

String & Penanganan



Definisi String



- String terdiri dari sederetan karakter yang dibatasi oleh tanda petik ganda.
- Menampilkan string dalam program java menggunakan perintah `System.out.println` (untuk menambahkan karakter ganti baris) dan `System.out.print` (cetak tanpa ganti baris).

```
public class Main {  
  
    /**  
     * @param args the command line arguments  
     */  
    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.");  
    }  
}
```

: Output - string1 (run)

```
init:  
deps-jar:  
Created dir: C:\Users\WieAljan\Documents\NetBeansProjects\string1\build\classes  
Compiling 1 source file to C:\Users\WieAljan\Documents\NetBeansProjects\string1\build\classes  
compile:  
run:  
Perancang C: Brian Kernighan dan Dennis Ritchie.  
Perancang Java:  
James Gosling.  
BUILD SUCCESSFUL (total time: 1 second)
```

Class String



- Class String dipakai untuk merepresentasikan data yang berbentuk string.
- Beberapa methods/metode yang terdapat pada class String adalah :
 - ❑ Length()
 - ❑ CharAt(int x)
 - ❑ Concat(String str)
 - ❑ Replace(char lama, char baru)
 - ❑ Substring (int awal, int akhir)
 - ❑ IndexOf()

Length



- Fungsi atau metode **length()** dipakai untuk menghitung panjang dari string.

- Contoh :

```
String s1 = "JBuilder 9.0";
```

```
int lenStr = s1.length(); // lenStr = 11
```

```
System.out.println(lenStr); // 11
```

CharAt(int x)



- Fungsi atau metode `charAt(int x)` dipakai untuk mendapatkan karakter pada posisi `x` dari suatu string.
- Posisi awal suatu string adalah 0, sama seperti index awal pada suatu array.
- 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'
```

Concat (String str)



- Fungsi atau metode `concat(String x)` dipakai untuk menggabungkan string `x` dengan suatu string.
- Contoh : `“to”.concat(“get”).concat(“her”)` akan menghasilkan `“together”`.

Replace (char lama, char baru)



- Fungsi atau metode `replace(char lama, char baru)` dipakai untuk mengubah huruf lama menjadi huruf baru.
- Contoh :
 - `replace("a","i")` akan mengubah huruf a menjadi i.
 - `replace("ri","ya99")` akan mengubah kumpulan huruf ri menjadi ya99.

Substring (int awal, int akhir)



- Fungsi atau metode `substring(int awal, int akhir)` dipakai untuk mengambil sejumlah string dari posisi ‘awal’ sampai ‘akhir’.
- Tapi ‘akhir’ adalah akhir-1.
- Posisi awal suatu string adalah 0.
- Contoh:
 - “Bahasa”.`substring(1,5)` akan mengambil string dari posisi 1 sebanyak 5, Hasilnya “ahas”
 - “Programming”.`substring(3,6)` akan menghasilkan “gra”.

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).

Class StringBuffer



- Kelas yang memungkinkan penambahan atau penyisipan string.
- Ada beberapa method yang terdapat pada class StringBuffer adalah :
 - ❑ Append ()
 - ❑ Insert ()
 - ❑ Delete ()

Append ()

- Fungsi atau metode `append(x)` dipakai untuk menambah karakter atau string yang terdapat pada `x` ke suatu string. **Contoh :**

```
public class Main {  
  
    public static void main(String[] args) {  
        StringBuffer c;  
        c=new StringBuffer("Belajar Java");  
        String b=" Programming";  
        System.out.println("c:"+c);  
        System.out.println("b:"+b);  
        System.out.print("Setelah dikenai perintah " + "c.append(b) maka menjadi :");  
        System.out.println(c.append(b));  
    }  
}
```

Output - append (run)

```
init:  
deps-jar:  
Compiling 1 source file to C:\Users\WieAljan\Documents\NetBeansProjects\append\build\classes  
compile:  
run:  
c:Belajar Java  
b: Programming  
Setelah dikenai perintah c.append(b) maka menjadi :Belajar Java Programming  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Insert (x,y)

- Fungsi atau metode insert(x,y) dipakai untuk menyelipkan karakter atau string yang terdapat pada y ke suatu string pada posisi x.
- **Contoh :**

```
public class Main {  
  
    public static void main(String[] args) {  
        StringBuffer c;  
        c = new StringBuffer("Belajar Programming");  
        System.out.println(c);  
        System.out.print("setelah dikenai " + "insert(8.\"java\"), menjadi :");  
        System.out.println(c.insert(8, "java "));  
    }  
}
```

Output - insert (run)

```
init:  
deps-jar:  
Compiling 1 source file to C:\Users\WieAljan\Documents\NetBeansProjects\insert\build\classes  
compile:  
run:  
Belajar Programming  
setelah dikenaiinsert(8."java"), menjadi :Belajar java Programming  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Delete (awal,akhir)



- Fungsi atau metode delete(awal,akhir) dipakai untuk menghapus string yg terdapat pada posisi 'awal' sampai pada posisi 'akhir'-1. **Contoh :**

```
public class Main {  
  
    public static void main(String[] args) {  
        StringBuffer c;  
        c=new StringBuffer("Belajar Programming");  
        System.out.println(c);  
        System.out.print("Setelah dikenai delete (0,8)) menjadi :");  
        System.out.println(c.delete (0,8));  
    }  
  
}
```

Output - delete (run)

```
init:  
deps-jar:  
Created dir: C:\Users\WieAljan\Documents\NetBeansProjects\delete\build\classes  
Compiling 1 source file to C:\Users\WieAljan\Documents\NetBeansProjects\delete\build\classes  
compile:  
run:  
Belajar Programming  
Setelah dikenai delete (0,8)) menjadi :Programming  
BUILD SUCCESSFUL (total time: 1 second)
```

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";
```

```
String sUp = s1.toUpperCase(); // = JBUILDER UNTUK JAVA
```

```
String sDown = s1.toLowerCase(); // = jbuilder untuk java
```

Latihan



```
1. public class Main {
```

```
    public static void main(String[] args) {  
        StringBuffer a=new StringBuffer("bahasa java");  
        int panjang;
```

```
        panjang = a.length();
```

```
        System.out.println("panjang string \"" +a + "\"adalah"  
+panjang);
```

```
    }
```

```
}
```

```
2. public class Main {
    public static void main(String[] args) {
        String a="java";
        String b=" programming";
        System.out.println("penggabungan \" + a + \" dan \" + b + \" adalah
        "+a.concat(b));
    }
}
```

```
3. public class Main {
    public static void main(String[] args) {
        String a = "java";
        for(int i=0; i < a.length(); i++)
            System.out.println("Karakter " + i + " adalah" + a.charAt(i));
    }
}
```




TERIMA KASIH