

String

Di dalam bahasa pemrograman Java, string merupakan sebuah tipe data reference. Tipe data ini diperlakukan sebagai sebuah obyek. String adalah kumpulan beberapa karakter (char). String sendiri sebenarnya merupakan sebuah *class* yang terdapat dalam *library* Java dan digunakan untuk memanipulasi karakter. Contoh sintaks untuk mendeklarasikan sebuah string:

```
1. String greeting = "Hello World!";
```

Jika kode di atas dijalankan, maka konsol akan menampilkan teks Hello World!

Di dalam kelas **String** terdapat beberapa *constructor* yang memungkinkan kita untuk memberikan nilai awal untuk string dari sumber yang berbeda. Sebagai contoh, misalnya kita ingin membuat variabel String dari sebuah *array character*.

```
1. char[] dicodingChars = { 'd', 'i', 'c', 'o', 'd', 'i', 'n', 'g' };
2. String dicodingString = new String(dicodingChars);
3. System.out.println(dicodingString);
```

Dengan kode di atas, kita telah membuat sebuah *array character* yang berisi beberapa karakter.

Kemudian *array character* tersebut kita gunakan sebagai parameter untuk membuat String baru. Kode tersebut akan menghasilkan teks "dicoding".

Perlu diperhatikan bahwa kelas String bersifat *final*. *Alhasil*, kita tidak bisa mengubah nilai dari objek yang sudah kita buat. Jika kita ingin mengubah nilai tersebut, kita membutuhkan **StringBuffer** dan **StringBuilder**.

Mengetahui Panjang String

Kelas String memiliki sebuah *method* untuk mengetahui panjang dari sebuah string, yakni *method* **length()**.

Fungsi tersebut akan mengembalikan/menghasilkan sejumlah karakter dari string, contohnya:

```
1. String dicoding = "dicoding";
2. int length = dicoding.length();
3. System.out.println(length);
```

Fungsi **length()** pada kode di atas akan menghitung jumlah karakter dari variabel **dicoding**. Jika dijalankan, maka konsol akan menampilkan angka 8.

Mengambil Karakter Dari Sebuah String

Kita juga bisa mengambil sebuah karakter secara spesifik dari sebuah String dengan menggunakan *method* `charAt(int index)` yang sudah tersedia dalam kelas String. Misalnya saat ingin mengambil sebuah karakter dari teks dicoding, kita bisa menggunakan kode berikut:

```
1. String dicoding = "dicoding";
2. char result = dicoding.charAt(7);
3. System.out.println(result);
```

Maka jika Anda menjalankan kode di atas, konsol akan menampilkan karakter urutan ke-8 dari teks tersebut, yaitu "g".

Selain *method* `length()` dan `charAt(int index)`, masih banyak *method* lain yang disediakan oleh kelas String yang bisa mempermudah kita memanipulasi sebuah teks. Berikut adalah kumpulan beberapa *method* yang sering digunakan:

No.	Nama	Deskripsi
1	length()	Digunakan untuk mengetahui panjang atau jumlah karakter dalam string.
2	charAt(int index)	Digunakan untuk mengambil sebuah karakter berdasarkan indeks tertentu.
3	format(String format, Object... args)	Digunakan untuk memformat sebuah string.
4	substring(int beginIndex)	Mengembalikan/menghasilkan substring berdasarkan indeks yang diberikan.
5	contains(CharSequence s)	Mengembalikan/menghasilkan nilai true atau false setelah mencocokkan karakter.
6	equals(Object object)	Memeriksa apakah nilai objek sama dengan nilai string.
7	isEmpty()	Memeriksa apakah sebuah string itu kosong atau tidak.
8	concat(String s)	Mengkonsolidasikan sebuah string.
9	replace(char a, char b)	Mengganti suatu karakter di dalam string.
10	indexOf(String a)	Mengetahui indeks dari sebuah substring.
11	toLowerCase()	Mengubah format string menjadi huruf kecil semua.
12	toUpperCase()	Mengubah format string menjadi huruf kapital semua.
13	trim()	Menghapus spasi awal dan akhir dari string.

14	valueOf(int value)	Mengkonversi tipe yang diberikan menjadi sebuah string.
15	compareTo()	Membandingkan dua nilai

Selengkapnya tentang string, kunjungi: <https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>.

[← Sebelumnya](#)

[Selanjutnya →](#)