String

- Dalam bahasa lain seperti C/C++ dan Pascal (Delphi), string merupakan array karakter.
- Dalam Java, string adalah object dan bukan array karakter.
  - Dapat dibuat array karakter, tetapi ia bukan string.
- String terdiri dari sederetan karakter yang dibatasi oleh tanda petik ganda (double-quote).
- Berikut perbedaan antara deklarasi char dan deklarasi String:

```
char inputKey;
char ampersand = '&';

String perancangJava = "James Gosling";
String strAmpersand = "&";
```

  - Perhatikan tanda kutip tunggal dan kutip ganda yang membedakan karakter dan string.
- Meskipun kedua variabel berisi data satu karakter ampersand, tetapi
  - Variable ampersand mempunyai tipe primitif char.
  - Variable strAmpersand mempunyai tipe object String.

Operasi String

- concat(str) → digunakan untuk mengabungkan 2 buah string string.
- isEmpty() → digunakan untuk mengecek apakah string kosong atau tidak.
- trim() → digunakan untuk membuang spasi di sebelah kiri dan kanan string.
- length() → digunakan untuk menghitung banyak karakter dalam string.
- equals(str) → digunakan untuk membandingkan 2 buah string.
- substring(<awal>, <akhir>) → digunakan untuk mengambil substring dari string.
- charAt(<index>) → digunakan untuk mengambil karakter yang ada pada index.
- lastIndexOf(kar) → digunakan untuk mencari index terakhir dari string yang mengandung karakter kar.
- equalsIgnoreCase(str) → digunakan untuk membandingkan string dengan str, tidak membedakan huruf besar dan kecil.
- copyValueOf(arr_char) → digunakan untuk membentuk string dari array karakter.
- replace(<old>, <new>) → digunakan untuk mengganti <old> menjadi <new>.

Menampilkan String

- Untuk menampilkan string dalam program Java non-GUI (teks), kita dapat memakai perintah System.out.println() ataupun System.out.print().
  - println() akan menambahkan karakter ganti baris (CR+LF) di akhir string, sedang print() tidak.

```
class TampilString {
public static void main(String args[]) {
   System.out.print("Perancang C: ");
   System.out.print("Brian Kernighan dan Dennis Ritchie.");

   System.out.println();          // berganti baris

   System.out.println("Perancang Java: ");
   System.out.println("  James Gosling.");
  }
}
```

- Perhatikan perbedaan antara print() dan println().
- Output program (ditampilkan string sebagai berikut):

```
Perancang C++: Brian Kernighan dan Dennis Ritchie.
Perancang Java:
  James Gosling.
```

- Object String akan mengenali operator penggabungan string (concat) berupa tanda (+).
- Contoh:

```
String s1 = "Java Micro - ";
String s2 = s1 + "Java Enterprise";

System.out.println(s2);
```

- Akan menampilkan string:

```
Java Micro - Java Enterprise
```

- Selain untuk menjumlahkan tipe variable String dengan String, **operator +** juga dapat dipakai untuk menjumlahkan tipe variable String dengan tipe lainnya:
  - String dengan boolean,
  - String dengan integer (byte, short, int, long, char),
  - String dengan floating-point (float, double),
  - String dengan object.

Contoh Program

```
class Concat {
public static void main(String args[ ]) {
   boolean bool = true;
   long intg = 4567890;
   double flot = 45678.9123e-11;

   Concat objt1 = null;
   Concat objt2 = new Concat( );
```

```
    System.out.println("Concatenation:" +
        "\nString + boolean: " + false + " - " + bool +
        "\nString + integer: " + 123 + " ~ " + intg +
        "\nString + float  : " + 3.14 + " ~ " + flot +
        "\nString + object : " + objt1 + " ~ " + objt2);
  }
}
```

String dengan operator new

- Sebuah object String dapat dibuat dengan memakai operator new.
  - hasilnya akan sama saja dengan cara deklarasi yang telah dijelaskan di atas.
- Contoh:

```
String s1 = "JBuilder 9.";
String s2 = new String("JBuilder 9.");
```

  - Kedua variable s1 dan s2 akan sama-sama merupakan object String yang berisikan teks: JBuilder 9.
- Dengan operator new, dapat juga membuat object String dari array karakter.
- Contoh:

```
char chArray[] = {'J', 'B', 'u', 'i', 'l', 'd', 'e', 'r', '9'};
String s1 = new String(chArray);        // s1 = "JBuilder 9."
```

- Dapat juga memilih sub-array dari array karakter untuk dikonversi menjadi object String.
  - Bentuk umumnya adalah sebagai berikut:

```
String [nama var] = new String( char[ ] chArray, int offset, int count )
```

  - Di mana,
    - chArray merupakan array karakter,
    - offset merupakan index awal dari sub-array,
    - count merupakan jumlah karakter yang akan diambil.
  - Contoh:

```
char chAr[] = {'J', 'B', 'u', 'i', 'l', 'd', 'e', 'r', ' ', '9'};
String s1 = new String(chAr, 1, 5);                    //s1 = "Build“
```

- Index array maupun String dimulai dari angka 0 (zero-based index).
  - Dengan offset = 1, maka karakter awalnya adalah 'B' (index 1).
  - Dengan count = 5, akan diambil sebanyak 5 karakter mulai dari 'B'.
  - Sub-array yang diambil adalah: 'Build'.

String sebagai class

- Dalam Java, variable dengan tipe String merupakan sebuah object dari class String.
  - Ketika dideklarasikan sebuah string, maka secara otomatis, compiler Java akan membuatkan sebuah object String.
- Dalam JDK, class String disimpan dalam package java.lang.
  - Jadi full-name bagi class ini adalah: java.lang.String.
- Sebagai sebuah class, String mempunyai beberapa member berupa method.
- Berikut beberapa method penting
  - PANJANG STRING: **LENGTH( )**
  - MENGAMBIL SUB-STRING: **SUBSTRING( )**
  - MENGAMBIL KARAKTER: **CHARAT( )**
  - MENGAMBIL INDEX: **INDEXOF( )**
  - CASING: **TOUPPERCASE( )**, **TOLOWERCASE( )**

LENGTH()

- Method length() dipakai untuk mengambil panjang dari variable String.
- Contoh:

```
String s1 = "JBuilder 9.";
int lenStr = s1.length();          // lenStr = 11
System.out.println(lenStr);        // 11
```

SUBSTRING()

- Method substring() dipakai untuk mengambil sub-string dari sebuah object String.
- Ada dua bentuk dari method ini
  - Bentuk pertama (overload method):

```
String substring(int beginIndex, int endIndex);
String substring(int beginIndex);
```

    - Bentuk tersebut mengambil sub-string dimulai dari posisi beginIndex sampai posisi endIndex-1.
  - Bentuk kedua.

```
// posisi:  "01234567890"
 String s1 = "JBuilder 9.";

 String cutStr = s1.substring(5);       // cutStr = "der 9."
 String subStr = s1.substring(1, 6);    // subStr = "Build“
```

    - Bentuk tersebut mengambil sub-string dimulai dari posisi beginIndex sampai akhir string

CHARAT()

- Method charAt() dipakai untuk mengambil karakter dari object String pada posisi index tertentu.
- Contoh:

```
// posisi:  "01234567890"
String s1 = "JBuilder 9.";

char c1 = s1.charAt(1);   // c1 = 'B'
char c2 = s1.charAt(9);   // c2 = '9'
char c3 = s1.charAt(6);   // c3 = 'e'
```

INDEXOF()

- Method indexOf() dipakai untuk mengambil nilai index berupa posisi karakter tertentu di dalam object String.
  - Posisi karakter yang dicari adalah posisi dari karakter pertama yang ditemui.
- Contoh:

```
// posisi:  "0123456789012345678"
String s1 = "JBuilder untuk Java";

int pos1 = s1.indexOf('u');   // pos1 = 2
int pos2 = s1.indexOf('t');   // pos2 = 11
int pos3 = s1.indexOf('J');   // pos3 = 0
```

- Nilai index yang dikembalikan adalah index dari karakter pertama yang ditemui. Misalnya ada 3 buah karakter 'u' dalam s1, tetapi yang dikembalikan adalah posisi dari karakter 'u' yang pertama (yaitu 2).

TOUPPERCASE(), TOLOWERCASE()

- Method toUpperCase() atau toLowerCase() dipakai untuk konversi semua karakter di dalam object String menjadi huruf besar atau huruf kecil.

```
String toUpperCase();
String toLowerCase();
```

- Contoh:

```
String s1 = "JBuilder untuk Java\u2122";
String sUp = s1.toUpperCase();    // = JBUILDER UNTUK JAVA™
String sDown = s1.toLowerCase();  // = jbuilder untuk java™
```