Comparator

🡪 metode perbandingan(arg1, arg2) dengan dua argumen yang mewakili objek yang dibandingkan.

Ada 4 metode comparator:

1. Menggunakan operator “==”

* digunakan untuk membandingkan dua atau lebih dari dua objek, Jika mereka merujuk ke objek yang sama maka mereturn true, jika tidak maka mereturn false

class GFG {

    public static void main(String[] args)

    {

        // Get some Strings to compare

        String s1 = "A";

        String s2 = "A";

        String s3 = "A";

        String s4 = new String("A");

        // Compare s1 and s2

        // It should return true as they both

        // refer to same object in memory

        System.out.println(s1 + " == " + s2

                         + ": " + (s1 == s2));

        // Compare s1 and s3

        // It should return true as they both

        // refer to same object in memory

        System.out.println(s1 + " == " + s3

                         + ": " + (s1 == s3));

**Output:**

A == A: true

A == A: true

1. Menggunakan metode equals()

* membandingkan dua string yang diberikan berdasarkan data / konten string.

class GFG {

    public static void main(String[] args)

    {

        // Get some Strings to compare

        String s1 = "A";

        String s2 = "A";

        String s3 = "a";

        String s4 = new String("A");

        // Compare s1 and s2

        // It should return true as they both

        // have the same content

        System.out.println(s1 + " .equals " + s2

                         + ": " + s1.equals(s2));

        // Compare s1 and s3

        // It should return false as they both

        // have the different content

        System.out.println(s1 + " .equals " + s3

                         + ": " + s1.equals(s3));

**Output:**

A .equals A: true

A .equals a: false

1. Menggunakan metode compareTo()

class GFG {

    public static void main(String[] args)

    {

        // Get some Strings to compare

        String s1 = "A";

        String s2 = "A";

        String s3 = "a";

        String s4 = new String("A");

        // Compare s1 and s2

        // It should return 0 as they both

        // have the same ASCII value

        System.out.println(s1 + " .compareTo " + s2

                         + ": " + s1.compareTo(s2));

         // Compare s1 and s4

         // It should return 0 as they both

         // have the same ASCII value

         System.out.println(s1 + " .compareTo " + s4

                          + ": " + s1.compareTo(s4));

**Output:**

A .compareTo A: 0

A .compareTo A: 0

1. Menggunakan metode equalsIgnoreCase()

class GFG {

    public static void main(String[] args)

    {

        // Get some Strings to compare

        String s1 = "A";

        String s2 = "A";

        String s3 = "a";

        String s4 = new String("A");

        // Compare s1 and s2

        // It should return true as they both

        // have the same content

        System.out.println(s1 + " .equalsIgnoreCase " + s2

                         + ": " + s1.equalsIgnoreCase(s2));

          // Compare s1 and s3

        // It should return true as they both

        // have the same content being case insensitive

         System.out.println(s1 + " .equalsIgnoreCase " + s3

                          + ": " + s1.equalsIgnoreCase(s3));

**Output:**

A .equalsIgnoreCase A: true

A .equalsIgnoreCase a: true

1. Menggunakan metode compare()

// Java program to demonstrate

// use of .compareTo operator in Java

class GFG {

    public static void main(String[] args)

    {

        // Get some Strings to compare

        String s1 = "A";

        String s2 = "A";

        String s3 = "a";

        String s4 = new String("A");

        // Compare s1 and s2

        // It should return 0 as they both

        // have the same ASCII value

        System.out.println(s1 + " .compareTo " + s2

                         + ": " + s1.compareTo(s2));

        // Compare s1 and s3

        // It should return -32 as they both

        // have the different ASCII value

        System.out.println(s1 + " .compareTo " + s3

                         + ": " + s1.compareTo(s3));

**Output:**

A .compareTo A: 0

A .compareTo a: -32