

**PRAKTIKUM PBO LANJUT
MODUL PENANGANAN STRING**

A. TUJUAN PRAKTIKUM

Mahasiswa mampu memahami konsep penanganan String.

B. DASAR TEORI

Kelas String

Karakter merupakan bangunan dasar dari bahasa pemrograman Java. Setiap program dibentuk atas sekumpulan karakter bermakna yang diterjemahkan oleh komputer sebagai sekumpulan perintah.

Konstruktor String

Kelas String menyediakan berbagai cara untuk menginisialisasikan sebuah string.

Method Kelas String

- Method **length** merupakan suatu method yang mengembalikan panjang karakter pada suatu string.
- Method **charAt** merupakan suatu method yang mengembalikan karakter pada suatu posisi tertentu.
- Method **getChars** merupakan suatu method yang mengembalikan sekumpulan karakter sesuai posisi yang telah ditentukan. Keluaran kumpulan karakter ditampung dalam suatu larik karakter.

Pembandingan Obyek String

Untuk membandingkan 2 buah string, gunakan method:

- **equals** : keluaran sistem adalah true jika string sama persis
- **equalsIgnoreCase** : keluaran sistem adalah true jika string sama tanpa memperhatikan case sensitif.

Method penting lainnya:

- **contains** : mengetahui apakah suatu string berisi suatu kumpulan karakter tertentu.
- **concat** : menggabungkan 2 string
- **split** : memisahkan string ke suatu larik berdasar karakter pemisah

- **replaceAll** : mengganti kumpulan karakter yang sesuai dengan parameter masukan
- **replaceFirst** : mengganti kumpulan karakter pertama yang sesuai dengan parameter masukan
- **trim** : menghilangkan whitespace pada awal dan akhir String.
- **matches** : mencari string sesuai dengan ekspresi reguler yang diberikan.
- **compareTo** : membandingkan posisi karakter dalam string.
- **toString** : mengembalikan nilai String dari sebuah obyek kelas

Kelas StringTokenizer

Kelas ini merupakan kelas yang memecah suatu string ke dalam beberapa token. Token merupakan kumpulan karakter yang dipisahkan oleh pembatas seperti tabulasi, spasi atau baris baru. Method yang penting:

- **countTokens** : keluaran method adalah jumlah token
- **hasMoreTokens** : keluaran method adalah true jika masih ada token
- **nextToken** : keluaran method adalah token selanjutnya dari obyek StringTokenizer

Latihan

1. Kerjakan latihan dari teori dibawah ini !

- Ch9TestChar.javaFile
- Ch9TestString.javaFile
- Ch9CountVowels
- Ch9CountVowels2
- Ch9CountJava.java
- Ch9CountWords.java
- Ch9ExtractWords.java
- Ch9MatchJavaldentifier.java
- Ch9MatchPhoneNumber.java
- Ch9ReplaceAll.java
- Ch9MatchJavaldentifier2.java
- Ch9PMCountJava.java
- Ch9PMCountJava2.java
- StringBufferInsert.java
- StringBufferAppend.java
- TokenTest.java

2. Buatlah suatu kelas Latihan StringConstructors di bawah ini!

```
import java.util.*;

public class StringConstructors {
```

```

public static void main(String args[]){
    char charArray[] ={'b','i','r','t','h',' ','d','a','y'};
    String s = new String("hello");

    String s1 = new String();
    String s2 = new String(s);
    String s3 = new String(charArray);
    String s4 = new String(charArray,6,3);

    System.out.println("s1 = "+s1);
    System.out.println("s2 = "+s2);
    System.out.println("s3 = "+s3);
    System.out.println("s4 = "+s4);

    if (s.equals("hello"))
        System.out.println("s is equals as \"hello\"");
    else
        System.out.println("s is not equals as \"hello\"");

    if (s.equalsIgnoreCase("HELLO"))
        System.out.println("s is equals as \"hello\"");
    else
        System.out.println("s is not equals as \"hello\"");

    if(s3.contains("day"))
        System.out.println("s3 contains \"day\"");
    else
        System.out.println("s3 do not contains \"day\"");

    String s5 =s.concat(" "+s3);
    System.out.println(s5);

    String s6 = s5.replaceAll("hello","happy");
    System.out.println(s6);

    String[] s7 = new String[3];
    s7 = s6.split(" ");
    for(int i=0;i<s7.length;i++)
        System.out.println(s7[i]);

    StringTokenizer tokens = new StringTokenizer(s6);
    System.out.println("Number of Token =" +tokens.countTokens());

    while (tokens.hasMoreTokens())
        System.out.println(tokens.nextToken());

    }
}

```

3. Buatlah suatu kelas Latihan RegexSubstitution di bawah ini!

```

public class RegexSubstitution
{
    public static void main( String args[] )
    {
        String firstString = "Every day is sunday";
    }
}

```

```

        String secondString ="Vini pergi ke pasar 3 jam lalu, bersama
        4 temannya";

        System.out.printf( "Original String 1: %s\n", firstString );
        firstString = firstString.replaceAll( "\\bday\\b", "week
        there" ); // jelaskan
        System.out.printf(
            "Original string 1 become : %s\n", firstString );

        System.out.printf( "Original String 2: %s\n", secondString );
        secondString = secondString.replaceAll( "[a]", "o"
        );//jelaskan
        System.out.printf(
            "Original string 2 become : %s\n", secondString );

        System.out.printf( "Every word replaced by \"word\": %s\n\n",
            firstString.replaceAll( "\\w+", "word" )); // jelaskan

        secondString = secondString.replaceFirst( "\\d", "digit" );

        // jelaskan
        System.out.printf("?? %s\n", secondString ); //isilah ??
        dengan kalimat yang sesuai

        String output = "String split at commas: [";
        String[] results = secondString.split( ",\\s*" ); // jelaskan

        for ( String string : results ) // jelaskan sintaks ini
            output += "\"" + string + "\", ";
        System.out.println( "Hasil split : "+output );
        output = output.substring( 0, output.length() - 2 ) + "];
        System.out.println( output );
    } // end main
} // end class RegexSubstitution

```

4. Gunakan regular expression untuk menguji masukan string dibawah. Setiap satu nomor berupa satu regular expression.

- Masukan berupa angka 5 – 10 karakter.
- Masukan berakhiran '*' atau '#', diawali dengan angka, karakter berikutnya berupa minimal satu huruf besar dan berikutnya huruf kecil atau huruf besar, panjang minimal 8 karakter.
- Masukan berupa huruf kecil atau huruf besar dan tidak boleh mengandung angka.
- Menghapus tag HTML dari dokumen HTML.
Contoh teks HTML :
<html lang="id">
<head>
<meta charset="UTF-8">

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Contoh Halaman Web</title>
</head>
<body> <header> <h1>Selamat Datang di Situs Kami!</h1> </header>
<main> <p>Ini adalah paragraf pertama dari halaman web contoh ini. HTML
adalah bahasa yang digunakan untuk membuat struktur halaman web.</p>
<p>Anda bisa belajar lebih lanjut tentang HTML di <a
href="https://www.htmlschools.com"
target="_blank">HTMLSchools</a>.</p> </main>
<footer> <p> 2024 Contoh Halaman Web. Semua hak cipta dilindungi.</p>
</footer>
</body>
</html>
```

Akan menghasilkan :

Selamat Datang di Situs Kami!

Ini adalah paragraf pertama dari halaman web contoh ini. HTML adalah bahasa yang digunakan untuk membuat struktur halaman web.

Anda bisa belajar lebih lanjut tentang HTML di HTMLSchools
2024 Contoh Halaman Web. Semua hak cipta dilindungi

C. LAPORAN

1. Penjelasan kode program, capture GUI untuk setiap soal 1-3 (18 x (@4%)
2. Kode program, analisa setiap potongan program tugas 4 (4 x @ 7%)