

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

5-1

Ekspresi Boolean dan konstruksi if/else

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

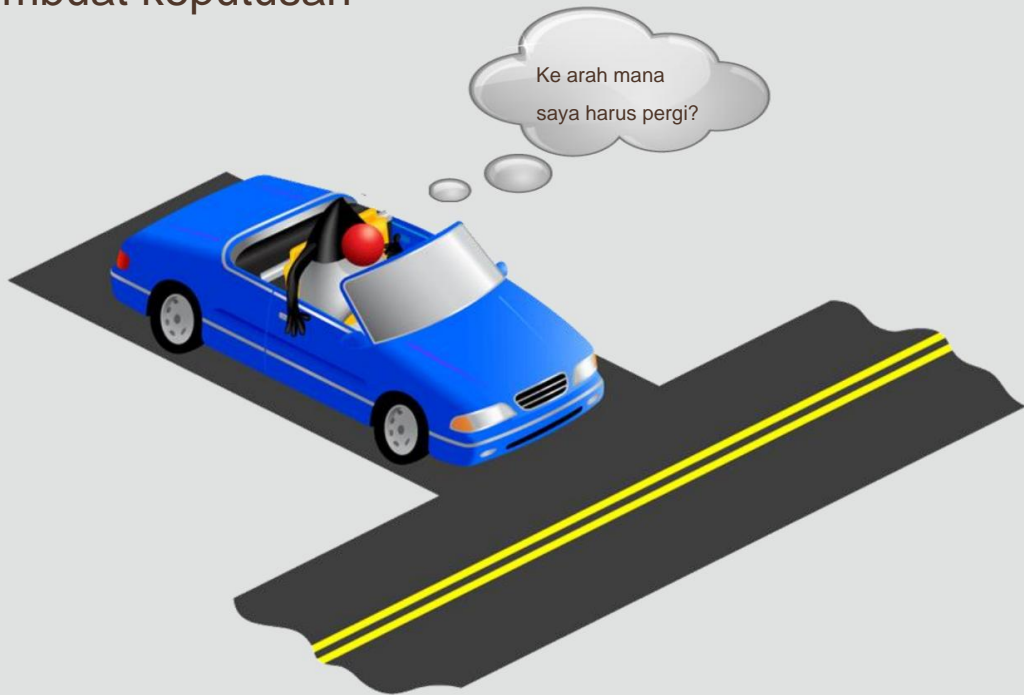
- Pelajaran ini mencakup tujuan-tujuan berikut:
 - Deklarasikan, inisialisasi, dan gunakan variabel boolean
 - Bandingkan ekspresi boolean menggunakan operator relasional
 - Membuat pernyataan if
 - Buat konstruksi if/else
 - Bandingkan String



Membuat keputusan

- Sejauh ini di pelajaran sebelumnya, Anda melihat berbagai tipe data yang didukung di Java
- boolean adalah tipe data lain di Java yang membantu menambahkan logika ke program
- Membantu membuat keputusan

Membuat keputusan



ORACLE
Academy

JFo 5-1
Ekspresi Boolean dan konstruksi if/else

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.

5

Membuat keputusan

- Katakanlah Anda sedang mengemudi ke sekolah •

Anda berhenti di persimpangan • Dan sekarang Anda

harus membuat keputusan yang logis: ¿Jika saya belok kiri,
apakah saya akan sampai ke sekolah?

¿Jika saya lurus, apakah itu akan membawa saya ke sekolah?

¿Jika saya belok kanan, apakah itu akan membawa saya ke sekolah?

- Hanya ada dua jawaban untuk setiap pertanyaan ini:

¿ya atau tidak

Tipe Data boolean Java

- Ini pada dasarnya sama di Java, di mana boolean akan memberi tahu program mana tindakan terbaik untuk diambil
- Di Java nilai untuk tipe data boolean adalah benar dan salah, bukan ya dan tidak
- Anda mendeklarasikan tipe data boolean dengan menggunakan kata kunci boolean

Menggunakan Tipe Data boolean Java: Contoh

- Catatan: Nilai variabel boolean ditampilkan sebagai benar atau salah

```
public static void main(String args[]) {
```

```
    boolean lulus, tempat besar, kelas;
```

Mendeklarasikan
variabel boolean

```
    lulus = benar;
```

```
    tempatbesar = salah;
```

```
    nilai = lulus;
```

Menetapkan nilai ke
variabel boolean

```
    System.out.println(lulus);
```

```
    System.out.println(Tempatbesar);
```

```
    System.out.println(grade); }// akhir
```

Mencetak nilai variabel
boolean

metode utama

Ingatlah bahwa tipe data boolean hanya dapat memiliki dua kemungkinan nilai: benar dan salah.

Catatan: Benar dan salah tidak pernah disertakan dalam tanda kutip karena merupakan nilai boolean, bukan String.

Output dari contoh yang ditampilkan di slide: benar
salah

BENAR

Tipe Data boolean: Skenario

- Bagaimana jika Anda mengendarai mobil yang memiliki sistem GPS terinstal yang berjalan di Jawa?
- Sebelum Anda meninggalkan rumah, Anda meminta sistem GPS untuk membawa Anda ke sekolah
- Kode sederhana apa yang akan Anda tulis untuk membantu Anda memutuskan ke arah mana?

Tipe Data boolean: Skenario

- Mari kita mulai

```
public static void main(String args[]) {  
    String kiri = "museum";  
    String lurus = "gym";  
    String kanan = "restoran";  
    boolean kiri = salah;  
    boolean isStraight = true;  
    boolean isRight = false;  
    System.out.println(" Langsung saja");  
} // akhir metode utama
```

Berdasarkan kode sederhana ini, mobil akan menuju ke arah yang memiliki variabel boolean dengan nilai BENAR.

Catatan: Menyempurnakan contoh ini dengan lebih banyak kode akan dibahas lebih lanjut dalam pelajaran ini.

Ekspresi dan Variabel

- Ekspresi matematika dapat berupa...

- Dicitak

- Ditugaskan ke variabel int atau ganda

```
System.out.println(2 + 2);
```

```
int x = 2 + 2;
```

Ekspresi dan Variabel

- ekspresi boolean dapat berupa ...
 - Dicitak
 - Ditugaskan ke variabel boolean

```
System.out.println(x == 5);
```

```
boolean adalahLima = x == 5;
```

Gunakan tanda sama dengan (=) untuk membuat tugas dan gunakan tanda == untuk membuat perbandingan dan mengembalikan boolean.

Kesetaraan dan Penugasan

- `==` adalah operator relasional
- Operator ini menguji untuk melihat apakah kedua sisi ekspresi boolean sama atau berbeda • Ekspresi boolean mengembalikan nilai benar atau salah

```
x == 5
```

Kesetaraan dan Penugasan

- = adalah operator penugasan
- Operator ini memberikan nilai ke variabel
- Variabel boolean dapat diberikan nilai mana saja yang dikembalikan oleh ekspresi boolean

```
int x = 4;
```

```
boolean adalahLima = x == 5;
```

Nilai dalam Ekspresi boolean

- Gunakan == untuk menguji kesetaraan antara nilai primitif • ekspresi

boolean mungkin berisi variabel atau hard

nilai-nilai yang dikodekan

```
boolean res1 = 24 == 15;  
System.out.println("res1: " + res1);  
int nilai1 = 15;  
int nilai2 = 24;  
boolean res2 = nilai1 == nilai2;  
System.out.println("res2: " + res2);
```

Nilai dalam Ekspresi boolean

- Kedua ekspresi di bawah mengembalikan nilai yang sama:

• Jika nilai1 dan nilai2 memiliki nilai yang sama, ekspresinya
mengembalikan hasil yang benar

• Jika tidak, ekspresi mengembalikan salah

```
boolean res1 = 24 == 15;  
System.out.println("res1: " + res1);  
int nilai1 = 15;  
int nilai2 = 24;  
boolean res2 = nilai1 == nilai2;  
System.out.println("res2: " + res2);
```


Operator Relasional

- Gunakan operator relasional dalam ekspresi boolean yang digunakan untuk mengevaluasi pernyataan if/else

Operator Relasional

Kondisi	Operator	Contoh
Adalah sama dengan	<code>==</code>	<code>int i=1;</code> (saya == 1)
Tidak sama dengan	<code>!=</code>	<code>int i=2;</code> (saya != 1)
Kurang dari	<code><</code>	<code>int i=0;</code> (saya < 1)
Kurang dari atau sama dengan	<code><=</code>	<code>int i=1;</code> (saya <= 1)
Lebih besar dari	<code>></code>	<code>int i=2;</code> (saya > 1)
Lebih besar atau sama dengan	<code>>=</code>	<code>int i=1;</code> (saya >= 1)

Di sini Anda melihat daftar operator relasional yang lebih lengkap. Tabel mencantumkan kondisi berbeda yang dapat Anda uji dengan menggunakan operator relasional. Hasil dari semua operator relasional adalah nilai boolean. Semua contoh menghasilkan hasil boolean yang benar.

Operator Relasional: Contoh

- Catatan: Gunakan tanda sama dengan (=) untuk membuat tugas dan gunakan tanda == untuk membuat perbandingan dan mengembalikan nilai boolean

```
public static void main(String args[]) {
    int a = 10; int b =
    20;
    System.out.println(a == b);
    System.out.println(a != b); System.out.println(a
    > b); System.out.println(a < b);
    System.out.println(b >= a);
    System.out.println(b <= a); }// akhir metode
    utama
```

Untuk nilai
primitif == periksa

pengujian kesetaraan

Pengujian

Kesetaraan: Untuk objek, gunakan metode yang sama untuk menguji kesetaraan.

Keluaran:

salah

BENAR

Salah

BENAR

BENAR

Salah

Latihan 1

- Buat proyek baru dan tambahkan file AgeValidity.java ke proyek
- Modifikasi

AgeValidity.java untuk menerapkan hal berikut:

- Minta pengguna memasukkan usia mereka
- Deklarasikan variabel boolean, `drivingUnderAge`
- Inisialisasi `drivingUnderAge` ke `false`
- Tulis ekspresi boolean untuk memeriksa jika usia yang dimasukkan oleh pengguna kurang dari atau sama dengan 18, lalu setel `drivingUnderAge` ke `true`

• Cetak nilai `drivingUnderAge`

Pernyataan bersyarat

- Pernyataan bersyarat mari kita pilih pernyataan mana yang akan dieksekusi selanjutnya
- Keputusan ini didasarkan pada ekspresi boolean (atau kondisi) yang mengevaluasi benar atau salah
- Pernyataan bersyarat dalam Java adalah:
 • pernyataan
 • if / pernyataan if /
 • else / pernyataan ganti

Memahami pernyataan if

- Sebuah pernyataan if terdiri dari ekspresi boolean yang diikuti oleh satu atau lebih pernyataan • Sintaks:

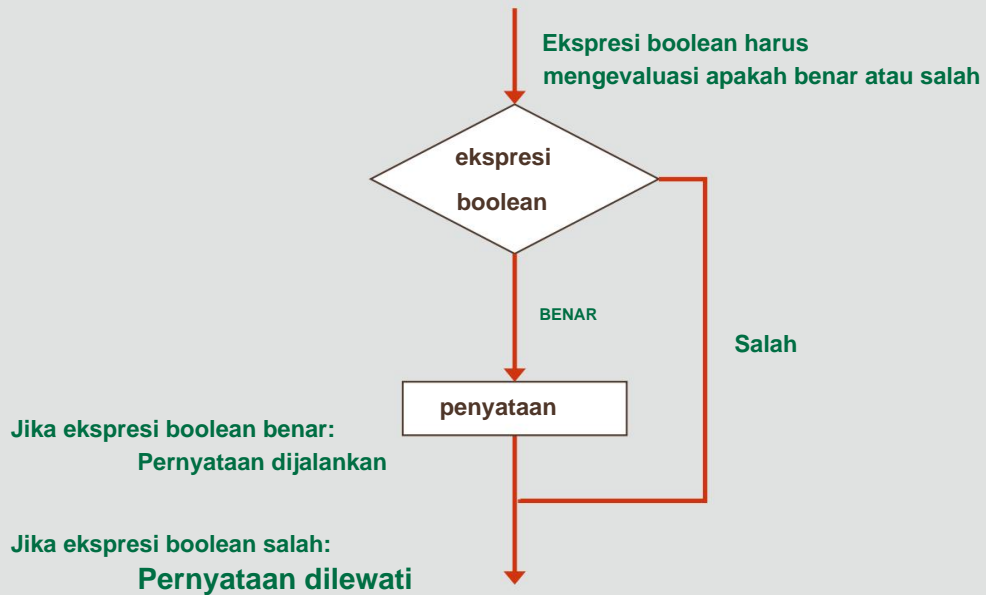
ekspresi boolean

```
if ( <kondisi tertentu benar> ){
```

```
//Pernyataan akan dijalankan jika //ekspresi boolean benar }//endif
```

Kondisi yang akan dievaluasi dikelilingi oleh tanda kurung. Ini disebut sebagai ekspresi boolean karena harus mengevaluasi apakah benar atau salah.

Memahami pernyataan if



Menggunakan Ekspresi boolean dalam Pernyataan if

```
public static void main(String args[]) {  
    String kiri = "museum";  
    String lurus = "gym";  
    String kanan = "restoran";  
  
    if (kiri == "gym")  
        { System.out.println("Belok Kiri"); } //berakhir jika  
  
    if (straight == "gym")  
        { System.out.println("Drive Lurus"); } //berakhir jika  
  
    if (kanan == "gym")  
        { System.out.println("Belok Kanan"); } //endif } //akhir  
metode utama
```

Blok ini
dijalankan

Dalam contoh slide, ekspresi boolean di pernyataan if kedua mengembalikan nilai true. Oleh karena itu, "Drive Straight" dicetak ke konsol.

Menjalankan Blok Kode

1. Blok kode tidak diperlukan untuk membuat satu pernyataan dieksekusi oleh pernyataan if.

Ini contohnya:

```
harilnFeb = 28;  
jika(adalahTahun Kabisat) Hanya pernyataan  
ini daysInFeb = 29; dilakukan  
System.out.println(tahun + " adalah tahun kabisat");
```

1

Dalam contoh 1, Anda ingin kedua pernyataan dieksekusi ketika kondisinya benar. Karena tidak ada blok kode, hanya pernyataan pertama yang dieksekusi.

Menjalankan Blok Kode

2. Namun, selalu disarankan agar Anda menggunakan blok kode, meskipun hanya ada satu pernyataan untuk dieksekusi di blok tersebut

```
hariInFeb = 28;  
if(isLeapYear)  
{ hariInFeb = 29; Blok ini akan menjadi System.out.println(tahun + " adalah tahun  
kabisaat"); dieksekusi } //endif
```

2

Pada contoh 2, kedua pernyataan dijalankan ketika kondisinya benar karena ada blok kode untuk pernyataan if .

Untuk menghindari kesalahan ini, Anda harus menggunakan blok kode meskipun hanya ada satu pernyataan untuk dieksekusi di blok if.

if Pernyataan: Contoh

```
public static void main(String args[]) { int grade = 85;

    if (grade > 88)
        { System.out.println("Anda membuat Daftar Kehormatan."); } //endif

    if (grade <=88)
        { System.out.println("Anda memenuhi syarat untuk bimbingan belajar."); }//
    berakhir jika

} // akhir metode utama
```

Kedua jika
pernyataan

•Keluaran:

Anda memenuhi syarat untuk bimbingan belajar.

ORACLE
Academy

JFo 5-1
Eksresi Boolean dan konstruksi if/else

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.

27

Dalam contoh slide, ada dua pernyataan if: Tes pertama untuk nilai lebih besar dari 88. Tes kedua untuk nilai kurang dari atau sama dengan 88.

Keduanya jika pernyataan dievaluasi, bahkan jika yang pertama benar.

Latihan 2

- Tambahkan file ChkOddEven.java ke proyek yang Anda buat untuk latihan 1
- Modifikasi ChkOddEven.java untuk mengimplementasikan hal berikut:
 - Masukkan angka antara 1 dan 10
 - Gunakan pernyataan if

• Uji apakah suatu bilangan ganjil atau genap

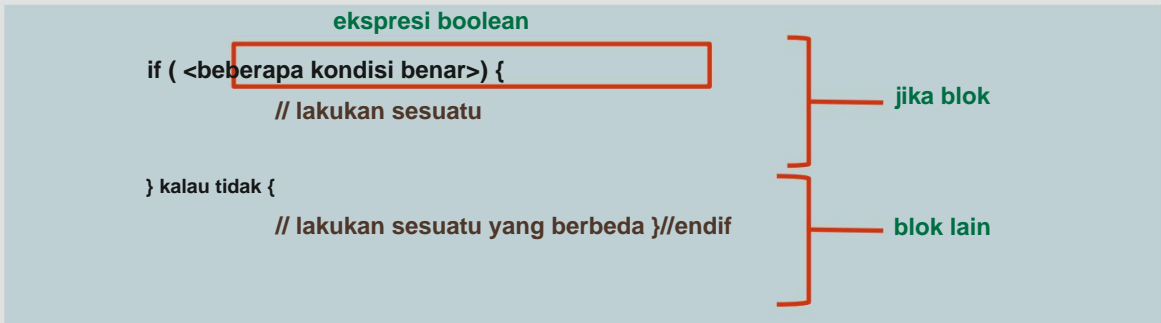
- Program harus menghasilkan keluaran berikut:

• Masukkan angka: 7

• Jumlahnya ganjil 7

Memilih Antara Dua Alternatif

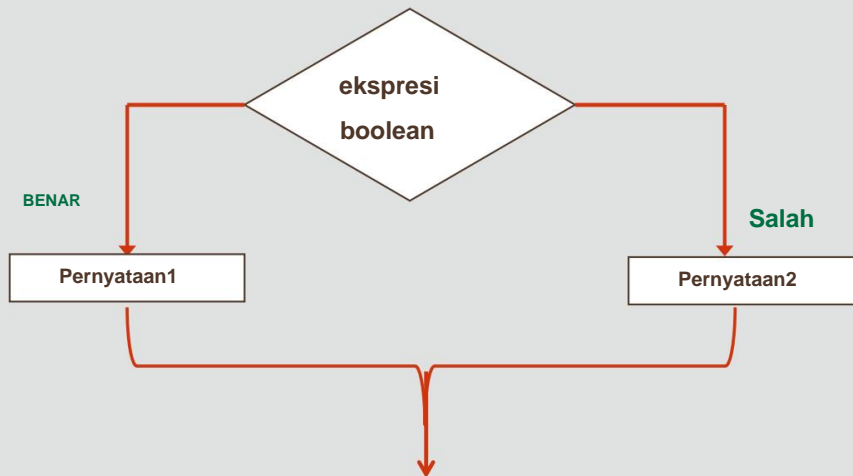
- Jika Anda ingin memilih antara dua alternatif, gunakan pernyataan if/else • Sintaks:



Pernyataan if/else adalah salah satu cara untuk mencabangkan kode Anda tergantung pada beberapa kondisi. Ini menggunakan dua kata kunci Java, jika dan lainnya.

- Jika suatu kondisi benar, jalankan kode di dalam blok if.
- Jika kondisi tersebut salah, jalankan kode di blok else.

Memahami pernyataan if/else



Jika ekspresi boolean benar: Pernyataan1 dijalankan
Jika ekspresi boolean salah: Pernyataan2 dilewati

Pernyataan if/else: Contoh 1

```
Perkiraan string; suhu  
ganda = getTemperature();
```

```
if (suhu <= 32,0) { prakiraan = "SALJU";
```

```
} else
```

```
{ prakiraan = "HUJAN"; }//
```

```
berakhir jika
```

Blok ini
dijalankan



30,3 °F

Slide ini menunjukkan contoh if/else. Metode `getTemperature()` mengembalikan suhu sebagai 30,3. Karena suhunya kurang dari 32 derajat, ekspresi boolean (`suhu <= 32.0`) mengembalikan nilai true, dan blok if dijalankan.

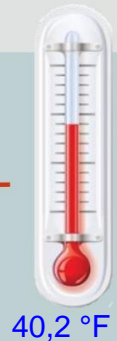
Pernyataan if/else: Contoh 2

```
Perkiraan string;  
suhu ganda = getTemperature();
```

```
if (suhu <= 32,0) { prakiraan =  
    "SALJU";
```

```
} else  
{ prakiraan =  
    "HUJAN"; }//berakhir jika
```

Blok ini
dieksekusi



40,2 °F

Slide ini menunjukkan contoh if/else, metode `getTemperature()` mengembalikan suhu sebagai 40.2. Dalam pernyataan if/else, karena `suhu <= 32` derajat, ekspresi boolean, `suhu <= 32.0` mengembalikan false, dan blok else adalah dieksekusi.

Pernyataan if/else: Contoh 3

- Anda dapat mengganti kedua pernyataan if dengan if/else pernyataan
- Pernyataan if/else lebih efisien karena hanya satu perbandingan yang dibuat

```
public static void main(String args[]) { int grade = 85; if  
    (grade > 88) { System.out.println("Anda membuat  
    Daftar Kehormatan.");  
  
    } else  
        { System.out.println("Anda lulus."); } //endif } //  
akhir metode utama
```

Latihan 3

- Tambahkan file AgeCheck.java ke proyek yang Anda buat untuk latihan 1
- Periksa AgeCheck.java:
 - Program memiliki masalah logika
 - Untuk beberapa nilai, ia mencetak jawaban yang salah
 - Temukan masalah dan perbaiki. (Anda mungkin perlu menjalankan program beberapa kali dan mencoba nilai yang berbeda untuk melihat mana yang gagal)
 - Ganti kedua pernyataan if dengan pernyataan if/else

Latihan 4

- Tambahkan file ShoppingCart.java ke proyek yang Anda buat untuk latihan 1
- Periksa ShoppingCart.java
- Gunakan pernyataan if/else untuk mengimplementasikan hal berikut:
 - Deklarasikan dan inisialisasi variabel boolean, outOfStock
 - Jika jumlah > 1, ubah pesan variabel untuk menunjukkan jamak
 - Jika item kehabisan stok, beri tahu pengguna bahwa item tersebut tidak tersedia, jika tidak, cetak pesan dan total biaya

Membandingkan Variabel

- Saat Anda membandingkan nilai dengan menggunakan ekspresi boolean, Anda perlu memahami nuansa tipe data tertentu
- Operator relasional seperti `==` adalah ...

• Bagus untuk membandingkan primitif

• Sangat buruk untuk membandingkan String (dan objek lain)

- Mari kita periksa alasannya

Membandingkan Primitif

- Nilai z ditetapkan sebagai jumlah dari $x + y$
- Ketika ekspresi boolean menguji persamaan antara z dan jumlah dari $x + y$, hasilnya benar

```
int x = 3; int  
y = 2; int z =  
x + y;  
  
uji boolean = (z == x + y);  
System.out.println(test);           //BENAR
```

Membandingkan String

- Nilai z diatur menjadi gabungan dari $x + y$
- Ketika ekspresi boolean menguji kesetaraan antara z dan gabungan dari $x + y$, hasilnya salah

```
String x = "Ora";  
String y = "cle";  
Tali z = x + y;
```

```
uji boolean = (z == x + y);  
System.out.println(test);
```

//Salah

Why?

Mengapa Ada Hasil yang Bertentangan?

- Primitif dan objek disimpan secara berbeda dalam memori
 - String diberi perlakuan khusus
 - Ini dibahas nanti dalam kursus
- Akibatnya ...
 - == membandingkan nilai primitif
 - == membandingkan lokasi objek di memori
- Kemungkinan besar Anda perlu membandingkan konten String dan bukan lokasinya di memori

Bagaimana Seharusnya Anda Membandingkan String?

- Anda hampir tidak boleh membandingkan String menggunakan `==`
- Alih-alih, bandingkan String menggunakan metode `equals()`
 - Metode ini adalah bagian dari kelas String
 - Menerima satu argumen String, memeriksa apakah isi String sama, lalu mengembalikan boolean
 - Ada juga metode serupa, `equalsIgnoreCase()`

```
String x = "Ora";
String y = "cle";
Tali z = x + y; tes
boolean = z.equals(x + y);
System.out.println(test); //BENAR
```

Bagaimana Anda menulis pernyataan:

`boolean test = z.equals(x + y);`

Sehingga nilai tes akan dievaluasi **benar** ketika huruf besar dan kecil tidak penting?

Jawaban: uji `boolean = z.equalsIgnoreCase(x + y);`

Latihan 5

- Tambahkan file `StringEquality.java` ke proyek yang Anda buat untuk latihan 1
- Periksa `StringEquality.java` • Gunakan pernyataan `if` dan `if/else`:
 - Deklarasikan nama variabel `String`
 - Minta pengguna memasukkan nilai untuk nama tersebut
 - Periksa apakah namanya "Moe", lalu cetak "You are the king rock and roll"
 - Jika tidak, cetak "Kamu bukan raja"
 - Jangan gunakan `==`

Ringkasan

- Dalam pelajaran ini, Anda seharusnya telah mempelajari cara untuk:
 - Deklarasikan, inisialisasi, dan gunakan variabel boolean
 - Bandingkan nilai primitif menggunakan operator relasional
 - Membuat pernyataan if
 - Buat konstruksi if/else
 - Bandingkan String



ORACLE
Academy

JFo 5-1
Ekspresi Boolean dan konstruksi if/else

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.

42

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 two horizontal dark gray bars, one at the top and one at the bottom.

ORACLE
Academy