar, tetapkan hasil = nilai1</p> <p>Jika tidak, tetapkan hasil = nilai2</p> <p><b>Catatan:</b> nilai1 dan nilai2 harus sama datanya</p> <p>Tipe</p>	<p><b>?:</b> hasil=kondisi ? nilai1 : nilai2</p> <p>Contoh: <code>int</code></p> <p><code>x = 2, y = 5, z = 0;</code></p> <p><code>z = (y &lt; x) ? x : y;</code></p>

## Pernyataan setara

```
z = (y < x) ? x : y;
```

```
jika(y<x)
```

```
{ z=x;
```

```
}
```

```
lain{ z=y; }//berakhir jika
```

**ORACLE**  
Academy

JFo 5 -2  
Memahami Eksekusi Bersyarat

Hak Cipta © 2022, Oracle dan/atau afiliasinya. Oracle, Java, dan MySQL adalah merek dagang terdaftar dari Oracle dan/atau afiliasinya. Nama lain mungkin merupakan merek dagang dari pemiliknya masing-masing.

19

Operator ternary adalah operator kondisional yang membutuhkan tiga operan. Ini memiliki sintaks yang lebih ringkas daripada pernyataan if/else.

Gunakan operator ternary alih-alih pernyataan if/else jika Anda ingin membuat kode Anda lebih pendek.

Ada tiga operan dalam contoh slide: `(y < x)`: Ekspresi

(kondisi) boolean ini sedang dievaluasi. `? x` : Jika `(y < x)` benar, `z` akan diberi nilai `x`. `: y` : Jika `(y < x)` salah, `z` akan diberi nilai `y`.

Dalam contoh slide, `z = 5`.

## Operator Bersyarat Terner: Skenario

- Asumsikan bahwa Anda sedang memainkan pertandingan sepak bola dan melacak sasaran sebagai berikut:

```
public static void main(String args[]) {
    int JumlahTujuan = 5;
    String s; if
    (jumlahTujuan == 1) { s = "tujuan";

    } else { s
        = "tujuan"; }//endif
    System.out.println("Saya
mendapat nilai " + jumlahTujuan + " + s);
} //end method main
```

Berdasarkan jumlah gol yang dicetak, contoh-contoh ini akan mencetak bentuk “gol” tunggal atau jamak yang sesuai. Operasi ini kompak karena hanya dapat menghasilkan dua hasil berdasarkan ekspresi boolean.

## Operator Kondisional Terner: Contoh

- Hasil serupa dicapai dengan operator ternary dengan mengganti seluruh pernyataan if/else dengan satu baris

```
int JumlahTujuan = 1;
```

```
System.out.println("Saya mendapat skor " +  
                    numberOfGoals + + (numberOfGoals == 1 ? "goal" : "goals")  
);
```

## Operator Kondisional Terner: Contoh

- Keuntungan: Tempatkan operasi secara langsung dalam sebuah ekspresi

```
int JumlahTujuan = 1;
```

```
System.out.println("Saya mendapat skor " + numberOfGoals + +  
                    (numberOfGoals == 1 ? "goal" : "goals")  
);
```

- Kerugian: Hanya dapat memiliki dua hasil potensial

```
(numberOfGoals==1 ? "goal" : "goals" : "Lebih banyak gol");
```

boolean

BENAR

Salah

???

ORACLE  
Academy

JFo 5 -2  
Memahami Eksekusi Bersyarat

Hak Cipta © 2022, Oracle dan/atau afiliasinya. Oracle, Java, dan MySQL adalah merek dagang terdaftar dari Oracle dan/atau afiliasinya. Nama lain mungkin merupakan merek dagang dari pemiliknya masing-masing.

22

Seperti yang Anda lihat, operator ternary dapat berguna untuk mengurangi jumlah baris kode, tetapi dapat membuat kode Anda sulit dibaca sehingga tidak ideal untuk pernyataan bersarang.

## Latihan 2

- Tambahkan file TernaryOperator.java ke proyek yang Anda buat untuk latihan 1
- Modifikasi TernaryOperator.java untuk menduplikasi logika yang diberikan dalam pernyataan if/else dengan menggunakan operator ternary

## Menangani Kondisi Kompleks dengan Chained if Membangun

- Pernyataan if yang dirantai:
  - Menghubungkan beberapa kondisi menjadi satu konstruksi
  - Cenderung membingungkan untuk dibaca dan sulit dipertahankan



# Chaining konstruksi if/else

- Anda dapat menggabungkan konstruksi if dan else untuk menyatakan beberapa hasil untuk beberapa ekspresi yang berbeda
- Sintaks:

```
jika (<kondisi1>) { //
    kode_blok1

} else if (<kondisi2>) { //
    kode_block2

} else
    { // kode_default }//
endif
```

Sintaks untuk konstruk if/else yang dirantai ditampilkan dalam contoh slide:

Setiap kondisi adalah ekspresi boolean.

code\_block1 mewakili baris kode yang dijalankan jika kondisi1 benar. code\_block2

mewakili baris kode yang dieksekusi jika kondisi1 salah dan kondisi2 benar. default\_code

mewakili baris kode yang dieksekusi jika kedua kondisi bernilai false.

Catatan: Beberapa pernyataan lain jika dapat dievaluasi. Pernyataan lain adalah opsional.

## Chaining if/else Konstruksi: Contoh

```
public static void main(String args[]) { penghasilan
    ganda = 30000, pajak;

    if (penghasilan <= 15000)
        { pajak = 0;

    } else if (penghasilan <= 25000) {
        pajak = 0,05 * (penghasilan - 15.000);

    } else
        { pajak = 0,05 * (penghasilan - (25.000 - 15.000)); pajak
          += 0,10 * (penghasilan - 25000); }//endif }//akhir metode
    utama
```

Contoh ini menunjukkan konstruksi if/else berantai untuk menguji beberapa kondisi. Pernyataan else dijalankan jika semua kondisi salah.

## Bisakah jika Pernyataan Bersarang?

- Dalam Java, pernyataan if dapat berada di dalam tubuh pernyataan if lainnya

```
if (tvType == "warna") { if
    (ukuran == 14) { discPercent
        = 8;

    } else
        { discPercent = 10; }//
    endif }//endif
```

- Dalam contoh ini, pernyataan else dipasangkan dengan pernyataan if (size==14)

Dalam pernyataan if

bersarang: Sangat penting untuk memastikan konstruksi mana lagi yang sesuai dengan konstruksi if mana. Lakukan ini sangat membantu kejelasan kode bagi pembaca.

Dalam contoh ini, jika pernyataan if bagian luar benar, maka pernyataan if bagian dalam dijalankan.

## Memahami Pernyataan if bersarang

- Dalam contoh ini, pernyataan else dipasangkan dengan pernyataan if luar (TVType=="color")

```
if (tvType == "warna") { if
    (ukuran == 14) { discPercent
        = 8; } //berakhir jika

} else
    { discPercent = 10; } //
berakhir jika
```

## Latihan 3

- Tambahkan file ComputeFare.java ke proyek yang Anda buat untuk latihan 1
- Periksa ComputeFare.java
- Implementasikan berikut ini menggunakan konstruksi if/else:
  - Deklarasikan variabel bilangan bulat, usia
  - Minta pengguna memasukkan nilai untuk usia
  - Menggunakan konstruksi if berantai, hitung tarif berdasarkan usia sesuai dengan ketentuan berikut:
    - Jika usia kurang dari 11, maka tarif = \$3
    - Jika usia lebih dari 11 dan kurang dari 65, maka tarif = \$5
    - Selain itu untuk semua usia lainnya, maka tarif = \$3

# Ringkasan

- Dalam pelajaran ini, Anda seharusnya telah mempelajari cara untuk:
  - Jelaskan eksekusi bersyarat
  - Jelaskan operator logika
  - Memahami evaluasi "korsleting" operator logika
  - Build dirantai jika konstruksi



**ORACLE**  
Academy

JFo 5 -2  
Memahami Eksekusi Bersyarat

Hak Cipta © 2022, Oracle dan/atau afiliasinya. Oracle, Java, dan MySQL adalah merek dagang terdaftar dari Oracle dan/atau afiliasinya. Nama lain mungkin merupakan merek dagang dari pemiliknya masing-masing.

30

The Oracle Academy logo is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by a thin black border, with dark gray horizontal bars at the top and bottom.

**ORACLE**  
Academy