

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

BAB : COLLECTION STRING
 NAMA : Dani Adrian
 NIM : 225150200111012
 ASISTEN : Tengku Muhammad Rafi Rahardiansyah
 Muhammad Bin Djafar Almasyhur
 TGL PRAKTIKUM : 31 Mei 2023

BAB 10

A. STRING

Pelaksanaan Percobaan

1. String

Pertanyaan

- a. Jelaskan program `JavaString.java` kemudian tuliskan outputnya

Source Code

JavaString.java	
1.	<code>public class JavaString {</code>
2.	<code> public static void main(String[] args) {</code>
3.	<code> String s1 = "Hello";</code>
4.	<code> String s2 = "Hello";</code>
5.	<code> String s3 = new String("Hi");</code>
6.	<code> String s4 = "Hi";</code>
7.	
8.	<code> System.out.println((s1 == s2));</code>
9.	<code> System.out.println((s3 == s4));</code>
10.	
11.	<code> s3 = s3.intern();</code>
12.	<code> System.out.println((s3 == s4));</code>
13.	
14.	<code> System.out.println(s1.concat(s4));</code>
15.	<code> System.out.println(s1.repeat(3));</code>
16.	<code> System.out.println(s1.replace('e', 'o'));</code>
17.	<code> System.out.println(s1.substring(3));</code>
18.	<code> System.out.println(s1.toLowerCase());</code>
19.	<code> System.out.println(s1.toUpperCase());</code>
20.	<code> System.out.println(s1.charAt(1));</code>
21.	<code> System.out.println(s1.indexOf('l'));</code>
22.	<code> System.out.println(s1.equals(s2));</code>
23.	<code> System.out.println(s1.compareTo(s2));</code>
24.	<code> System.out.println(s1.startsWith("He"));</code>

25.	<code>System.out.println(s1.endsWith("lo"));</code>
26.	<code>}</code>
27.	<code>}</code>

Output

The screenshot shows a Java IDE with a dark theme. On the left, the output console displays the following text:

```
true
false
true
HelloHi
HelloHelloHello
Hollo
lo
hello
HELLO
e
-1
true
0
true
true
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeduabelas>
```

On the right, a code editor window titled "Dani Adri" is open, showing the following code:

```
Dani Adrian
225150201111009
```

The code editor window also shows a status bar at the bottom with the text: "Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8".

Penjelasan

Pertanyaan

- b. Jelaskan hasil output line 7,8 dan 10, mengapa bernilai *true* atau *false*

Source Code

1.	Tulis souce code di sini pake courier new 12
----	--

Output

Penjelasan

-Baris 7: `System.out.println((s1 == s2));`

Output: true

Penjelasan: Karena s1 dan s2 memiliki nilai yang sama yaitu "Hello" dan keduanya dideklarasikan dengan menggunakan literal string, maka keduanya akan merujuk pada objek string yang sama di dalam string pool Java. Oleh karena itu, perbandingan menggunakan operator "==", menghasilkan nilai true.

-Baris 8: `System.out.println((s3 == s4));`

Output: false

Penjelasan: s3 adalah objek string yang dibuat dengan menggunakan operator "new", sehingga secara eksplisit membuat objek baru di dalam heap memory, bukan di dalam string pool. Sementara itu, s4 dideklarasikan dengan menggunakan literal string, sehingga merujuk pada objek string yang sama di dalam string pool. Karena objek s3 dan s4 berbeda, maka perbandingan menggunakan operator "==", menghasilkan nilai false.

-Baris 10: `System.out.println((s3 == s4));`

Output: true

Penjelasan: Setelah menjalankan perintah `s3 = s3.intern()`, s3 akan ditempatkan ke dalam string pool (jika belum ada di dalamnya). Kemudian, karena s4 juga merujuk pada objek string yang ada di dalam string pool, perbandingan menggunakan operator "==", menghasilkan nilai true.

Pertanyaan

- c. Jelaskan beda dari method *equal* dan *compareTo* pada string

Source Code

1.	Tulis souce code di sini pake courier new 12
----	--

Output

Penjelasan

Method `equals(Object obj)`: Method ini digunakan untuk membandingkan isi dari dua objek string. Method ini mengembalikan nilai boolean, yaitu `true` jika isi kedua objek string sama dan `false` jika tidak sama.

Method `compareTo(String anotherString)`: Method ini digunakan untuk membandingkan urutan leksikografis antara dua objek string. Method ini mengembalikan nilai integer, yaitu 0 jika kedua objek string sama, nilai negatif jika objek string pemanggil berada sebelum objek string yang dibandingkan, dan nilai positif jika objek string pemanggil berada setelah objek string yang dibandingkan.

2. StringBuffer

Pertanyaan

- Jalankan program `JavaStringBuffer.java` kemudian tuliskan outputnya

Source Code

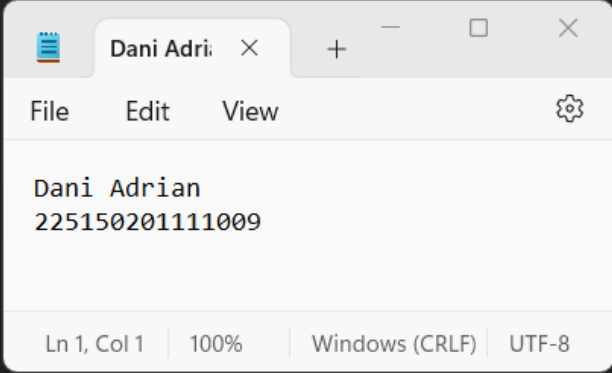
JavaStringBuffer.java	
1.	<code>public class JavaStringBuffer {</code>
2.	<code> public static void main(String[] args) {</code>
3.	<code> StringBuffer sb = new StringBuffer();</code>
4.	<code> System.out.println(sb.capacity());</code>
5.	<code> System.out.println(sb.toString());</code>
6.	<code> StringBuffer sb1 = new StringBuffer(1000);</code>
7.	<code> System.out.println(sb1.capacity());</code>
8.	<code> System.out.println(sb1.toString());</code>
9.	<code> StringBuffer sb2 = new StringBuffer("Hello");</code>
10.	<code> System.out.println(sb2.capacity());</code>
11.	<code> System.out.println(sb2.toString());</code>
12.	<code> CharSequence seq = new StringBuffer("Hello");</code>
13.	<code> StringBuffer sb3 = new StringBuffer(seq);</code>
14.	<code> System.out.println(sb3.capacity());</code>
15.	<code> System.out.println(sb3.toString());</code>
16.	<code> sb3.append(" Java");</code>
17.	<code> System.out.println(sb3.toString());</code>
18.	<code> sb3.insert(10, " Programming");</code>
19.	<code> System.out.println(sb3.toString());</code>
20.	<code> System.out.println(sb3.compareTo(sb2));</code>
21.	<code> System.out.println(sb3.equals(sb2));</code>
22.	<code> System.out.println(sb3.charAt(2));</code>
23.	<code> System.out.println(sb3.substring(3));</code>
24.	<code> System.out.println(sb3.length());</code>
25.	<code> System.out.println(sb3.reverse());</code>
26.	<code> }</code>
27.	<code>}</code>

Output

```

16
1000
21
Hello
21
Hello
Hello Java
Hello Java Programming
17
false
1
lo Java Programming
22
gnimmargorP avaJ olleH
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeduabelas>

```



Penjelasan

Pertanyaan

- b. Mengapa baris 5 dan 8 hanya ditampilkan output kosong?

Source Code

1.	Tulis souce code di sini pake courier new 12
----	--

Output

Penjelasan

Baris 5 dan 8 hanya menampilkan output kosong karena pada saat pembuatan objek StringBuffer dengan menggunakan konstruktor default atau konstruktor dengan kapasitas awal tertentu, objek StringBuffer tersebut tidak memiliki nilai atau karakter awal yang ditetapkan. Sehingga saat mengubah objek StringBuffer menjadi string menggunakan toString(), nilainya kosong.

Pertanyaan

- c. Mengapa pada baris 10 kalau dilihat kapasitasnya akan mengeluarkan angka 21 padahal string "Hello" hanya terdiri dari 5 karakter?

Source Code

1.	Tulis souce code di sini pake courier new 12
----	--

Output**Penjelasan**

Pada baris 10, kapasitas StringBuffer sb2 menghasilkan angka 21 karena saat objek StringBuffer dibuat dengan menggunakan konstruktor StringBuffer(String str), kapasitasnya akan dihitung berdasarkan panjang string awal ditambah dengan 16. Dalam hal ini, string "Hello" memiliki panjang 5, sehingga kapasitas yang dihasilkan adalah $5 + 16 = 21$.

Pertanyaan

- d. Apakah bedanya method *append* dan *insert* pada klas *StringBuffer*?

Source Code

1.	Tulis souce code di sini pake courier new 12
----	--

Output**Penjelasan**

Perbedaan antara method `append()` dan `insert()` pada kelas `StringBuffer` adalah sebagai berikut:

-Method `append(String str)`: Method ini digunakan untuk menambahkan string baru di akhir objek `StringBuffer`. Method ini akan mengubah objek `StringBuffer` yang ada dan menambahkan string baru ke akhirnya.

-Method `insert(int offset, String str)`: Method ini digunakan untuk menyisipkan string baru pada posisi yang ditentukan dalam objek `StringBuffer`. Method ini akan mengubah objek `StringBuffer` yang ada dan menyisipkan string baru pada posisi yang diinginkan, dengan menggeser karakter yang ada setelah posisi tersebut.

3. StringBuilder

Pertanyaan

- a. Jalankan program `JavaStringBuilder.java` kemudian tuliskan outputnya

Source Code

JavaStringBuilder.java	
1.	<code>public class JavaStringBuilder {</code>
2.	<code> public static void main (String[] args) {</code>
3.	<code> StringBuilder sb = new StringBuilder();</code>
4.	<code> System.out.println(sb.capacity());</code>
5.	<code> System.out.println(sb.toString());</code>
6.	<code> StringBuilder sb1 = new StringBuilder(1000);</code>
7.	<code> System.out.println(sb1.capacity());</code>
8.	<code> System.out.println(sb1.toString());</code>
9.	<code> StringBuilder sb2 = new StringBuilder("Java");</code>
10.	<code> System.out.println(sb2.capacity());</code>
11.	<code> System.out.println(sb2.toString());</code>
12.	<code> CharSequence seq = new StringBuilder("String");</code>
13.	<code> StringBuilder sb3 = new StringBuilder(seq);</code>
14.	<code> System.out.println(sb3.capacity());</code>
15.	<code> System.out.println(sb3.toString());</code>
16.	<code> sb3.append(" Java");</code>
17.	<code> System.out.println(sb3.toString());</code>
18.	<code> sb3.insert(11, " Builder");</code>
19.	<code> System.out.println(sb3.toString());</code>
20.	<code> System.out.println(sb3.compareTo(sb2));</code>
21.	<code> System.out.println(sb3.equals(sb2));</code>
22.	<code> System.out.println(sb3.charAt(2));</code>
23.	<code> System.out.println(sb3.substring(3));</code>
24.	<code> System.out.println(sb3.length());</code>
25.	<code> System.out.println(sb3.reverse());</code>
26.	<code> }</code>

27.	}
-----	---

- b. Apabila pada baris 17 ditampilkan juga capacity dari sb3, berapakah output yang dihasilkan? Jelaskan alasannya
- c. Apabila pada baris 19 ditampilkan juga capacity dari sb3, berapakah output yang dihasilkan? Jelaskan alasannya.

Source Code

```

1. public class JavaStringBuilder {
2.     public static void main(String[] args) {
3.         StringBuilder sb = new StringBuilder();
4.         System.out.println(sb.capacity());
5.         System.out.println(sb.toString());
6.         StringBuilder sb1 = new StringBuilder(1000);
7.         System.out.println(sb1.capacity());
8.         System.out.println(sb1.toString());
9.         StringBuilder sb2 = new StringBuilder("Java");
10.        System.out.println(sb2.capacity());
11.        System.out.println(sb2.toString());
12.        CharSequence seq = new StringBuilder("String");
13.        StringBuilder sb3 = new StringBuilder(seq);
14.        System.out.println(sb3.capacity());
15.        System.out.println(sb3.toString());
16.        sb3.append(" Java");
17.        System.out.println(sb3.toString());
18.        System.out.println("Penambahan capacity pertanyaan
B");
19.        System.out.println(sb3.capacity());
20.        sb3.insert(11, " Builder");
21.        System.out.println(sb3.toString());
22.        System.out.println("Penambahan capacity pertanyaan
C");
23.        System.out.println(sb3.capacity());
24.        System.out.println(sb3.compareTo(sb2));
25.        System.out.println(sb3.equals(sb2));
26.        System.out.println(sb3.charAt(2));
27.        System.out.println(sb3.substring(3));
28.        System.out.println(sb3.length());
29.        System.out.println(sb3.reverse());
30.    }

```


31. }

Output

The screenshot shows a Java IDE with a dark theme. On the left, the output console displays the following text:

```

16
1000
20
Java
22
String
String Java
Penambahan capacity pertanyaan B
22
String Java Builder
Penambahan capacity pertanyaan C
22
9
false
r
ing Java Builder
19
redliuB avaJ gnirts
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeduabelas>

```

On the right, a preview window titled "Dani Adri:" is open, showing the text:

```

Dani Adrian
225150201111009

```

Below the text in the preview window, it shows "Ln 1, Col 1", "100%", "Windows (CRLF)", and "UTF-8".

Penjelasan

Penjelasan B

Jika pada baris 17 ditampilkan juga capacity dari sb3, output yang dihasilkan akan menjadi 22. Hal ini disebabkan oleh karena pada baris 17, sb3 dibuat menggunakan konstruktor `StringBuilder(CharSequence seq)`, yang menggunakan panjang dari objek `CharSequence` (dalam hal ini "String") untuk menentukan kapasitas awal `StringBuilder`. String "String" memiliki panjang 6 karakter, sehingga kapasitas awal `StringBuilder` akan menjadi $6 + 16 = 22$.

Penjelasan C

Jika pada baris 19 ditampilkan juga capacity dari sb3, output yang dihasilkan tetap menjadi 22. Kapasitas `StringBuilder` tidak akan berubah ketika menggunakan method `append()` atau `insert()`. Kapasitas `StringBuilder` hanya akan meningkat jika string yang ditambahkan melebihi kapasitas awalnya.

32. StringJoiner

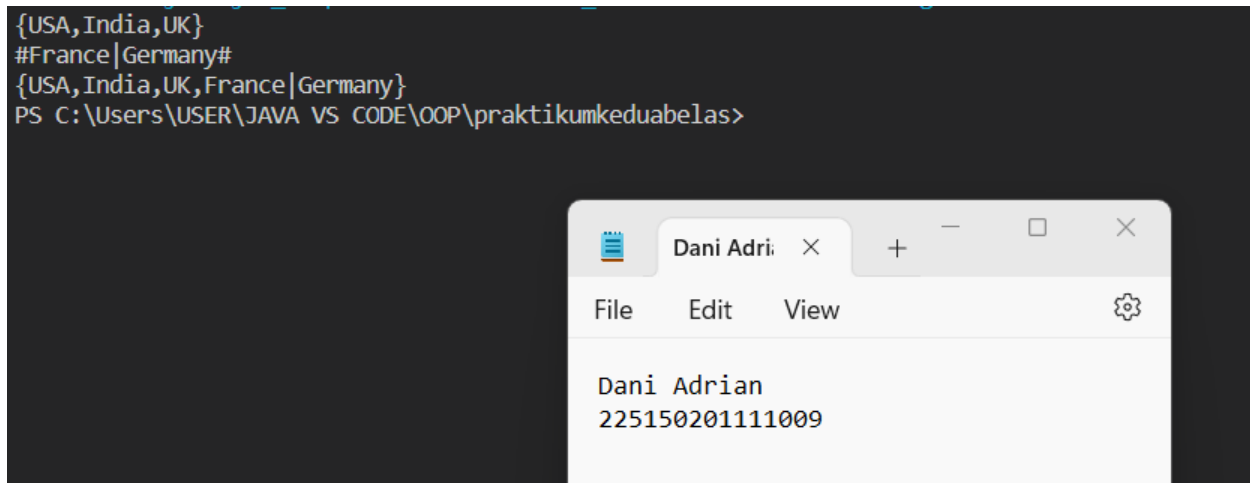
Pertanyaan

- a. Jalankan program `JavaStringJoiner.java` kemudian tuliskan outputnya

Source Code

JavaStringJoiner.java	
1.	<code>import java.util.StringJoiner;</code>
2.	
3.	<code>public class JavaStringJoiner {</code>
4.	<code> public static void main(String[] args) {</code>
5.	<code> StringJoiner sj = new StringJoiner(",", "{", "}");</code>
6.	<code> sj.add("USA");</code>
7.	<code> sj.add("India");</code>
8.	<code> sj.add("UK");</code>
9.	<code> System.out.println(sj);</code>
10.	<code> StringJoiner sj1 = new StringJoiner(" ", "#",</code>
	<code> "#");</code>
11.	<code> sj1.add("France");</code>
12.	<code> sj1.add("Germany");</code>
13.	<code> System.out.println(sj1);</code>
14.	<code> sj.merge(sj1);</code>
15.	<code> System.out.println(sj);</code>
16.	<code> }</code>
17.	<code>}</code>

Output



Pertanyaan

- b. Apabila pada baris 13 diganti menjadi `sj1.merge (sj);`

Source Code

```

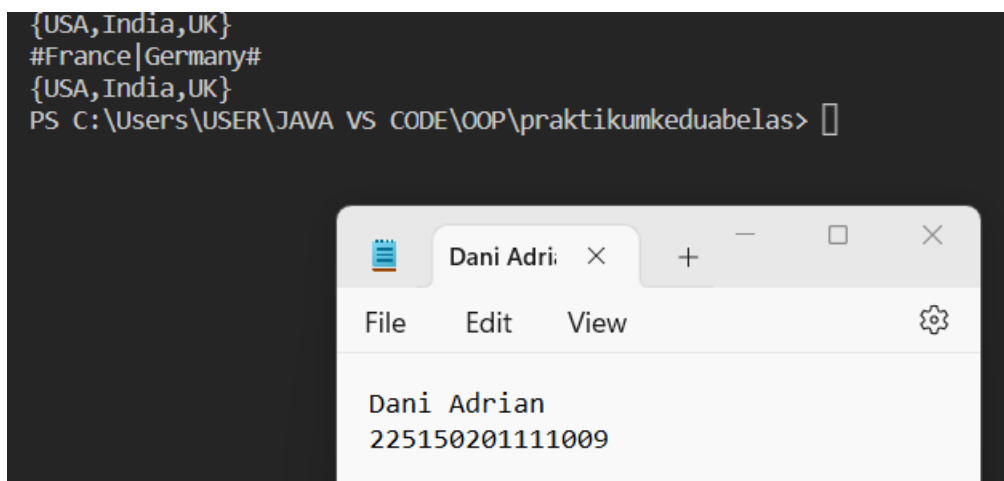
JavaStringJoiner.java
1. import java.util.StringJoiner;
2.
3. public class JavaStringJoiner {
4.     public static void main(String[] args) {
5.         StringJoiner sj = new StringJoiner(",", "{", "}");
6.         sj.add("USA");
7.         sj.add("India");
8.         sj.add("UK");
9.         System.out.println(sj);
10.        StringJoiner sj1 = new StringJoiner("|", "#",
11.        "#");
12.        sj1.add("France");
13.        sj1.add("Germany");
14.        System.out.println(sj1);
15.        sj1.merge(sj);
16.        System.out.println(sj);
17.    }
18. }

```

Pertanyaan

- c. Apakah output yang dihasilkan? Jelaskan alasannya

Output



```

{USA,India,UK}
#France|Germany#
{USA,India,UK}
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeduabelas>

```

Dani Adri: × + − □ ×

File Edit View ⚙

Dani Adrian
225150201111009

Penjelasan

Output berbeda dengan output sebelumnya karena pada baris 13, `sj1.merge(sj)` digunakan untuk menggabungkan dua objek `StringJoiner`, yaitu `sj1` dan `sj`. Operasi merge ini akan menggabungkan elemen-elemen dari `sj` ke `sj1`. Setelah operasi merge, `sj1` akan berisi elemen-elemen dari `sj` yang telah digabungkan. Akan tetapi di sini yang diprint adalah `sj` bukan `sj1` sehingga membuat output yang dikeluarkan sesuai dengan elemen `sj`.

33. StringTokenizer

Pertanyaan

- Jalankan program `JavaStringTokenizer.java` kemudian tuliskan outputnya

Source Code

JavaStringTokenizer.java	
1.	<code>import java.util.Iterator;</code>
2.	<code>import java.util.StringTokenizer;</code>
3.	
4.	<code>public class JavaStringTokenizer {</code>
5.	<code> public static void main(String[] args) {</code>
6.	<code> String csv = "Java,Python,Android";</code>
7.	<code> StringTokenizer st = new StringTokenizer(csv,</code>
	<code>" , ");</code>
8.	<code> printTokens(st);</code>
9.	<code> String line = "Welcome to</code>
	<code>JavaString.net\nlearn\tJava\tProgramming";</code>
10.	<code> StringTokenizer st1 = new StringTokenizer(line);</code>
11.	<code> printTokens(st1);</code>
12.	<code> StringTokenizer st2 = new StringTokenizer("Hi</code>
	<code>Hello Yes");</code>
13.	<code> System.out.println(st2.countTokens());</code>
14.	<code> st2.nextToken();</code>
15.	<code> System.out.println(st2.countTokens());</code>
16.	<code> StringTokenizer st3 = new</code>
	<code>StringTokenizer("Hi Hello Yes", " ", true);</code>
17.	<code> System.out.println(st3.countTokens());</code>
18.	<code> printTokens(st3);</code>
19.	<code> StringTokenizer st4 = new StringTokenizer("Hello</code>
	<code>Java World");</code>
20.	<code> Iterator<Object> iterator = st4.asIterator();</code>
21.	<code> while (iterator.hasNext()) {</code>
22.	<code> System.out.println(iterator.next());</code>
23.	<code> }</code>
24.	<code> }</code>
25.	

```

26.     private static void printTokens(StringTokenizer st) {
27.         while (st.hasMoreTokens()) {
28.             System.out.println(st.nextToken());
29.         }
30.     }
31. }

```

b. Mengapa pada baris ke 9 bisa dipisah token berdasarkan lebih dari satu delimiter?

Output

```

Java
Python
Android
Welcome
to
JavaString.net
learn
Java
Programming
3
2
5
Hi
|
Hello
|
Yes
Hello
Java
World
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeduabelas>

```

Penjelasan

Pada baris ke-9, objek StringTokenizer st1 dibuat dengan menggunakan string line sebagai argumen pertama tanpa menyediakan argumen kedua. Ketika tidak ada argumen kedua yang diberikan, delimiter default yang digunakan oleh StringTokenizer adalah spasi.

StringTokenizer secara default akan memisahkan string menjadi token-token berdasarkan spasi. Namun, StringTokenizer juga dapat memisahkan string berdasarkan delimiter tambahan seperti tab ("t"), karakter baru ("n"), atau karakter kustom lainnya jika argumen kedua diberikan.

Dalam kasus ini, string line memiliki spasi dan tab sebagai delimiter. Jadi, ketika objek StringTokenizer st1 dibuat tanpa menyediakan argumen kedua, spasi dan tab akan dianggap sebagai delimiter default.

B. Tugas Praktikum

Pertanyaan

1. Buatlah program yang dapat memecah String menjadi suatu token dan menghitung frekuensi masing-masing token untuk paragraph berikut ini!

Saya belajar Bahasa Java. Bahasa Java mempunyai kelas berupa String. Belajar String Java itu mudah. String Java juga sering diimplementasikan.

Token	Frekuensi
Saya	1
belajar	2
bahasa	3
Java	4
....
dst	dst

Source Code

```
import java.util.*;

public class StringJava {
    public static void main(String[] args) {
        String Kalimat = new String("Saya belajar bahasa
Java.");
        String Kalimat2 = new String("Bahasa Java mempunyai
kelas berupa String.");
        String Kalimat3 = new String("Belajar String Java itu
mudah.");
        String Kalimat4 = new String("String Java juga sering
diimplementasikan.");
        StringJoiner paragraf = new StringJoiner(" ");
        paragraf.add(Kalimat);
        paragraf.add(Kalimat2);
        paragraf.add(Kalimat3);
        paragraf.add(Kalimat4);

        Map<String, Integer> frequencyMap = new HashMap<>();
        StringTokenizer tokenizer = new
StringTokenizer(paragraf.toString(), " .:(),");

        while (tokenizer.hasMoreTokens()) {
            String token = tokenizer.nextToken().toLowerCase();
            frequencyMap.put(token,
```

```

frequencyMap.getOrDefault(token, 0) + 1);
    }
    System.out.println("Paragraf:");
    System.out.println("\t"+paragraf.toString());
    System.out.println();
    System.out.println("Tabel Hitung Token:");

    System.out.println("=====
=====");
    System.out.printf("|%20s%-10s%-4s%2s%-13s|\n", "Token", "
", "|", "", "Frekuensi");

    System.out.println("=====
=====");
    for (Map.Entry<String, Integer> entry :
frequencyMap.entrySet()) {
        String token = entry.getKey();
        int frequency = entry.getValue();
        System.out.printf("|%20s%-10s%-4s%5s%-
10s|\n", token, " ", "|", "", frequency);

    }

    System.out.println("=====
=====");
    }
}

```

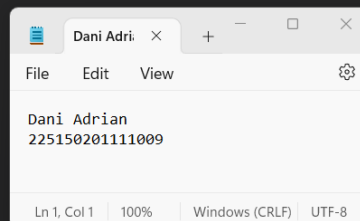
Output

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeduabelas> java -cp .\src\String.java
Paragraf:
    Saya belajar bahasa Java. Bahasa Java mempunyai kelas berupa String. Belajar String Java itu mudah. String Java juga sering diimplementasikan.

Tabel Hitung Token:
=====
|                Token                |   Frekuensi   |
=====
|          saya          |             1  |
|         sering         |             1  |
|        bahasa         |             2  |
|       string          |             3  |
|      berupa           |             1  |
|     mempunyai         |             1  |
|         itu           |             1  |
|        java           |             4  |
|       juga            |             1  |
|      mudah            |             1  |
|     kelas             |             1  |
|    belajar            |             2  |
| diimplementasikan     |             1  |
=====
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeduabelas>

```



Penjelasan

Program di atas adalah contoh program yang menghitung frekuensi token (kata) dalam sebuah paragraf. Berikut penjelasan dari kode tersebut:

1. Pertama, kita membuat beberapa objek String yang berisi kalimat-kalimat yang akan digabungkan menjadi sebuah paragraf.
2. Kemudian, kita membuat objek StringJoiner bernama ``paragraf`` untuk menggabungkan kalimat-kalimat menjadi satu paragraf dengan pemisah spasi.
3. Selanjutnya, kita membuat objek ``frequencyMap`` dari kelas HashMap yang akan digunakan untuk menyimpan token dan frekuensi kemunculannya.
4. Kita menggunakan objek StringTokenizer untuk memecah paragraf menjadi token-token menggunakan pemisah yang telah ditentukan, yaitu " .:(),\". Setiap token yang ditemukan, kita konversi menjadi huruf kecil menggunakan ``toLowerCase()`` dan kemudian kita masukkan ke dalam ``frequencyMap``.
5. Dalam loop while, kita menggunakan metode ``hasMoreTokens()`` untuk mengecek apakah masih ada token yang tersisa. Jika ada, kita ambil token selanjutnya menggunakan ``nextToken()``.
6. Setiap token yang diambil, kita cek apakah sudah ada dalam ``frequencyMap``. Jika sudah ada, kita tambahkan frekuensinya dengan 1 menggunakan ``getOrDefault()`` dan memasukkan kembali ke dalam ``frequencyMap``. Jika belum ada, kita masukkan token tersebut ke dalam ``frequencyMap`` dengan frekuensi awal 1.
7. Setelah selesai menghitung frekuensi token, kita mencetak paragraf awal menggunakan ``paragraf.toString()``.
8. Selanjutnya, kita mencetak tabel hitung token dengan format yang telah ditentukan. Pada bagian ini, kita menggunakan loop ``for-each`` untuk mengiterasi setiap entri dalam ``frequencyMap``.
9. Dalam setiap iterasi, kita ambil token dan frekuensi dari entri saat ini menggunakan metode ``getKey()`` dan ``getValue()``. Kemudian, kita mencetak token dan frekuensinya dengan format yang telah ditentukan.
10. Setelah selesai mencetak tabel, kita menampilkan garis pemisah yang terdiri dari karakter \"=\".