



Modul Praktikum Pemrograman Berorientasi Objek

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung

2022
Semester Ganjil

Modul 5. String

String merupakan sebuah tipe data Non-Primitive. Artinya, String merupakan sebuah class tersendiri yang memiliki berbagai method yang dapat digunakan untuk memanipulasi nilai didalamnya.

Capaian Pembelajaran

1. Mahasiswa mampu membuat program yang menerapkan konsep String
2. Mahasiswa mampu membuat program memanfaatkan method-method untuk memanipulasi String pada java

Materi

Java String

String di java mewakili kumpulan dari karakter. Semua string dalam program Java, seperti "abc", diimplementasikan sebagai instance dari kelas String.

String di java tidak dapat diubah setelah dibuat. Karena objek String tidak dapat diubah, objek tersebut dapat diuraikan. Sebagai contoh:

```
String str = "abc";
```

setara dengan:

```
char data[] = {'a', 'b', 'c'};  
String str = new String(data);
```

Berikut adalah beberapa contoh bagaimana string dapat digunakan:

```
System.out.println("abc");  
String cde = "cde";  
System.out.println("abc" + cde);  
String c = "abc".substring(2, 2);  
String d = cde.substring(1, 2);
```

String Concatenation

Penggabungan string dalam cc java dapat dilakukan dengan banyak cara. Diantara cara yang paling sering digunakan adalah sebagai berikut:

Operator (+)

```
String s = "Sachin" + " Tendulkar";
```

Method concat()

```
String firstName = "John ";
String lastName = "Doe";
System.out.println(firstName.concat(lastName));
```

String Method

Berikut list method string pada java

Method	Description	Return Type
charAt()	Returns the character at the specified index (position)	char
codePointAt()	Returns the Unicode of the character at the specified index	int
codePointBefore()	Returns the Unicode of the character before the specified index	int
codePointCount()	Returns the number of Unicode values found in a string.	int
compareTo()	Compares two strings lexicographically	int
compareToIgnoreCase()	Compares two strings lexicographically, ignoring case differences	int
concat()	Appends a string to the end of another string	String
contains()	Checks whether a string contains a sequence of characters	boolean

<u>contentEquals()</u>	Checks whether a string contains the exact same sequence of characters of the specified CharSequence or StringBuffer	boolean
<u>copyValueOf()</u>	Returns a String that represents the characters of the character array	String
<u>endsWith()</u>	Checks whether a string ends with the specified character(s)	boolean
<u>equals()</u>	Compares two strings. Returns true if the strings are equal, and false if not	boolean
<u>equalsIgnoreCase()</u>	Compares two strings, ignoring case considerations	boolean
<u>format()</u>	Returns a formatted string using the specified locale, format string, and arguments	String
<u>getBytes()</u>	Encodes this String into a sequence of bytes using the named charset, storing the result into a new byte array	byte[]
<u>getChars()</u>	Copies characters from a string to an array of chars	void
<u>hashCode()</u>	Returns the hash code of a string	int
<u>indexOf()</u>	Returns the position of the first found occurrence of specified characters in a string	int
<u>intern()</u>	Returns the canonical representation for the string object	String
<u>isEmpty()</u>	Checks whether a string is empty or not	boolean
<u>lastIndexOf()</u>	Returns the position of the last found occurrence of specified characters in a string	int
<u>length()</u>	Returns the length of a specified string	int
<u>matches()</u>	Searches a string for a match against a regular expression, and returns the	boolean

	matches	
offsetByCodePoints()	Returns the index within this String that is offset from the given index by codePointOffset code points	int
regionMatches()	Tests if two string regions are equal	boolean
replace()	Searches a string for a specified value, and returns a new string where the specified values are replaced	String
replaceFirst()	Replaces the first occurrence of a substring that matches the given regular expression with the given replacement	String
replaceAll()	Replaces each substring of this string that matches the given regular expression with the given replacement	String
split()	Splits a string into an array of substrings	String[]
startsWith()	Checks whether a string starts with specified characters	boolean
subSequence()	Returns a new character sequence that is a subsequence of this sequence	CharSequence
substring()	Returns a new string which is the substring of a specified string	String
toCharArray()	Converts this string to a new character array	char[]
toLowerCase()	Converts a string to lower case letters	String
toString()	Returns the value of a String object	String
toUpperCase()	Converts a string to upper case letters	String
trim()	Removes whitespace from both ends of a string	String



valueOf()	Returns the string representation of the specified value	String
-----------	--	--------

Langkah Praktikum

Membuat Project

1. Buka Netbeans
2. Buat Project baru dengan nama **Nama_Praktikum5**

Membuat variabel string

1. Buatlah sebuah variabel **firstname** bertipe data **String**. Kemudian isi nilai dari variabel tersebut dengan nama depan anda.
2. Kemudian buat variabel **lastname** dengan tipe data yang sama. Kemudian isi nilai dari variabel tersebut dengan nama belakang anda.
3. Tampilkan nilai dari kedua variabel tersebut.

```
public static void main(String[] args) {
    String firstname = "Ervan";
    String lastname = "Chodry";

    System.out.println(firstname);
    System.out.println(lastname);
}
```

Menggabungkan Dua Buah String

1. Buatlah sebuah variabel **fullname** dengan tipe data **String**. Isi variabel tersebut dengan menggabungkan String nama depan dengan nama belakang anda dipisahkan dengan space. Contoh: **"Nama Depan" + "Nama Belakang"**

```
public static void main(String[] args) {
    String firstname = "Ervan";
    String lastname = "Chodry";

    String fullname = "Ervan" + " " + "Chodry";

    System.out.println(fullname);
}
```

2. Selanjutnya buatlah variabel **fullname2** dan isi nilainya dengan menggabungkan variabel **firstname** dan **lastname** dipisahkan oleh space. Kemudian tampilkan nilainya.

```
public static void main(String[] args) {
    String firstname = "Ervan";
    String lastname = "Chodry";

    String fullname = "Ervan" + " " + "Chodry";
    String fullname2 = firstname + " " + lastname;

    System.out.println(fullname);
    System.out.println(fullname2);
}
```

Menggunakan Method toUpper dan toLower

1. Buatlah variabel **Upper** dan **Lower** dengan tipe data **String**. Kemudian isi nilai masing-masing variabel dengan **Uppercase** dan **Lowercase** dari variabel **fullname**.
2. Tampilkan nilai dari kedua variabel tersebut.

```
public static void main(String[] args) {  
    String firstname = "Ervan";  
    String lastname = "Chodry";  
  
    String fullname = firstname + lastname;  
  
    String upper = fullname.toUpperCase();  
    String lower = fullname.toLowerCase();  
  
    System.out.println(upper);  
    System.out.println(lower);  
}
```

Menghitung panjang string

1. Buat sebuah variabel **panjang** dengan tipe data **integer**, kemudian isi nilainya dengan panjang dari string **fullname** menggunakan method **length**.
2. Tampilkan nilai dari variabel **panjang**.

```
public static void main(String[] args) {  
    String firstname = "Ervan";  
    String lastname = "Chodry";  
  
    String fullname = firstname + lastname;  
  
    int panjang = fullname.length();  
  
    System.out.println(panjang);  
}
```

Menggunakan method substring

1. Anda dapat mengambil substring dari variabel **fullname** menggunakan method **substring**, dengan 2 parameter. Parameter pertama menunjukkan index awal substring. Parameter kedua menunjukkan index terakhir dari String yang diambil.

2. Cetak sebuah substring dari variabel **fullname**. Silakan anda sesuaikan parameternya sesuai keinginan anda.

```
public static void main(String[] args) {  
  
    String fullname = "Ervan" + " " + "Chodry";  
  
    System.out.println(fullname.substring(3, 5));  
}
```

Hasil running dari program tersebut adalah :

```
Run :  
an
```

Menggunakan method IndexOf

1. Buatlah sebuah variabel **idx** dengan tipe data **integer**.
2. Cari sebuah substring dari variabel **fullname** dengan fungsi IndexOf
3. Isi nilai dari variabel **idx** dengan hasil dari fungsi IndexOf

```
public static void main(String[] args) {  
  
    String fullname = "Ervan" + " " + "Chodry";  
  
    int idx = fullname.indexOf("van");  
  
    System.out.println(idx);  
}
```

Hasil running dari program tersebut adalah :

```
Run :  
2
```

Tugas

1. Buatlah sebuah program yang dapat menerima 2 buah input, yang terdiri dari:

- a. Baris pertama terdiri dari sebuah integer **n** yang menunjukkan jumlah inputan berikutnya
 - b. Lalu, **n** baris berikutnya masing-masing terdiri dari sebuah **String**

Program yang dibuat harus bisa menggabungkan seluruh String yang diinputkan dipisahkan dengan **spasi**. Kemudian tampilkan String yang telah digabungkan tersebut.
2. Buatlah sebuah program yang dapat menerima 2 baris input, yang terdiri dari :
 - a. Baris pertama berisi sebuah String
 - b. Baris kedua terdiri dari 2 buah integer **x** dan **y** yang dipisahkan oleh spasi. **x** menunjukkan index awal substring, dan **y** menunjukkan index akhir dari String yang akan diambil.

Tampilkan substring sesuai kalimat dan input yang diberikan!
3. Buatlah sebuah program yang dapat menerima 2 baris input, baris pertama berisi sebuah String **str1**, dan baris kedua berisi sebuah String **str2**. Program yang dibuat harus bisa memeriksa apakah **str2** merupakan substring dari **str1**. Jika benar maka tampilkan **"Ya"** jika salah maka tampilkan **"Tidak"**
4. Kerjakan tugas tersebut di Vclass!

