

Tujuan

• Pelajaran ini mencakup tujuan-tujuan berikut:

ÿJelaskan eksekusi bersyarat

ÿJelaskan operator logika

ÿMemahami evaluasi "korsleting" operator logika

ÿBuild dirantai jika konstruksi





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.

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:

```
ÿGrade harus >= 88
```

ÿJumlah hari absen = 0



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 masind-masind.

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
}//berakhir jika
```



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.

Menangani Berbagai Kondisi: Contoh

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



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.

Operator Logika Java

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

Operator Logika	Arti
&&	DAN
	ATAU
!	BUKAN



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.

ORACLE Academy

JFo 5 -2

Memahami Eksekusi Bersyarat

Tiga Operator Logika Operasi Operator Contoh && Jika satu kondisi DAN int i = 2; kondisi lain int i = 8; ((i < 1) && (j > 6))Jika salah satu syarat int i = 2; Ш int j = 8; ATAU kedua kondisi ((i < 1) || (j > 10))int i = 2; (! BUKAN (aku < 3))

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

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:

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



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 masino-masino.

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
                                                                                Mengevaluasi menjadi benar jika
        numberDaysAbsent = 0; int nilai = 95; ekspresi
                                                                                keduanya boolean
        benar if (grade >= 88 && numberDaysAbsent ==
                                                                   0) { System.out.println("Anda
        mernenuhi syarat untuk beasiswa.");
        } else
              { System.out.println("Anda tidak memenuhi syarat untuk beasiswa
        }//endif }//
   akhir metode utama
ORACLE
Academy
                                                   Hak Cipta © 2022, Oracle dan/atau afiliasinya. Oracle, Java, dan MySQL adalah merek dagang
                      JFo 5 -2
                                                   terdaftar dari Oracle dan/atau afiliasinya. Nama lain mungkin merupakan merek dagang dari
                                                                                                             10
                      Memahami Eksekusi Bersyarat
```

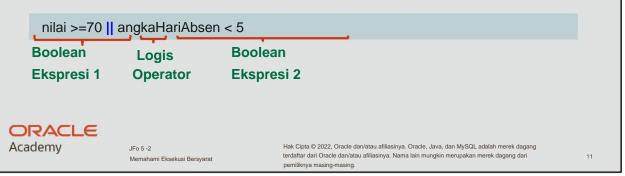
Contoh ini mengilustrasikan operator logika AND. Agar output ditampilkan sebagai "Anda memenuhi syarat untuk mendapatkan beasiswa", kedua syarat tersebut harus benar. Artinya, nilainya harus lebih besar atau sama dengan 88, dan jumlah hari absen harus sama dengan nol.

Logis OR Operator

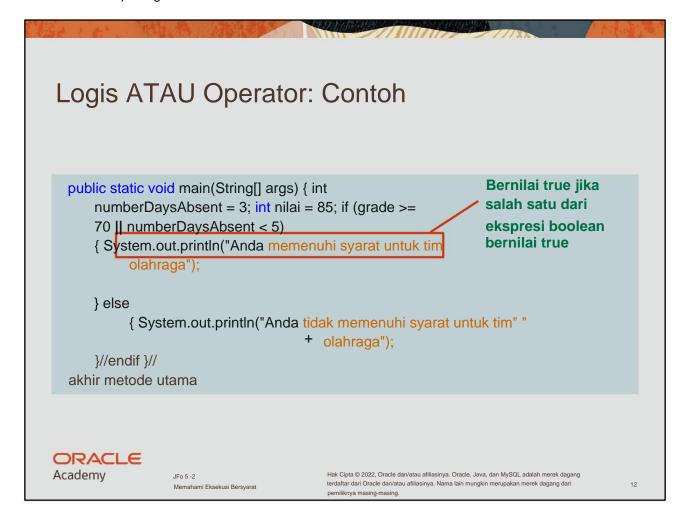
 Pertimbangkan skenario di mana seorang siswa memenuhi syarat untuk tim olahraga jika salah satu dari dua syarat berikut terpenuhi:

```
ÿTingkat >= 70
ÿJumlah hari absen < 5
```

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



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



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 >= 70 atau jumlah hari absen harus kurang dari lima.

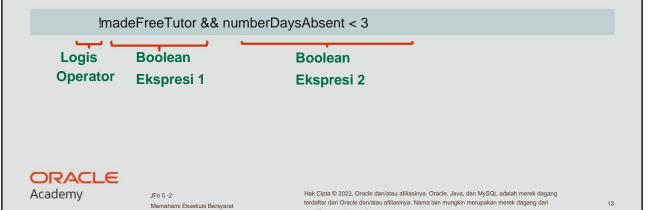
Operator NOT Logis

 Pertimbangkan skenario di mana seorang siswa memenuhi syarat untuk les gratis jika dua syarat berikut terpenuhi:

ÿTingkat < 88

ÿJumlah hari absen < 3

•Gunakan! operator logis



```
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");
       }//endif }//
  akhir metode utama
ORACLE
Academy
                                                   Hak Cipta © 2022, Oracle dan/atau afiliasinya. Oracle, Java, dan MySQL adalah merek dagang
                      JFo 5 -2
                                                   terdaftar dari Oracle dan/atau afiliasinya. Nama lain mungkin merupakan merek dagang dari
                      Memahami Eksekusi Bersyarat
```

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

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

Latihan 1

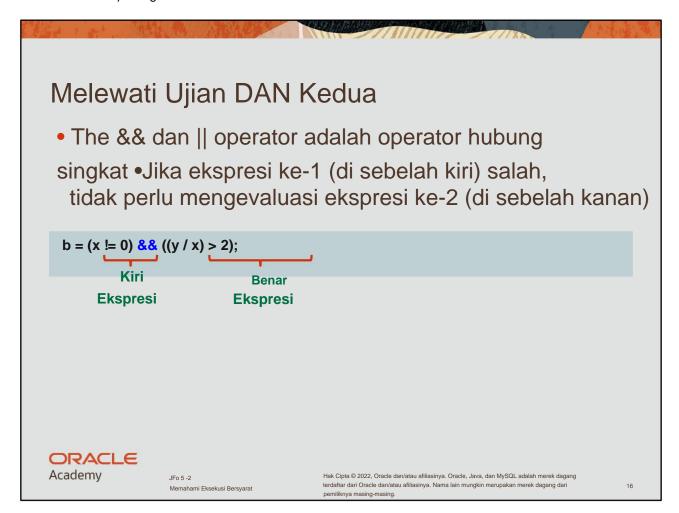
 Buat proyek baru dan tambahkan file WatchMovie.java ke proyek •Ubah

WatchMovie.java untuk menonton film yang memenuhi dua ketentuan berikut: • Harga film lebih dari atau sama dengan \$12 • Peringkat film sama dengan 5 ÿMenampilkan output sebagai "Saya tertarik menonton film"

ÿElse tampilkan output sebagai "Saya tidak tertarik menonton film"



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 masind-masing.



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 Benar

Ekspresi Ekspresi
```

•Jika x adalah 0 maka (x !=

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)



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.

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);

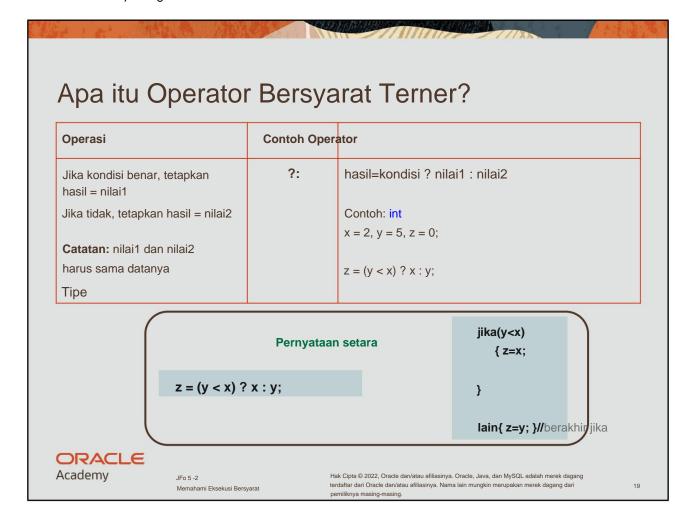
Kiri Benar

Ekspresi Ekspresi
```

- •Jika (x<=10) benar, maka (x>20) tidak dievaluasi karena tidak masalah jika (x>20) benar atau salah
- Hasil ungkapan ini adalah benar



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.



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

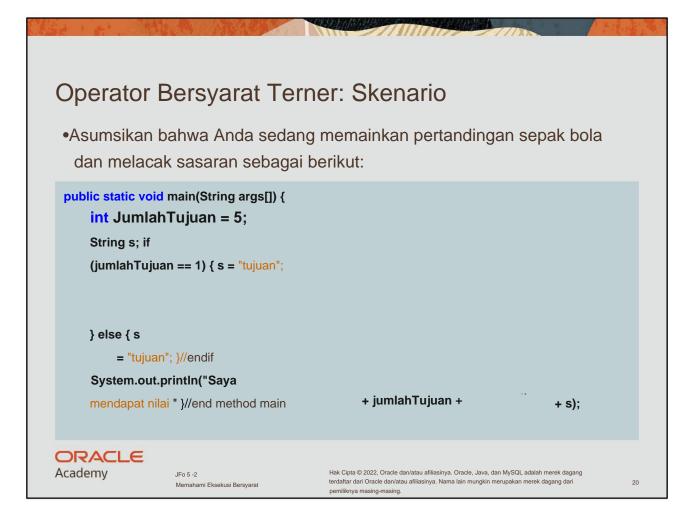
Gunakan operator ternary alih-alih pernyataan if/else jika Anda ingin membuat kode Anda lebih pendek.

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.



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

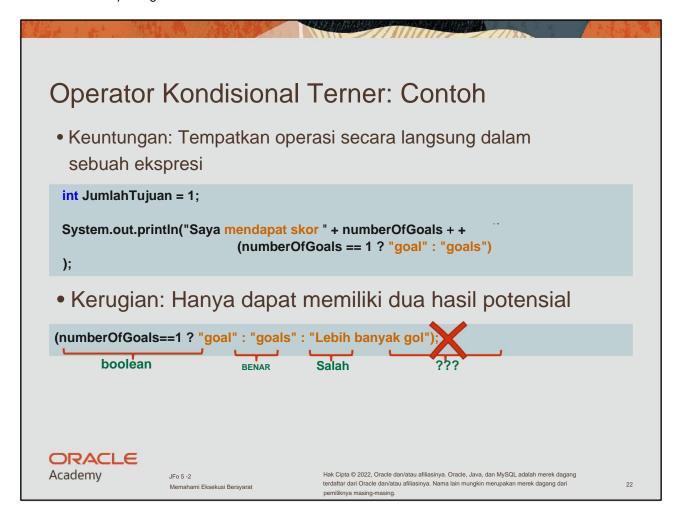
 Hasil serupa dicapai dengan operator ternary dengan mengganti seluruh pernyataan if/else dengan satu baris



JFo 5 -2

Memahami Eksekusi Bersyarat terdaftar dari Orac

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.



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



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.

Menangani Kondisi Kompleks dengan Chained if Membangun

• Pernyataan if yang dirantai:

ÿMenghubungkan beberapa kondisi menjadi satu konstruksi ÿCenderung membingungkan untuk dibaca dan sulit dipertahankan



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.

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>) { //
    code_block2

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



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.

25

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</pre>
```

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.

26

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

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



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 masino-masino.

27

Dalam pernyataan if

bersarang: Sangat penting untuk memastikan konstruksi mana lagi yang sesuai dengan konstruksi if mana. Lekukan 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")



JFo 5 -2 Memahami Eksekusi Bersvarat 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.

Latihan 3

- •Tambahkan file ComputeFare.java ke proyek yang Anda buat untuk latihan 1
- Periksa ComputeFare.java
- •Implementasikan berikut ini menggunakan konstruksi if/else:
 - ÿ Deklarasikan variabel bilangan bulat, usia ÿ Minta pengguna memasukkan nilai untuk usia •Menggunakan

konstruksi if berantai, hitung tarif berdasarkan usia sesuai dengan ketentuan berikut: ÿJika usia kurang dari 11, maka tarif = \$3 ÿJika usia lebih dari 11 dan kurang dari 65, maka tarif = \$5 ÿSelain itu untuk semua usia lainnya, maka tarif = \$3



JFo 5 -2 Memahami Eksekusi Bersvarat 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 masind-masing.

Ringkasan

•Dalam pelajaran ini, Anda seharusnya telah mempelajari cara untuk:

ÿJelaskan eksekusi bersyarat

ÿJelaskan operator logika

ÿMemahami evaluasi "korsleting" operator logika

ÿBuild 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.

