

# BAHASA PEMROGRAMAN JAVA

## Pertemuan XIII

# **STRING dan STRING BUFFER**

Oleh  
Achmad Arrosyidi



# **TUJUAN PEMBELAJARAN**

Mahasiswa dapat membandingkan perbedaan String dan String Buffer.



# **MATERI**

1. Kelas String.
  1. Konstruktor
  2. Metode
2. Kelas String Buffer.
  1. Konstruktor
  2. Metode
  3. Metode Append()
  4. Metode Insert()
  5. Operasi Konkatenasi
3. Kelas String Tokenizer
  1. Konstruktor
  2. Metode
4. Perbedaan Kelas String dan Kelas String Buffer
5. Rangkuman
6. Latihan



## 1. KELAS STRING

- String bukan termasuk kategori jenis/tipe dari suatu variabel.
- String adalah kelas yang menangani deretan karakter.
- Kelas ini mempunyai beberapa metode yang digunakan untuk operasi string.
- Operasi ini berguna untuk manipulasi string.



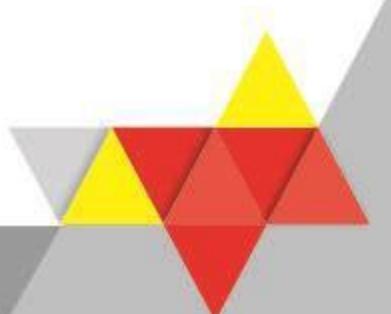
## 1. KELAS **STRING**

- Karakteristik dari objek yang dibentuk dari kelas String adalah kelas yang menyimpan data secara tetap.
- Tidak dapat ditambah ataupun dikurangi jumlah string-nya.
- Dengan kata lain bentuknya adalah tetap tidak berubah.



## 1. KELAS STRING

- Contoh sederhana dari manipulasi string adalah mengkonversi setiap huruf besar/kapital dari sebuah kata atau kalimat menjadi huruf kecil, ataupun sebaliknya.
- Contoh lain dari operasi string adalah menghitung jumlah karakter dalam suatu kata ataupun kalimat.



## 1.1. KONSTRUKTOR KELAS STRING

- Kelas string dalam Java memiliki konstruktor.
- Daftar konstruktor yang dimiliki oleh kelas string

NO	KONSTRUKTOR	KETERANGAN
1	String()	Menciptakan objek String yang berisi string kosong (jumlah karakter sama dengan nol).
2	String(byte[] v, int offset, int jumlah)	Menciptakan objek String yang berisi string yang berasal dari array yang dirujuk oleh v. Dalam hal ini, offset menentukan jumlah karakter diimulai dari posisi offset yang dijakan sebagai string.
3	String(char[] v)	Menciptakan objek string yang berisi string yang berasal dari array yang dirujuk oleh v.
4	String(chart[] v, int offset, int jumlah)	Serupa dengan: String(byte[] v, int offset, int jumlah).
5	String(String v)	Menciptakan objek string yang isinya sama dengan objek string argumennya.
6	String(StringBuffer v)	Menciptakan objek string yang isinya sama dengan string pada objek berkelas String buffer.

## 1.1. KONSTRUKTOR KELAS STRING

- Contoh

```
1 //File: Program_8_1.java
2 public class Program_8_1 {
3     public static void main (String[] arg) {
4         byte data[] = new byte[6]; //membuat array 1 dimensi dengan tipe data byte sebanyak 6 elemen dengan nama data, elemen
5         0 s/d 5
6         data[0]=64; //elemen 0 diisi angka 64
7         data[1]=65; //elemen 1 diisi angka 65
8         data[2]=66; //elemen 2 diisi angka 66
9         data[3]=67; //elemen 3 diisi angka 67
10        data[4]=68; //elemen 4 diisi angka 68
11        data[5]=69; //elemen 5 diisi angka 69
12
13        String s1 = "Selamat pagi"; //s1 diisi dengan text Selamat pagi
14        String s2 = new String("Good morning"); //s2 diisi dengan text Good morning;
15        String s3 = new String(data); //s3 diisi dengan data sebagai konstruktor sehingga s3 berisi kode ASCII dari @ABCDE.
16        Karena kode ASCII 64=@, 65=A, 66=B, 67=C, 68=D, 69=E
17        String s4 = new String(data, 2, 3); //s4 diisi data @ABCDE namun diambil dari posisi ke-2 yaitu huruf B dan diambil 3
18        huruf sesudahnya yaitu BCD
19
20        System.out.println("s1 = " + s1); //akan menampilkan tulisan ke layar s1 = Selamat pagi
21        System.out.println("s2 = " + s2); //akan menampilkan tulisan ke layar s2 = Good morning
22        System.out.println("s3 = " + s3); //akan menampilkan tulisan ke layar s3 = @ABCDE
23        System.out.println("s4 = " + s4); //akan menampilkan tulisan ke layar s4 = BCD
24    }
25 }
```

### Console

```
s1 = Selamat pagi
s2 = Good morning
s3 = @ABCDE
s4 = BCD
<<< Process finished (PID=16100). (Exit code 0)
===== READY =====
```

## 1.2. METODE KELAS STRING

- Beberapa metode pada kelas String

NO	METODE	KETERANGAN
1	copyValueOf(char data[])	Menghasilkan objek berkelas String yang berisi data yang sama dengan isi array data.
2	copyValueOf(char data[], int ofset, int jum)	Serupa dengan copyValueOf() di atas, tetapi string yang dibentuk adalah deretan jumlah karater dimulai dari posisi offset pda array data.
3	valueOf(boolean b)	Menghasilkan objek String yang berisi true jika argumen berupa true dan berisi false jika argumen berupa false.
4	valueOf(char c)	Menghasilkan objek String yang berisi sebuah karakter yang sesuai dengan isi c.
5	valueOf(double d)	Menghasilkan objek String yang berisi bilangan d.

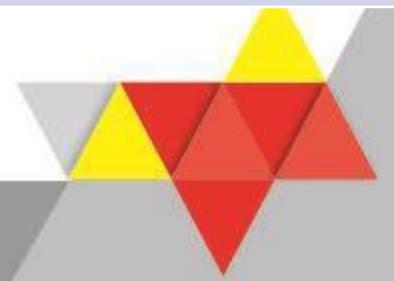


## 1.2. METODE KELAS STRING

NO	METODE	KETERANGAN
6	valueOf(float f)	Menghasilkan objek String yang berisi bilangan f.
7	valueOf(int i)	Menghasilkan objek String yang berisi bilangan i.
8	valueOf(Object obj)	Menghasilkan objek String berisi string yang direpresentasikan oleh objek obj (yaitu string yang bisa diperoleh jika menggunakan metode <code>toString()</code> milik objek tersebut).
9	charAt(int indeks)	Menghasilkan karakter yang terletak pada posisi indeks.
10	compareTo(String s)	<p>Menghasilkan nilai bertipe int yang berupa:</p> <ul style="list-style-type: none"> <li>• Bilangan positif jika string ini lebih besar daripada string s.</li> <li>• Nol jika string ini sama dengan string s.</li> <li>• Bilangan negatif jika string ini lebih kecil daripada string s.</li> </ul> <p>Catatan:</p> <p>Pengertian lebih besar atau lebih kecil didasarkan pada pembandingan karakter pada posisi yang sama.</p>

## 1.2. METODE KELAS STRING

NO	METODE	KETERANGAN
11	concat(String s)	Menghasilkan objek string yang merupakan gabungan dari string ini dengan string s.
12	endsWith(String s)	Menghasilkan nilai true jika string ini berakhiran dengan string s.
13	equals(Object obj)	Menghasilkan nilai true jika string sekarang sama dengan objek obj.
14	equalsIgnoreCase(String s)	Menghasilkan nilai true jika string ini berakhiran dengan string s tanpa memperhatikan huruf kecil atau huruf kapital.
15	getBytes()	Menghasilkan array bertipe byte yang berisi nilai dari setiap karakter dalam string ini.



## 1.2. METODE KELAS STRING

NO	METODE	KETERANGAN
16	indexOf(String ch)	Menghasilkan nilai bertipe int yang menyatakan posisi pertama untuk karakter yang nilainya adalah ch pada string ini. Jika tidak ditemukan, nilai balik berupa -1.
17	indexOf(String s)	Menghasilkan nilai bertipe int yang menyatakan posisi pertama untuk substring s pada string ini. Jika tidak ditemukan maka nilai balik berupa -1.
18	indexOf(String s, int indeks)	Menghasilkan nilai bertipe int yang menyatakan posisi pertama untuk substring s pada string ini dimulai dari posisi indeks. Jika tidak ditemukan maka nilai balik berupa -1.
19	lastIndexOf(int ch)	Menghasilkan nilai bertipe int yang menyatakan posisi terakhir untuk karakter yang nilainya adalah ch pada string ini. Jika tidak ditemukan maka nilai balik berupa -1.
20	lastIndexOf(String s)	Menghasilkan nilai bertipe int yang menyatakan posisi terakhir untuk substring s pada string ini. Jika tidak ditemukan maka nilai balik berupa -1.

## 1.2. METODE KELAS STRING

NO	METODE	KETERANGAN
21	lastIndexOf(String s, int indeks)	Menghasilkan nilai bertipe int yang menyatakan posisi terakhir untuk substring s pada string ini dimulai dari posisi indeks. Jika tidak ditemukan maka nilai balik berupa -1.
22	length()	Menghasilkan nilai bertipe int yang menyatakan jumlah karakter dalam string.
23	replace(char lama, char baru)	Menghasilkan objek String yang berisi string dengan semua karakter yang sesuai dengan argumen pertama diganti dengan karakter pada argumen kedua.
24	startsWith(String s)	Menghasilkan nilai true jika string ini berawalan dengan string s.
25	startsWith(String s, int ofset)	Menghasilkan nilai true jika string ini berawalan dengan string s dimulai dari posisi ofset.
26	Substring(int indeks)	Menghasilkan objek string yang berisi substring dimulai dari karakter pada posisi indeks hingga karakter terakhir pada string ini.

## 1.2. METODE KELAS STRING

NO	METODE	KETERANGAN
27	Substring(int indeksAwal, int indeksAkhir)	Menghasilkan objek String yang berisi substring dimulai dari karakter pada posisi indeksAwal hingga karakter pada posisi karakter pada posisi indeksAkhir pada string ini.
28	toCharArray()	Menghasilkan array bertipe char yang berisi karakter-karakter yang menyusun string ini.
29	toLowerCase()	Menghasilkan objek String dengan seluruh huruf berupa huruf kecil.
30	toString()	Menghasilkan objek string ini.
31	toUpperCase()	Menghasilkan objek String dengan seluruh huruf berupa huruf kapital.
32	Trim()	Menghasilkan objek String yang menghilangkan seluruh spasi ataupun karakter kontrol yang terletak di awal dan di akhir string ini.

## 1.2. METODE KELAS STRING

```
1 //File: Program_8_2.java
2 public class Program_8_2 {
3     public static void main (String[] arg) {
4         String st = " Prodi D3 Manajemen Informatika ... 16.2 "; // membuat objek dengan nama
5         // st dari kelas string, diisi dengan text " Prodi D3 Manajemen Informatika ... 16.2 "
6         // tanpa tanda kutip
7         System.out.println("toLowerCase: " + st.toLowerCase()); //Menampilkan isi dari st
8         // dalam bentuk huruf kecil semua
9         System.out.println("toUpperCase: " + st.toUpperCase()); //Menampilkan isi dari st dalam
10        // bentuk huruf kapital semua
11        System.out.println("tanpa trim: " + '[' + st + ']');
12        System.out.println("dengan trim: " + '[' + st.trim() + ']');
13        System.out.println("Karakter dengan indeks 4: " + st.charAt(4)); //Menampilkan posisi
14        // karakter dari indeks ke-4 yaitu huruf d
15        System.out.println("Posisi 12: " + st.indexOf("16")); //Menampilkan posisi dari
16        // karakter 16
17        System.out.println("Jumlah karakter: " + st.length()); //Menampilkan jumlah panjang
18        // karakter dari st yaitu sebanyak 41 karakter
19        System.out.println("substring(2,5): " + st.substring(2,5)); //Menampilkan text dengan
20        // membuang karakter spasi. Text yang ditampilkan dari indeks ke 0=r, 1=0 dan 2=d dari
21        // Prodi
22    }
23 }
```

- Contoh

```
Console
toLowerCase: prodi d3 manajemen informatika ... 16.2
toUpperCase: PRODI D3 MANAJEMEN INFORMATIKA ... 16.2
tanpa trim: [ Prodi D3 Manajemen Informatika ... 16.2 ]
dengan trim: [Prodi D3 Manajemen Informatika ... 16.2]
Karakter dengan indeks 4: d
Posisi 12: 36
Jumlah karakter: 41
substring(2,5): rod
<<< Process finished (PID=11016). (Exit code 0)
===== READY =====
```



## 2. KELAS **STRING BUFFER**

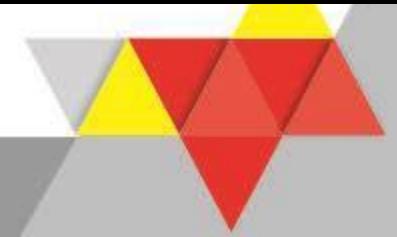
- Kelas yang memungkinkan penambahan atau penyisipan teks.
- Kelas ini memiliki metode bernama `append()` yang digunakan untuk menambahkan string di bagian akhir.



## 2. KELAS STRING BUFFER

```
1  //File: Program_8_3.java
2  public class Program_8_3 {
3      public static void main (String[] arg) {
4          String st = "Prodi D3 Manajemen Informatika"; //Deklarasi dan
5          inisialisasi variabel st bertipe data String yang berisi Prodi
6          D3 Manajemen Informatika
7          int hasil; //Deklarasi variabel hasil yang bertipe data integer
8          boolean logika; //Deklarasi variabel logika yang bertipe data
9          boolean
10
11         System.out.println("compareTo"); //Menampilkan text compareTo
12         ke layar
13
14         hasil = st.compareTo("prodi d3 manajemen informatika");
15         //Membandingkan isi st dengan text prodi d3 manajemen
16         informatika yang hasilnya berupa sebuah angka bilangan negatif
17         karena hasil perbandingan text prodi d3 manajemen informatika
18         lebih kecil daripada isi st, kemudian akan dimasukkan ke
19         variabel hasil
20
21         System.out.println(hasil); // Menampilkan isi variabel hasil ke
22         layar
23
24         hasil = st.compareTo("PRODI D3 MANAJEMEN INFORMATIKA");
25         //Membandingkan isi st dengan text PRODI D3 MANAJEMEN
26         INFORMATIKA yang hasilnya berupa sebuah angka bilangan positif
27         karena hasil perbandingan text PRODI D3 MANAJEMEN INFORMATIKA
28         lebih besar daripada isi st, kemudian akan dimasukkan ke
29         variabel hasil
```

```
14         System.out.println(hasil); // Menampilkan isi variabel hasil ke layar
15
16         hasil = st.compareTo("Prodi D3 Manajemen Informatika");
17         //Membandingkan isi st dengan text Prodi D3 Manajemen Informatika
18         yang hasilnya akan dimasukkan ke variabel hasil
19         System.out.println(hasil); // Menampilkan hasil perbandingan isi st
20         dengan text Prodi D3 Manajemen Informatika, karena lebih besar maka
21         akan menampilkan bilangan positif.
22
23
24         System.out.println("equals dan equalsIgnoreCase"); //Menampilkan
25         text equals dan equalsIgnoreCase ke layar
26
27         logika = st.equals("prodi d3 manajemen informatika");
28         //Membandingkan secara tepat isi st dengan text prodi d3 manajemen
29         informatika yang hasilnya berupa true/false akan dimasukkan ke
30         variabel hasil, karena perbandingan tidak sama maka nilai variabel
31         logika berisi false
32
33         System.out.println(logika); //Menampilkan isi variabel logika ke layar
34
35
36         logika = st.equalsIgnoreCase("prodi d3 manajemen informatika");
37         //Membandingkan dengan tidak memperhatikan bentuk uppercase atau
38         lowercase dari isi st dengan text prodi d3 manajemen informatika
39         yang hasilnya berupa true/false akan dimasukkan ke variabel hasil,
40         karena perbandingan dianggap sama (besar kecil tidak dianggap) maka
41         nilai variabel logika berisi true
42
43         System.out.println(logika); //Menampilkan isi variabel logika ke layar
44     }
```



## 2. KELAS **STRING BUFFER**

Console

```
compareTo
```

```
-32
```

```
32
```

```
0
```

```
equals dan equalsIgnoreCase
```

```
false
```

```
true
```

```
<<< Process finished (PID=17828), (Exit code 0)
```

```
===== READY =====
```



## 2.1. KONSTRUKTOR **KELAS STRING BUFFER**

NO	KONSTRUKTOR	KETERANGAN
1	StringBuffer()	Menciptakan objek StringBuffer yang kosong.
2	StringBuffer(int n)	Menciptakan objek StringBuffer yang dengan n karakter.
3	StringBuffer(String s)	Menciptakan objek StringBuffer yang dengan String berupa s.



## 2.2. METODE KELAS STRING BUFFER

NO	METODE	KETERANGAN
1	capacity()	Menghasilkan kapasitas StringBuffer.
2	charAt(int indeks)	Menghasilkan karakter pada posisi indeks.
3	getChar(int i, int n, char c[])	Menyalin n buah karakter pada StringBuffer dimulai dari posisi i ke array yang dirujuk oleh c.
4	length()	Menghasilkan nilai bertipe int yang menyatakan jumlah karakter dalam StringBuffer.
5	setCharAt(int indeks, char kar)	Mengubah karakter pada posisi indeks dengan karakter kar.
6	setLength(int n)	Mengubah panjang string menjadi n karakter.
7	toString()	Menghasilkan string yang terdapat pada StringBuffer.



## 2.2. METODE KELAS STRING BUFFER

```
1 //File: Program_8_4.java
2 public class Program_8_4 {
3     public static void main (String[] arg) {
4         StringBuffer sbuf = new StringBuffer(25); //Membuat objek sbuf dari kelas StringBuffer dengan kapasitas 25
5
6         System.out.println("Isi: " + sbuf.toString()); //Isi objek sbuf masih kosong
7         System.out.println("Kapasitas: " + sbuf.capacity()); //Kapasitas objek sbuf sebanyak 25 mengacu pada baris ke 4
8         System.out.println("Panjang: " + sbuf.length()); //Karena masih kosong maka panjangnya = 0
9         System.out.println();
10
11        sbuf.append("Prodi "); //Mengisi objek sbuf dengan text Prodi
12        sbuf.append("D3 Manajemen Informatika ");//menambahkan isi objek sbuf dengan text D3 Manajemen Informatika
13
14        System.out.println("Isi: " + sbuf.toString()); //Menampilkan isi sbuf
15        System.out.println("Kapasitas: " + sbuf.capacity()); //Menampilkan kapasitas sbuf
16        System.out.println("Panjang: " + sbuf.length()); //Menampilkan panjang sbuf = 31
17        System.out.println();
18
19        sbuf.setLength(7); //Menset panjang sbuf menjadi 7
20        System.out.println("Isi: " + sbuf.toString()); //Menampilkan isi sbuf
21        System.out.println("Kapasitas: " + sbuf.capacity()); //Menampilkan kapasitas sbuf
22        System.out.println("Panjang: " + sbuf.length()); //Menampilkan panjang sbuf = 7
23    }
24 }
```

## 2.2. METODE KELAS STRING BUFFER

Console

Isi:

Kapasitas: 25

Panjang: 0

Isi: Prodi D3 Manajemen Informatika

Kapasitas: 52

Panjang: 31

Isi: Prodi D

Kapasitas: 52

Panjang: 7

<<< Process finished (PID=13700). (Exit code 0)

===== READY =====



## 2.3. METODE **APPEND()**

- Bentuk metode append() tersedia pada kelas StringBuffer :
- Append(Object obj)
- Append(String st)
- Append(char ch)
- Append(char[] c)
- Append(char[] c, int n, int m)
- Append(boolean b)
- Append(int i)
- Append(long bl)
- Append(float f)
- Append(double d)



## 2.3. METODE APPEND()

```
1 //File: Program_8_5.java
2 public class Program_8_5 {
3     public static void main (String[] args) {
4         StringBuffer sbuf = new StringBuffer("Baris 1\n");
5         sbuf.append(true);
6         sbuf.append('\n');
7
8         sbuf.append((double) 76.8999999);
9         sbuf.append('\n');
10
11        sbuf.append((float) 76.8999999);
12        sbuf.append('\n');
13
14        System.out.println("Isi sbuf: " + sbuf.toString());
15    }
16 }
```

### Console

```
Isi sbuf: Baris 1
true
76.8999999
76.9
```

```
<<< Process finished (PID=16776), (Exit code 0)
===== READY =====
```



## 2.4. METODE **INSERT()**

- Metode `insert()` adalah metode yang digunakan untuk menyisipkan teks dibagian tertentu.
- Berbagai bentuk metode:
  - `insert(int posisi, Object obj)`
  - `insert(int posisi, String s)`
  - `insert(int posisi, char[] c)`
  - `insert(int posisi, boolean b)`
  - `insert(int posisi, char c)`
  - `insert(int posisi, int i)`
  - `insert(int posisi, long l)`
  - `insert(int posisi, float f)`
  - `insert(int posisi, double d)`
- Argumen pertama menyatakan posisi untuk menyisipkan teks.



## 2.4. METODE **INSERT()**

```
1 //Files: Program_8_6.java
2 public class Program_8_6 {
3     public static void main (String[] args) {
4         StringBuffer sbuf = new StringBuffer("ABCDEFGH");
5         sbuf.insert(3, "123");
6         System.out.println("Isi sbuf: " + sbuf.toString());
7     }
8 }
```

---

Console

---

```
Isi sbuf: ABC123DEFGH
<<< Process finished (PID=6972). (Exit code 0)
===== READY =====
```



## 2.5. OPERASI KONKATENASI

- Operasi konkatenasi string yang dilakukan dengan menggunakan operator + sesungguhnya menggunakan kelas StringBuffer.
- Jika terdapat kode seperti

berikut: String s, s1, s2;

s = s1 + s2;

- Java akan memperlakukan sebagai berikut:

s = new StringBuffer().append(s1).append(s2).toString();



### 3. KELAS **STRING TOKENIZER**

- Kelas StringTokenizer terdapat pada paket `java.util`.
- Kelas ini mendukung pemecahan suatu string ke dalam token-token.
- Pemisah antar token dapat ditentukan oleh *programmer*.
- Namun, kalau tidak diatur maka karakter-karakter spasi-putih dijadikan sebagai pemisah antar token



### 3.1. KONSTRUKTOR **KELAS STRINGTOKENIZER**

NO	KONSTRUKTOR	KETERANGAN
1	String Tokenizer(string s)	Membentuk objek dengan s sebagai string yang akan diproses. Pemisah token berupa spasi-putih (spasi, tab, carriage return, atau linefeed).
2	String Tokenizer(string s, string p)	Membentuk objek dengan s sebagai string yang akan diproses dengan p berupa daftar karakter yang menjadi pemisah token.
3	String Tokenizer(string s, string p, boolean b)	Membentuk objek dengan s sebagai string yang akan diproses dan p berupa daftar karakter yang menjadi pemisah token. Jika b bernilai true maka karakter yang menjadi pemisah token akan dikembalikan pula sebagai token.



## 3.2. METODE KELAS STRINGTOKENIZER

NO	METODE	KETERANGAN
1	countTokens()	Memberikan nilai balik bertipe int yang menyatakan jumlah token yang belum diproses.
2	hasMoreElements()	Memberikan nilai balik bertipe boolean yang menyatakan masih ada atoken yang belum diproses.
3	hasMoreToken()	Idem.
4	nextElement()	Meberikan nilai balik bertipe Object yang menyatakan token berikutnya.
5	nextToken()	Meberikan nilai balik bertipe String yang menyatakan token berikutnya.
6	nextToken(String delim)	Meberikan nilai balik bertipe String yang menyatakan token berikutnya dengan permisah token berupa delim.

## 3.2. METODE KELAS STRINGTOKENIZER

```
1 //File: Program_8_7.java
2 import java.util.StringTokenizer;
3
4 public class Program_8_7 {
5     public static void main (String[] args) {
6         String puisi = "Anginpun menyapa\nSelamat Pagi\n" + "diwarnai kehangatan sinar mentari";
7
8         StringTokenizer st = new StringTokenizer(puisi);
9         while(st.hasMoreTokens())
10             System.out.println(st.nextToken());
11     }
12 }
```

Console

```
Anginpun
menyapa
Selamat
Pagi
diwarnai
kehangatan
sinar
mentari
<<< Process finished (PID=5828). (Exit code 0)
===== READY =====
```

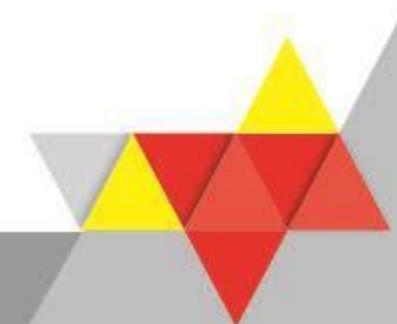


## 3.2. METODE KELAS STRINGTOKENIZER

```
1 //File: Program_8_8.java
2 import java.util.StringTokenizer;
3
4 public class Program_8_8 {
5     public static void main (String[] args) {
6         String info = "11245|Ali Mashudi|Akunting|3500000";
7
8         StringTokenizer st = new StringTokenizer(info, "|");
9         while(st.hasMoreTokens())
10            System.out.println(st.nextToken());
11
12 }
```

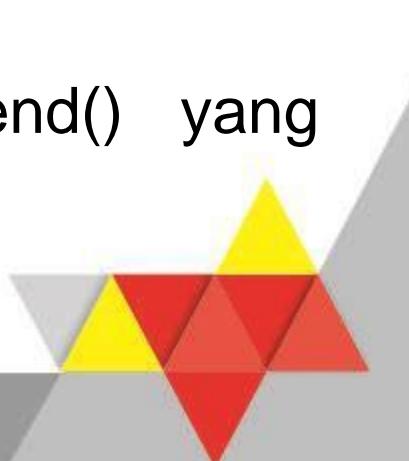
### Console

```
11245
Ali Mashudi
Akunting
3500000
<<< Process finished (PID=15192). (Exit code 0)
===== READY =====
```



## 4. PERBEDAAN **STRING DAN STRING BUFFER**

- Jika kelas String adalah kelas yang menyimpan string yang konstan.
- Sejak objek String telah diciptakan maka string tidak dapat diubah.
- Karena tidak bisa ditambah ataupun dikurangi string yang terbentuk, maka objek dari kelas String tidak memiliki metode append().
- Kelas StringBuffer adalah kelas yang memungkinkan penambahan atau penyisipan teks.
- Itulah sebabnya kelas ini memiliki metode bernama append() yang digunakan untuk menambahkan string di bagian akhir.



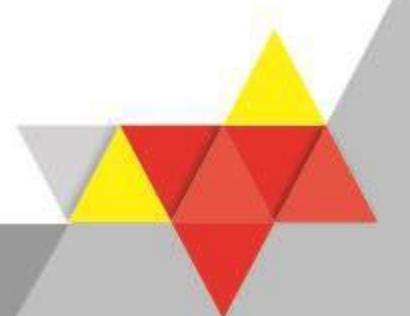
## 5. RINGKASAN

- Didalam java untuk memanipulasi string dapat menggunakan kelas string yang memiliki beberapa metode yang telah disediakan.
- Namun karena keterbatasan dari kelas string yang hanya bisa menyimpan teks secara tetap sehingga dibutuhkan kelas StringBuffer yang memungkinkan penambahan atau penyisipan teks.
- Disamping itu untuk operasi konkatenasi atau menghubungkan string dapat menggunakan operator (+).
- Sebaliknya terdapat kelas StringTokenizer untuk memecah string dengan memanfaatkan metode yang terdapat dalam kelas StringTokenizer.



## 6. LATIHAN

1. Tunjukkan pada baris ke berapa lokasi deklarasi kelas String yang terdapat didalam program 8.1 ContohBab8Ke1.java!
2. Tunjukkan pada baris ke berapa lokasi deklarasi kelas String Buffer yang terdapat didalam program 8.4 ContohBab8Ke4.java!
3. Buatlah program manipulasi string yang didalamnya terdapat penggunaan kelas:
  - a. String!
  - b. String Buffer!



**SELESAI  
TERIMA KASIH**

