

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

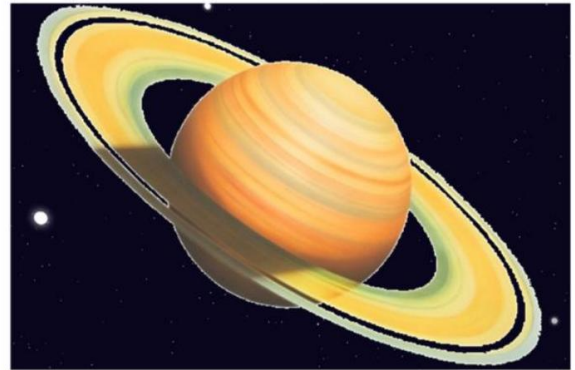
Academy

Yayasan Jawa

6-1

untuk Loop

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:

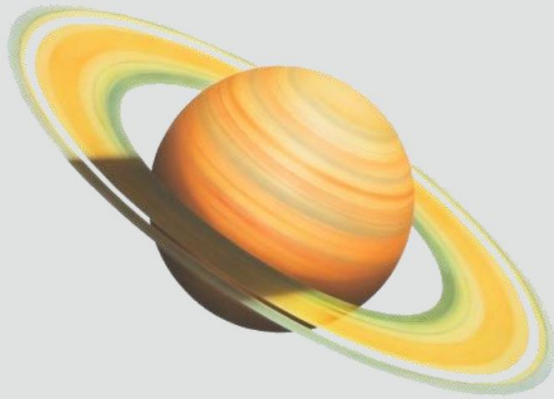
- Pelajaran ini mencakup tujuan-tujuan berikut:

- Memahami komponen standar untuk loop
- Memahami cara membuat dan menggunakan perulangan for
- Memahami ruang lingkup variabel
- Memahami teknik debug
- Menjelaskan bagaimana loop tak terbatas terjadi di Jawa



Misi ke Cincin Saturnus

- Kita akan meluncurkan kapal roket •Misinya adalah mempelajari cincin Saturnus •Apakah Anda memiliki pemikiran tentang cara memprogram penghitung waktu mundur?

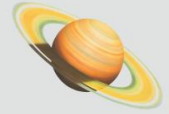


ORACLE
Academy

JFo 6-1
untuk Loop

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.

4



Hitung mundur

- Menghitung mundur dari 10 membutuhkan 10 baris kode

```
System.out.println("Hitung mundur untuk Meluncurkan: ");  
System.out.println(10); System.out.println(9);  
System.out.println(8); System.out.println(7);  
System.out.println(6); System.out.println(5);  
System.out.println(4); System.out.println(3);  
System.out.println(2); System.out.println(1);  
System.out.println("Ledakan !");
```



Hitung mundur

- Menghitung mundur dari 100 membutuhkan 100 baris kode
- Itu akan menyakitkan dan membosankan untuk diprogram.
- Apakah ada cara yang lebih praktis untuk menulis program ini?
- Bisakah kode mengakomodasi nilai awal dengan mudah?



Hitung mundur

```
System.out.println("Hitung mundur untuk  
Meluncurkan: "); System.out.println(100);  
System.out.println(99); System.out.println(98);  
System.out.println(97); System.out.println(96);  
System.out.println(95);
```

```
...  
...  
...  
...  
...  
...  
...
```

```
System.out.println(2);  
System.out.println(1);  
System.out.println("Ledakan !");
```

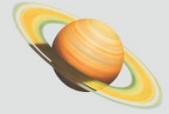


ORACLE
Academy

JFo 6-1
untuk Loop

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.

7



Dapatkan Variabel Membantu?

- Variabel agak membantu • Tapi kita masih harus menyalin dan menempelkan baris kode yang sama sampai 0 cetakan

```
System.out.println("Hitung mundur untuk Meluncurkan: ");
```

```
int i = 10;
```

```
System.out.println(i); saya--;
```

```
System.out.println(i); saya--;
```

```
System.out.println(i); saya--;
```

```
...
```

```
System.out.println("Ledakan !");
```

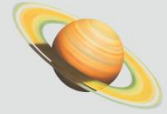


ORACLE
Academy

JFo 6-1
untuk Loop

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.

8



Kode Berulang

- Bisakah kita membuat baris kode yang sama berulang beberapa kali?
- Baris 7–10 menunjukkan blok kode yang ingin kita ulangi
- Ingat sifat program baris demi baris: Ketika program mencapai baris 10 ...
Kami ingin mengulang kembali ke baris 7

```
5 int i = 10; 6 7
```

```
{  
    System.out.println(i);  
    saya--;  
8 9 10}
```



ORACLE
Academy

JFo 6-1
untuk Loop

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

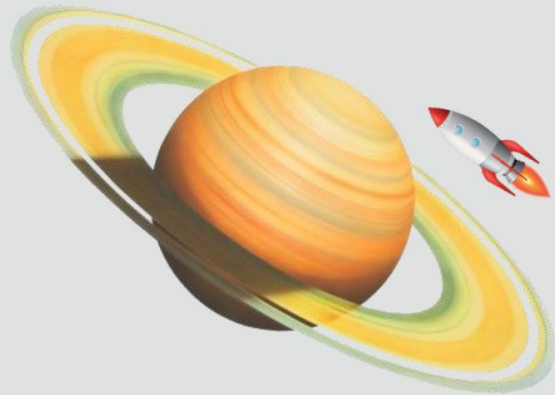
Pernyataan Lingkaran

- Pernyataan loop digunakan untuk mengulang baris kode. •Java menyediakan tiga jenis loop:

• untuk

• sementara

• lakukan-sementara



ORACLE
Academy

JFo 6-1
untuk Loop

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.

10

Perilaku Berulang



```
while (!areWeThereYet) {  
  
    membaca buku;  
    berdebat dengan saudara;  
    bertanya, "Apakah kita sudah sampai?";  
  
}  
  
Woo hoo!;  
Keluar dari mobil;
```

ORACLE
Academy

JFo 6-1
untuk Loop

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.

11

Persyaratan umum dalam sebuah program adalah mengulang sejumlah pernyataan. Biasanya, kode terus mengulangi pernyataan sampai ada perubahan. Kemudian kode keluar dari loop dan dilanjutkan dengan pernyataan berikutnya.

Contoh pseudocode menunjukkan while loop yang berulang hingga boolean areWeThereYet benar.

Loop

- Loop digunakan dalam program untuk eksekusi berulang dari satu atau lebih pernyataan sampai kondisi terminasi tercapai

• Sampai ekspresi salah

- atau

• Untuk jumlah waktu tertentu:

- Saya ingin mencetak angka dari 1 sampai 10 • Saya ingin menghitung jumlah angka dalam rentang tertentu

- Sebuah loop for mengeksekusi sejumlah waktu yang diketahui • for loop juga disebut loop pasti

Dalam pemrograman, ada kalanya Anda ingin bekerja dengan banyak masukan, tetapi Anda ingin menjalankan logika yang sama untuk setiap item masukan.

Pengulangan dimulai dari awal potongan kode, eksekusi logika, dan kemudian kembali ke awal pengulangan dengan input baru, siap mengeksekusi kode lagi.

Apa yang Kami Ketahui

- Dalam skenario Hitung Mundur, inilah yang kami ketahui:

Apa yang Kami Ketahui	Nama Teknis	Kode
Saat perulangan dimulai ... Ekspresi Inisialisasi <code>int i = 10;</code>		
Lanjutkan perulangan jika ...	Ekspresi Kondisi	<code>saya >= 0;</code>
Setelah setiap putaran...	Perbarui Ekspresi	<code>saya--;</code>
Kode untuk mengulang	Pernyataan Kode	<code>System.out.println(i);</code>



untuk Ikhtisar Loop

- Sintaks:

Tajuk

```
for(inisialisasi; kondisi; perbarui){
    Pernyataan kode
    Pernyataan
    kode }//akhir untuk •
```

Tubuh

Ekspresi inisialisasi menginisialisasi perulangan, ini dijalankan hanya sekali, saat perulangan dimulai • Saat ekspresi kondisi bernilai false, perulangan berakhir

- Ekspresi pembaruan dipanggil setelah setiap iterasi melalui loop, ekspresi ini dapat menambah atau mengurangi nilai
- Setiap ekspresi harus dipisahkan dengan titik koma (;)

Ekspresi Inisialisasi

- Dilakukan sekali saat perulangan dimulai •

Memberi tahu kompiler variabel apa (disebut penghitung perulangan) yang digunakan dalam perulangan

- Bisa mulai dari nilai berapa saja, tidak hanya 10

```
System.out.println("Hitung mundur untuk Meluncurkan: ");
```

```
for(int i = 10; i >= 0; i--)  
    { System.out.println(i); }//  
akhir untuk
```

```
System.out.println("Ledakan !");
```

Ekspresi Kondisi

- Perulangan berlanjut selama ekspresi boolean ini

BENAR

- Menggunakan operator pembandingan:

• (`==`, `!=`, `<`, `>`, `<=`, `>=`)

```
System.out.println("Hitung mundur untuk Meluncurkan: ");
```

```
for(int i = 10; i >= 0; i--)  
    { System.out.println(i); }//
```

akhir untuk

```
System.out.println("Ledakan !");
```


Perbarui Ekspresi

- Pernyataan ini dijalankan setelah setiap iterasi dari for loop •

Digunakan untuk mengupdate loop counter

```
System.out.println("Hitung mundur untuk Meluncurkan: ");
```

```
for(int i = 10; i >= 0; i--)  
    { System.out.println(i); }//
```

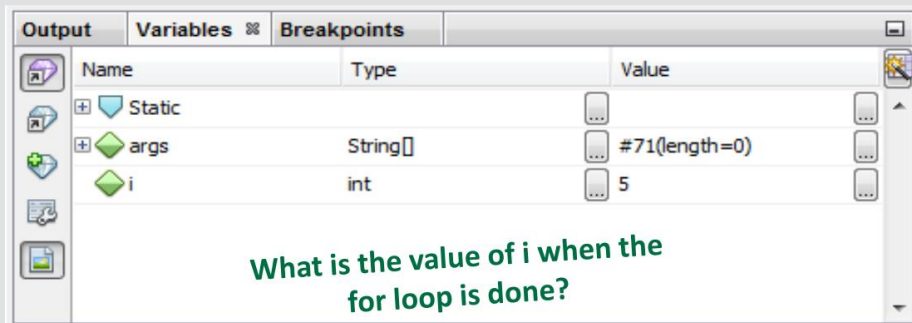
akhir untuk

```
System.out.println("Ledakan !");
```

Latihan 1, Bagian 1

- Buat proyek baru dan tambahkan file Countdown.java ke proyek
- Tetapkan breakpoint di Countdown.java dan amati...

• Bagaimana loop **for** memengaruhi eksekusi kode
• Bagaimana nilai **i** berubah



Latihan 1, Bagian 2

- Dapatkah Anda memodifikasi kode untuk menghitung dari 0 sampai 5?
- Dapatkah Anda mengubah kode untuk menghitung semua angka genap dari 0 sampai 20?

Apakah Saya Membutuhkan Ekspresi Pembaruan?

- Bagaimana jika saya menulis lingkaran saya seperti ini?

```
for(int i = 10; i >= 0; )  
    { System.out.println(i);  
      saya--; }//akhir untuk
```

- Ini juga berhasil! • Tetapi

Anda mungkin tidak ingin membuat kode dengan cara ini, karena perulangan Anda mungkin menjadi lebih rumit

Menghilangkan Ekspresi di for Loop

- Setiap ekspresi di header bersifat opsional
- Namun ada risiko jika Anda menghilangkan ekspresi:

• Tidak ada inisialisasi:

- Tidak ada inisialisasi yang dilakukan •

Mungkin tidak ada penghitung loop

• Tidak ada kondisi:

- Kondisi loop selalu dianggap benar • Loop adalah loop tak terbatas

• Tidak ada

pembaruan: • Tidak ada operasi penambahan yang

dilakukan • Penghitung loop mempertahankan nilai yang sama

Menghilangkan Semua Ekspresi di for Loop

- Periksa kode berikut:

- Ketiga ekspresi dalam perulangan for dapat dihilangkan
- Loop berulang tanpa batas

```
untuk(;;){
```

```
    System.out.println("Selamat Datang di Jawa");
```

```
}//akhir untuk
```

Latihan 2

- Tambahkan file InfiniteLoop.java ke proyek yang Anda buat untuk latihan 1 • Jalankan InfiniteLoop.java dan amati hasilnya
- Modifikasi **for** loop di InfiniteLoop.java untuk mencetak **“Hello”** sebanyak lima kali

Beberapa pernyataan dalam tubuh loop

- Untuk mengeksekusi banyak pernyataan di dalam tubuh ...
- Melampirkan pernyataan di dalam sepasang kurung kurawal
- Jika tidak, hanya pernyataan pertama di tubuh yang dieksekusi

```
untuk(int i = 1; i <= 5; i++)  
System.out.println(i);  
System.out.println(" baris kedua");
```

•Keluaran:

```
1  
2  
3  
4  
5  
baris kedua
```



Salah Satu Penggunaan untuk Loop

- Perulangan for menyediakan cara ringkas untuk mengulang rentang nilai
- Pengulangan tanpa perulangan for:

```
//Mencetak kuadrat dari 1 sampai 5
System.out.println("1 kuadrat = + 1 * 1);
System.out.println("2 kuadrat = + 2 * 2);
System.out.println("3 kuadrat = + 3 * 3);
System.out.println("4 kuadrat = + 4 * 4);
System.out.println("5 kuadrat = + 5 * 5);
```

- Pengulangan dengan for loop: `for(int i =`

```
1; i <= 5; i++){ System.out.println("i
squared = }//end for
+ saya * saya);
```

i Apakah Penghitung Putaran

- Setiap contoh yang kita lihat bergantung pada penghitung putaran

```
for(int i = 1; i <= 5; i++)  
    { System.out.println("i kuadrat = " + i * i);  
    }  
for
```

- saya bisa:

- Dicetak

- Memiliki nilainya berubah

- Be digunakan dalam perhitungan

- Ini bagus untuk:

- Menghitung

- Menghitung nilai dengan cepat

Memahami Ruang Lingkup Variabel

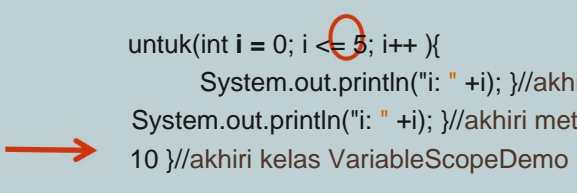
- Tapi `i` hanya ada dalam `for` loop. Ini dikenal sebagai ruang lingkup `i`. Ini tidak ada lagi ketika `for` loop berakhir. Jika `i` digunakan untuk menghitung nilai, kita tidak akan pernah mengeluarkan nilai tersebut dari `for` loop.
- Apakah Anda mengamati saya menghilang saat Anda men-debug `Countdown.java`?

```
for(int i = 1; i <= 5; i++)  
    { System.out.println("i kuadrat = " + i * i); }
```

Cakupan Variabel: Contoh

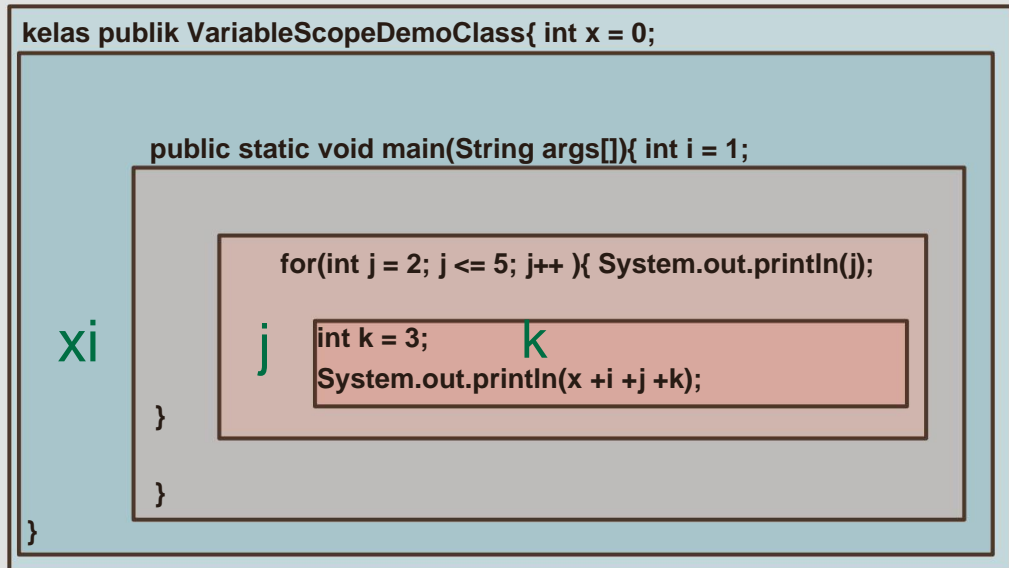
- Variabel `i` yang dideklarasikan pada for loop adalah variabel lokal dan tidak dapat diakses di luar loop
- Compiler error dihasilkan pada baris 8

```
1 kelas publik VariableScopeDemo { 2
3     public static void main(String args[]){
4         untuk(int i = 0; i <= 5; i++){
5             System.out.println("i: " +i); }//akhir untuk
6             System.out.println("i: " +i); }//akhiri metode main
7             10 }//akhiri kelas VariableScopeDemo
8
9
```



Lingkup Variabel

- Variabel tidak dapat eksis sebelum atau di luar bloknya
kode



Penggunaan Lain untuk Loops

- Misalkan Anda perlu menemukan jumlah empat angka

```
import java.util.Scanner; kelas publik
Add4Integers { public static void
    main(String[] args){ Scanner in = new Scanner(System.in);
        System.out.println(" Program ini menambahkan
        empat angka."); System.out.println("Masukkan setiap angka, diikuti dengan
        Enter."); int n1 = masuk.nextInt(); int n2 = masuk.nextInt(); int n3 = masuk.nextInt(); int
        n4 = masuk.nextInt(); int jumlah = n1 + n2 + n3 + n4; System.out.println(" Jumlahnya
        adalah "

                                + jumlah + ".");

        } //akhir metode main } //akhir
        kelas Add4Integers
```

Penggunaan Lain - untuk Loops

- Pendekatan ini sulit untuk diprogram jika Anda ingin menambahkan 100 nilai

```
int n1 = masuk.nextInt(); int  
n2 = masuk.nextInt(); int n3  
= masuk.nextInt(); int n4 =  
masuk.nextInt();  
...  
int n100 = masuk.nextInt(); int  
total = n1 + n2 + n3 + n4 +... + n100;
```

- Bisakah perulangan for membuat program ini lebih singkat?
- Bisakah perulangan for membantu menemukan jumlah variabel bilangan bulat?

Menggunakan Lingkup dengan for Loops

- Ini dapat diselesaikan dengan menggunakan ...

•A untuk loop dengan variabel lingkup yang berbeda

```
import java.util.Scanner; public
class Kode Praktik { public static void
main(String[] args){
    Pemindai masuk = Pemindai baru (System.in); int
    N = 100; int jumlah = 0; System.out.println("
    Program ini menambahkan " for(int i = 0; i < N; i++)
    { System.out.println(("Masukkan angka          + N +   " angka.");
    selanjutnya:");

        int nilai = masuk.nextInt();
        jumlah += nilai; }//akhir untuk
    System.out.println(" Totalnya adalah " +
    total + "."); }// akhir metode utama
```


Cakupan

```
import java.util.Scanner; public
class Kode Praktik { public static void
main(String[] args){
    Pemindai masuk = Pemindai baru (System.in); int
    N = 100; int jumlah = 0; System.out.println("
    Program ini menambahkan " for(int i = 0; i < N; i++)
    { System.out.println(("Masukkan angka          + N + " angka ");
    N total selanjutnya:"); nilai
    int nilai = masuk.nextInt();
    jumlah += nilai; }//akhir untuk
    System.out.println(" Totalnya adalah " +
    total + "."); }// akhir metode utama
```

Latihan 3

- Tambahkan file ScopeTest.java ke proyek Anda dibuat untuk latihan 1
- ScopeTest.java rusak • Bisakah Anda memperbaikinya? • Anda harus mendapatkan keluaran berikut:
64 32 16 8 4 2 1
0 1 2 3 4 5
5 4 3 2 1 0
2 4 8 16 32 64

Variabel Sudah Didefinisikan

- i dibuat sebelum perulangan for
- Lain i tidak dapat berada dalam lingkup yang sama
- Salah satu variabel ini membutuhkan nama yang berbeda

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

```
    int i = 0;
```

```
    for(int i = 64; i > 0; i=i/2 ){ System.out.print(i  
    }  
}
```

Keluar dari ruang lingkup

- j tidak bisa berada di luar ruang lingkup tempat ia dibuat •J yang berbeda dapat dibuat jika ruang lingkup tidak tumpang tindih

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

```
= 0; j<=5; j++){  
j    System.out.print(j + " ");  
}
```

```
untuk(int j = 5; j>=0; j--){  
j    System.out.print(j + " ");  
}
```

```
for(int k = 2; k<=64; k=k*2){ k  
    System.out.print(j + " "); }
```

```
}
```

ORACLE
Academy

JFo 6-1
untuk Loop

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.

36

Apakah Saya Membutuhkan Ekspresi Inisialisasi?

- Bagaimana jika saya menulis lingkaran saya seperti ini?

```
int i = 10; for(  
i >= 0; i--)  
    { System.out.println(i); }//  
akhir untuk
```

- Ini juga berhasil!

• Tapi saya ada di luar ruang lingkup for loop • Jika i hanya dimaksudkan sebagai penghitung loop, variabelnya terbuang percuma

Penyimpanan

• Pertahankan ruang lingkup sempit (sekecil mungkin)

• Variabel liar memperumit kode dan meningkatkan potensi bug

Ringkasan

- Dalam pelajaran ini, Anda seharusnya telah mempelajari cara untuk:
 - Memahami komponen standar untuk loop
 - Memahami cara membuat dan menggunakan perulangan for
 - Memahami ruang lingkup variabel
 - Memahami teknik debug
 - Menjelaskan bagaimana loop tak terbatas terjadi di Jawa



ORACLE
Academy

JFo 6-1
untuk Loop

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.

38

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