

**LAPORAN PRAKTIKUM
PEMROGRAMAN OBJEK 2**

MODUL 7

STRING, CHARACTER, DAN REGULAR EXPRESSION

DISUSUN OLEH :

FATHIR AHMAD NURPADLI - 2250081132



**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN INFORMATIKA
UNIVERSITAS JENDERAL ACHMAD YANI
TAHUN 2024**

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR.....	v
BAB I. HASIL PRAKTIKUM.....	1
I.1 Program 1	1
I.1.A. Source Code	1
I.1.B. Screenshot.....	1
I.1.C. Analisa	1
I.2 Program 2	2
I.2.A. Source Code	2
I.2.B. Screenshot.....	3
I.2.C. Analisa	3
I.3 Program 3	3
I.3.A. Source Code	3
I.3.B. Screenshot.....	5
I.3.C. Analisa	6
I.4 Program 4	6
I.4.A. Source Code	6
I.4.B. Screenshot.....	7
I.4.C. Analisa	7
I.5 Program 5	8
I.5.A. Source Code	8
I.5.B. Screenshot.....	9
I.5.C. Analisa	9
I.6 Program 6	10
I.6.A. Source Code	10

I.6.B.	Screenshot.....	10
I.6.C.	Analisa	11
I.7	Program 7	11
I.7.A.	Source Code	11
I.7.B.	Screenshot.....	11
I.7.C.	Analisa	12
I.8	Program 8	12
I.8.A.	Source Code	12
I.8.B.	Screenshot.....	13
I.8.C.	Analisa	13
I.9	Program 9	14
I.9.A.	Source Code	14
I.9.B.	Screenshot.....	14
I.9.C.	Analisa	14
I.10	Program 10	15
I.10.A.	Source Code	15
I.10.B.	Screenshot	16
I.10.C.	Analisa.....	16
I.11	Program 11	16
I.11.A.	Source Code	16
I.11.B.	Screenshot	17
I.11.C.	Analisa.....	17
I.12	Program 12	18
I.12.A.	Source Code	18
I.12.B.	Screenshot	20
I.12.C.	Analisa.....	20
I.13	Program 13	20

I.13.A.	Source Code	20
I.13.B.	Screenshot	22
I.13.C.	Analisa.....	22
I.14	Program 14	23
I.14.A.	Source Code	23
I.14.B.	Screenshot	24
I.14.C.	Analisa.....	24
I.15	Program 15	25
I.15.A.	Source Code	25
I.15.B.	Screenshot	27
I.15.C.	Analisa.....	27
BAB II.	TUGAS PRAKTIKUM	28
II.1	Tugas 1	28
II.1.A.	Source Code	28
II.1.B.	Screenshot	29
II.1.C.	Analisa.....	29
II.2	Tugas 2	29
II.2.A.	Source Code	29
II.2.B.	Screenshot	30
II.2.C.	Analisa.....	30
II.3	Tugas 3	30
II.3.A.	Source Code	30
II.3.B.	Screenshot	31
II.3.C.	Analisa.....	32
BAB III.	KESIMPULAN	33

DAFTAR GAMBAR

Gambar 1 Hasil Run dari Program 1.....	1
Gambar 2 Hasil Run dari Program 2.....	3
Gambar 3 Hasil Run dari Program 3.....	6
Gambar 4 Hasil Run dari Program 4.....	7
Gambar 5 Hasil Run dari Program 5.....	9
Gambar 6 Hasil Run dari Program 6.....	11
Gambar 7 Hasil Run dari Program 7.....	12
Gambar 8 Hasil Run dari Program 8.....	13
Gambar 9 Hasil Run dari Program 9.....	14
Gambar 10 Hasil Run dari Program 10.....	16
Gambar 11 Hasil Run dari Program 11.....	17
Gambar 12 Hasil Run dari Program 12.....	20
Gambar 13 Hasil Run dari Program 13.....	22
Gambar 14 Hasil Screenshot dari Program 14.....	24
Gambar 15 Hasil Run dari Program 15.....	27
Gambar 16 Hasil Run dari Tugas 1.....	29
Gambar 17 Hasil Run dari Tugas 2.....	30
Gambar 18 Hasil Run dari Tugas 3.....	32

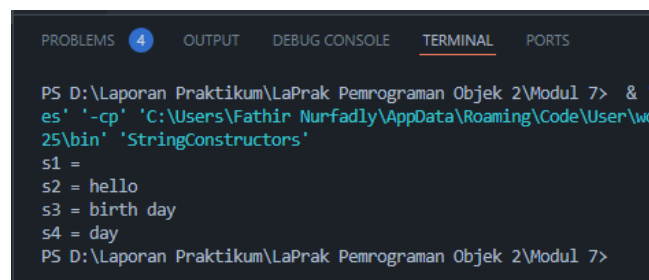
BAB I. HASIL PRAKTIKUM

I.1 Program 1

I.1.A. Source Code

```
public class StringConstructors {  
    public static void main(String[] args) {  
        char[] charArray = {'b', 'i', 'r', 't', 'h', ' ', 'd',  
'a', 'y'};  
  
        String s = new String("hello");  
  
        // menggunakan konstruktor  
        String s1 = new String();  
        String s2 = new String(s);  
        String s3 = new String(charArray);  
        String s4 = new String(charArray, 6, 3);  
  
        System.out.printf("s1 = %s\ns2 = %s\ns3 = %s\ns4 =  
%s\n", s1, s2, s3, s4);  
    }  
}
```

I.1.B. Screenshot



Gambar 1 Hasil Run dari Program 1

I.1.C. Analisa

Program di atas merupakan contoh penggunaan berbagai konstruktor dalam kelas String di Java. Pertama, sebuah array karakter 'charArray' didefinisikan dengan nilai karakter tertentu. Kemudian, objek String 's' dibuat menggunakan konstruktor dengan parameter string "hello". Selanjutnya, empat objek String lainnya 's1', 's2', 's3', dan 's4' dibuat menggunakan konstruktor tanpa parameter, konstruktor dengan parameter objek String 's', konstruktor dengan parameter array karakter 'charArray', dan konstruktor dengan parameter array karakter 'charArray' serta indeks awal dan panjangnya.

I.2 Program 2

I.2.A. Source Code

```
public class StringMiscellaneous {  
    public static void main(String[] args)  
    {  
        String s1 = "hello there";  
        char[] charArray = new char[5];  
  
        System.out.printf("s1: %s", s1);  
        System.out.printf("\nLength of s1: %d", s1.length());  
        System.out.printf("\nThe string reversed is: ");  
  
        for (int count = s1.length() - 1; count >= 0; count--)  
            System.out.printf("%c ", s1.charAt(count));  
  
        s1.getChars(0, 5, charArray, 0);  
        System.out.printf("\nThe character array is: ");
```

```

        for (char character : charArray)

            System.out.print(character);

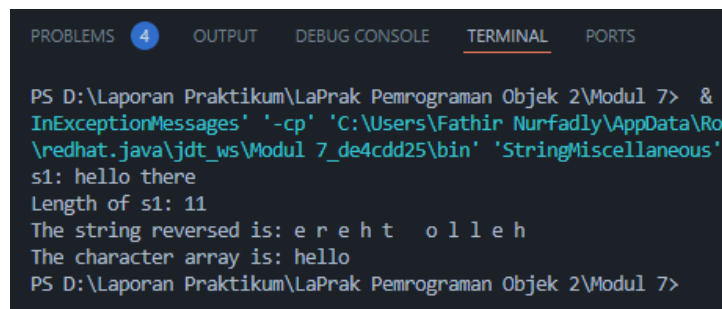
        System.out.println();

    }

}

```

I.2.B. Screenshot



```

PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7> & "C:\Program Files\Java\jdk-11.0.10\bin\java.exe" -Xmx1024m -Djava.class.path=.\StringMiscellaneous.jar -Djava.library.path=.\StringMiscellaneous\lib -cp "C:\Users\Fathir Nurfadly\AppData\Roaming\Microsoft\Windows\CurrentVersion\PathExpression\PathExpression\StringMiscellaneous.jar" StringMiscellaneous
s1: hello there
Length of s1: 11
The string reversed is: e r e h t o l l e h
The character array is: hello
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>

```

Gambar 2 Hasil Run dari Program 2

I.2.C. Analisa

Program di atas mengilustrasikan beberapa operasi yang dapat dilakukan dengan objek String di Java. Pertama, sebuah string 's1' dengan nilai "hello there" didefinisikan. Kemudian, panjang string tersebut dicetak menggunakan metode 'length()'. Selanjutnya, string 's1' dibalik dan karakter-karakternya dicetak secara terbalik menggunakan iterasi melalui indeks karakter dengan menggunakan metode 'charAt()'. Setelah itu, metode 'getChars()' digunakan untuk menyalin karakter-karakter dari string 's1' ke dalam array karakter 'charArray'. Hasilnya, karakter-karakter dalam 'charArray' dicetak.

I.3 Program 3

I.3.A. Source Code

```

public class StringCompare {

    public static void main(String[] args) {

        String s1 = new String("hello");

        String s2 = "goodbye";
    }
}

```

```

        String s3 = "Happy Birthday";

        String s4 = "happy birthday";

        System.out.printf("s1 = %s\ns2 = %s\ns3 = %s\ns4 = %s\n\n", s1, s2, s3, s4);


        if (s1.equals("hello"))
            System.out.println("s1 equals \"hello\"");
        else
            System.out.println("s1 does not equal \"hello\"");


        if (s1 == "hello")
            System.out.println("s1 is the same object as \"hello\"");
        else
            System.out.println("s1 is not the same object as \"hello\"");


        if (s3.equalsIgnoreCase(s4))
            System.out.printf("%s equals %s with case ignored\n", s3, s4);
        else
            System.out.println("s3 does not equal s4");


        System.out.printf("\ns1.compareTo(s2) is %d", s1.compareTo(s2));

        System.out.printf("\ns2.compareTo(s1) is %d", s2.compareTo(s1));

```

```
        System.out.printf("\ns1.compareTo(s1)      is      %d",
s1.compareTo(s1));

        System.out.printf("\ns3.compareTo(s4)      is      %d",
s3.compareTo(s4));

        System.out.printf("\ns4.compareTo(s3)      is      %d\n\n",
s4.compareTo(s3));


        if (s3.regionMatches(0, s4, 0, 5))

            System.out.println("First 5 characters of s3 and s4
match");

        else

            System.out.println("First 5 characters of s3 and s4
do not match");


        if (s3.regionMatches(true, 0, s4, 0, 5))

            System.out.println("First 5 characters of s3 and s4
match with case ignored");

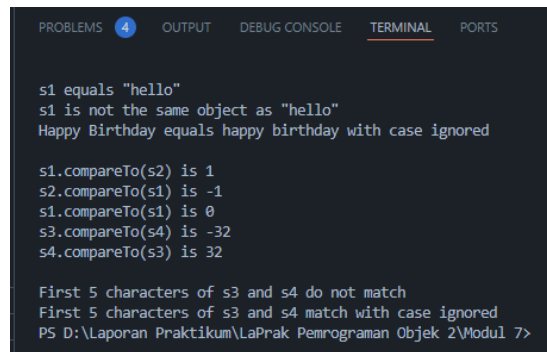
        else

            System.out.println("First 5 characters of s3 and s4
do not match");

    }

}
```

I.3.B. Screenshot



```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

s1 equals "hello"
s1 is not the same object as "hello"
Happy Birthday equals happy birthday with case ignored

s1.compareTo(s2) is 1
s2.compareTo(s1) is -1
s1.compareTo(s1) is 0
s3.compareTo(s4) is -32
s4.compareTo(s3) is 32

First 5 characters of s3 and s4 do not match
First 5 characters of s3 and s4 match with case ignored
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>
```

Gambar 3 Hasil Run dari Program 3

I.3.C. Analisa

Program di atas menunjukkan berbagai metode perbandingan string yang tersedia di Java. Pertama, beberapa objek string 's1', 's2', 's3', dan 's4' didefinisikan dengan nilai tertentu. Kemudian, program mencetak nilai-nilai dari keempat string tersebut. Selanjutnya, program menggunakan metode 'equals()' untuk memeriksa apakah 's1' sama dengan string "hello" secara leksikal, dan menggunakan operator '==' untuk memeriksa apakah 's1' mengacu pada objek yang sama dengan string "hello". Program juga menggunakan 'equalsIgnoreCase()' untuk membandingkan 's3' dan 's4' tanpa memperhatikan huruf besar kecilnya. Selain itu, metode 'compareTo()' digunakan untuk membandingkan urutan leksikal dari dua string, dan 'regionMatches()' digunakan untuk memeriksa apakah bagian tertentu dari dua string cocok, dengan atau tanpa memperhatikan huruf besar kecilnya.

I.4 Program 4

I.4.A. Source Code

```
public class StringStartEnd {

    public static void main(String[] args) {

        String[] strings={"started", "starting", "ended",
"ending"};

        for(String string : strings) {

            System.out.printf("\n%s\"starts with\"st\"\\n",
string);
```

```

    }

    System.out.println();

    for(String string : strings) {

        if(string.startsWith("art", 2)) {

            System.out.printf("\"%s\" starts with \"art\" at
position 2\n", string);

        }

        System.out.println();

        for(String stringh : strings) {

            if(stringh.endsWith("ed")){

                System.out.printf("\"%s\" ends with
\"ed\"\n", string);

            }

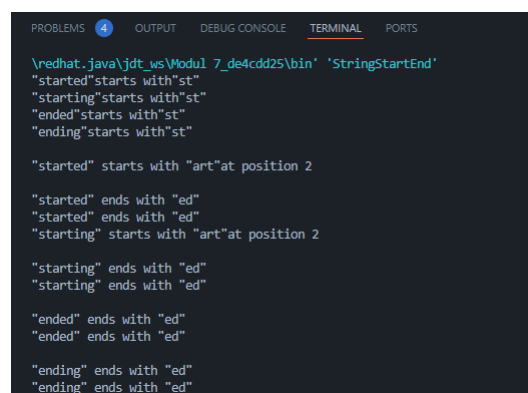
        }

    }

}

```

I.4.B. Screenshot



```

\redhat.java\jdk_ws\Modul 7_de4cdd25\bin' 'StringStartEnd'
"started" starts with "st"
"starting" starts with "st"
"ended" starts with "st"
"ending" starts with "st"

"started" starts with "art" at position 2

"started" ends with "ed"
"starting" ends with "ed"
"starting" starts with "art" at position 2

"starting" ends with "ed"
"starting" ends with "ed"

"ended" ends with "ed"
"ended" ends with "ed"

"ending" ends with "ed"
"ending" ends with "ed"

```

Gambar 4 Hasil Run dari Program 4

I.4.C. Analisa

Program di atas adalah contoh penggunaan metode 'startsWith()' dan 'endsWith()' pada objek string di Java. Pertama, array 'strings' didefinisikan dengan beberapa string. Kemudian, program mencetak setiap string dalam array yang dimulai dengan "st". Selanjutnya, program memeriksa setiap string dalam array untuk menemukan apakah string tersebut dimulai dengan "art" pada posisi kedua, jika iya, mencetak hasilnya. Kemudian, program memeriksa setiap string dalam array untuk menemukan apakah string tersebut diakhiri dengan "ed", dan mencetak hasilnya.

I.5 Program 5

I.5.A. Source Code

```
public class StringIndexMethods {  
    public static void main(String[] args) {  
        String letters = "abcdefghijklmabcdefghijklm";  
        System.out.printf("'c' is located at index %d\n",  
letters.indexOf('c'));  
  
        System.out.printf("'a' is located at index %d\n",  
letters.indexOf('a', 1));  
  
        System.out.printf("'$' is located at index %d\n\n",  
letters.indexOf('$'));  
  
        System.out.printf("Last 'c' is located at index %d\n",  
letters.lastIndexOf('c'));  
  
        System.out.printf("Last 'a' is located at index %d\n",  
letters.lastIndexOf('a', 25));  
  
        System.out.printf("Last '$' is located at index %d\n\n",  
letters.lastIndexOf('$'));
```



```

        System.out.printf("\"def\" is located at index %d\n",
letters.indexOf("def"));

        System.out.printf("\"def\" is located at index %d\n",
letters.indexOf("def", 7));

        System.out.printf("\"hello\" is located at index
%d\n\n", letters.indexOf("hello"));

        System.out.printf("Last \"def\" is located at index
%d\n", letters.lastIndexOf("def"));

        System.out.printf("Last \"def\" is located at index
%d\n", letters.lastIndexOf("def", 25));

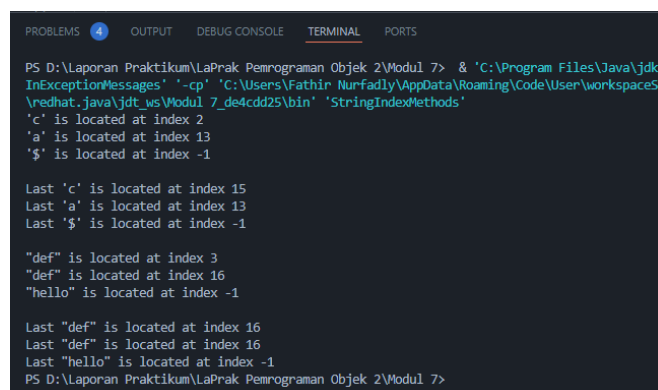
        System.out.printf("Last \"hello\" is located at index
%d\n", letters.lastIndexOf("hello"));

    }

}

```

I.5.B. Screenshot



```

PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7> & 'C:\Program Files\Java\jdk-
InExceptionMessages' '-cp' 'C:\Users\Fathir Nurfadly\AppData\Roaming\Code\User\workspaceSt
\redhat.java\jdt_ws\Modul 7_de4cdd25\bin' 'StringIndexMethods'
'c' is located at index 2
'a' is located at index 13
'$' is located at index -1

Last 'c' is located at index 15
Last 'a' is located at index 13
Last '$' is located at index -1

"def" is located at index 3
"def" is located at index 16
"hello" is located at index -1

Last "def" is located at index 16
Last "def" is located at index 16
Last "hello" is located at index -1
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>

```

Gambar 5 Hasil Run dari Program 5

I.5.C. Analisa

Program di atas menunjukkan penggunaan berbagai metode indeks string dalam Java. Pertama, string 'letters' didefinisikan dengan nilai tertentu. Kemudian, program mencetak indeks pertama dari karakter 'c' dalam string menggunakan metode 'indexOf()', serta indeks pertama dari karakter 'a' yang dimulai dari indeks 1 dan mencetak indeks karakter '\$' yang tidak

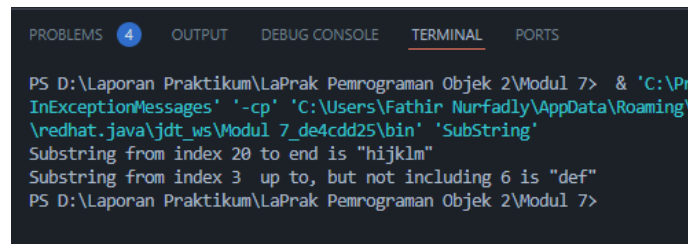
ada dalam string, yang mengembalikan -1. Selanjutnya, program mencetak indeks terakhir dari karakter 'c' dalam string menggunakan metode 'lastIndexOf()', serta indeks terakhir dari karakter 'a' yang dimulai dari indeks 25 dan mencetak indeks karakter '\$' yang tidak ada dalam string. Program juga mencetak indeks pertama dari substring "def" dalam string menggunakan 'indexOf()', serta indeks pertama dari substring "def" yang dimulai dari indeks 7 dan mencetak indeks substring "hello" yang tidak ada dalam string, yang mengembalikan -1. Terakhir, program mencetak indeks terakhir dari substring "def" dalam string menggunakan metode 'lastIndexOf()', serta indeks terakhir dari substring "def" yang dimulai dari indeks 25 dan mencetak indeks substring "hello" yang tidak ada dalam string. Program ini memberikan gambaran tentang bagaimana menggunakan metode 'indexOf()' dan 'lastIndexOf()' untuk menemukan indeks karakter atau substring dalam string.

I.6 Program 6

I.6.A. Source Code

```
public class SubString {  
    public static void main(String[] args) {  
        String letters = "abcdefghijklmabcdefghijklm";  
        System.out.printf("Substring from index 20 to end is  
\"%s\"\\n\", letters.substring(20));  
        System.out.printf(\"%s \"%s\"\\n\", \"Substring from index  
3 up to, but not including 6 is\", letters.substring(3, 6));  
    }  
}
```

I.6.B. Screenshot



```
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7> & 'C:\Program Files\Java\jdk-11.0.2\bin\java.exe' '-cp' 'C:\Users\Fathir Nurfadly\AppData\Roaming\Microsoft\Windows\CurrentVersion\Temp\Temporary Internet Files\Content.IE5\QJGK2H25\redhat.java\jdt_ws\Modul 7_de4cdd25\bin' 'SubString'
SubString
Substring from index 20 to end is "hijklm"
Substring from index 3 up to, but not including 6 is "def"
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>
```

Gambar 6 Hasil Run dari Program 6

I.6.C. Analisa

Program di atas menampilkan penggunaan metode 'substring()' dalam Java untuk mendapatkan sub-string dari string tertentu. Pertama, sebuah string 'letters' didefinisikan dengan nilai tertentu. Kemudian, program mencetak sub-string yang dimulai dari indeks 20 hingga akhir string menggunakan metode 'substring()', yang menghasilkan sub-string "ijklmabcdefghijklm". Selanjutnya, program mencetak sub-string yang dimulai dari indeks 3 hingga sebelum indeks 6 menggunakan metode 'substring()', yang menghasilkan sub-string "def".

I.7 Program 7

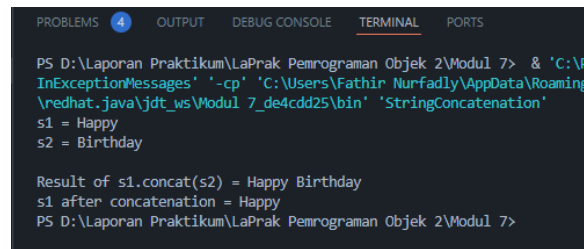
I.7.A. Source Code

```
public class StringConcatenation {
    public static void main(String[] args) {
        String s1 = "Happy ";
        String s2 = "Birthday";

        System.out.printf("s1 = %s\ns2 = %s\n\n",s1, s2);
        System.out.printf("Result of s1.concat(s2) = %s\n", s1.concat(s2));

        System.out.printf("s1 after concatenation = %s\n", s1);
    }
}
```

I.7.B. Screenshot



```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7> & 'C:\F
InExceptionMessages' '-cp' 'C:\Users\Fathir Nurfadly\AppData\Roaming
\redhat.java\jdt_ws\Modul 7_de4cdd25\bin' 'StringConcatenation'
s1 = Happy
s2 = Birthday

Result of s1.concat(s2) = Happy Birthday
s1 after concatenation = Happy
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>
```

Gambar 7 Hasil Run dari Program 7

I.7.C. Analisa

Program ini mengilustrasikan penggunaan operator konkatenasi string "+" dan metode 'concat()' untuk menggabungkan dua string dalam Java. Pertama, dua string 's1' dan 's2' didefinisikan dengan nilai "Happy " dan "Birthday". Kemudian, program mencetak nilai dari kedua string tersebut. Selanjutnya, program menggunakan operator konkatenasi "+" untuk menggabungkan kedua string tersebut dan mencetak hasilnya. Kemudian, program menggunakan metode 'concat()' untuk menggabungkan string 's1' dan 's2', dan mencetak hasilnya. Penting untuk dicatat bahwa penggunaan operator "+" membuat string baru yang berisi hasil konkatenasi, sementara metode 'concat()' menghasilkan string baru tetapi tidak memodifikasi string asli. Oleh karena itu, nilai 's1' tetap sama setelah menggunakan metode 'concat()'.

I.8 Program 8

I.8.A. Source Code

```
public class StringMiscellaneous2 {
    public static void main(String[] args) {
        String s1 = "hello";
        String s2 = "GOODBYE";
        String s3 = "    spaces    ";
        System.out.printf("s1 = %s\ns2 = %s\ns3 = %s\n\n", s1,
s2, s3);
        System.out.printf("Replace 'l' with 'L' in s1: %s\n\n",
s1.replace('l', 'L'));
    }
}
```

```

        System.out.printf("s1.toUpperCase()      =      %s\n",
s1.toUpperCase());

        System.out.printf("s2.toLowerCase()      =      %s\n",
s2.toLowerCase());

        System.out.printf("s3   after   trim   =   \"%s\"\n",
s3.trim());

        char[] charArray = s1.toCharArray();

        System.out.println("s1 as a character array = ");

        for (char character : charArray)

            System.out.print(character);

        System.out.println();

    }

}

```

I.8.B. Screenshot

```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7> & 'C:\Program Files\
InExceptionMessages' '-cp' 'C:\Users\Fathir Nurfadly\AppData\Roaming\Code\User\wo
\redhat.java\jdt_ws\Modul_7_de4cdd25\bin' 'StringMiscellaneous2'
s1 = hello
s2 = GOODBYE
s3 =   spaces

Replace 'l' with 'L' in s1: heLLo

s1.toUpperCase() = HELLO
s2.toLowerCase() = goodbye
s3 after trim = "spaces"
s1 as a character array =
hello
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>

```

Gambar 8 Hasil Run dari Program 8

I.8.C. Analisa

Program di atas menunjukkan penggunaan berbagai metode yang tersedia dalam kelas String di Java untuk melakukan operasi tertentu pada string. Pertama, tiga string 's1', 's2', dan 's3' didefinisikan dengan nilai tertentu. Kemudian, program mencetak nilai dari ketiga string tersebut. Selanjutnya, program menggunakan metode 'replace()' untuk mengganti semua kemunculan karakter 'l' dalam 's1' dengan 'L' dan mencetak hasilnya. Program juga menggunakan metode 'toUpperCase()' untuk mengubah semua karakter dalam 's1' menjadi huruf besar, serta metode 'toLowerCase()' untuk mengubah semua karakter dalam 's2' menjadi

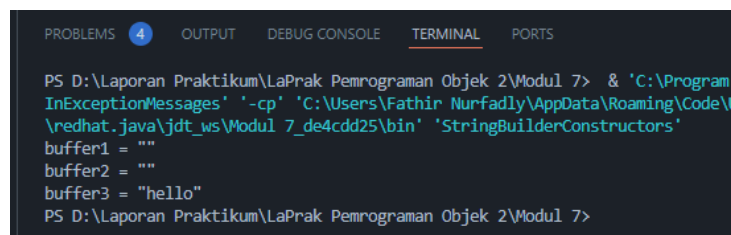
huruf kecil, dan mencetak hasilnya. Selain itu, metode 'trim()' digunakan untuk menghapus spasi di awal dan di akhir 's3', dan mencetak hasilnya. Terakhir, program menggunakan metode 'toCharArray()' untuk mengubah 's1' menjadi array karakter dan mencetak setiap karakter dalam array tersebut.

I.9 Program 9

I.9.A. Source Code

```
public class StringBuilderConstructors {  
    public static void main(String[] args) {  
        StringBuilder buffer1 = new StringBuilder();  
        StringBuilder buffer2 = new StringBuilder(10);  
        StringBuilder buffer3 = new StringBuilder("hello");  
  
        System.out.printf("buffer1 = \"%s\"\\n", buffer1);  
        System.out.printf("buffer2 = \"%s\"\\n", buffer2);  
        System.out.printf("buffer3 = \"%s\"\\n", buffer3);  
    }  
}
```

I.9.B. Screenshot



```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7> & 'C:\Program  
InExceptionMessages' '-cp' 'C:\Users\Fathir Nurfadly\AppData\Roaming\Code\U  
\redhat.java\jdt_ws\Modul 7_de4cdd25\bin' 'StringBuilderConstructors'  
buffer1 = ""  
buffer2 = ""  
buffer3 = "hello"  
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>
```

Gambar 9 Hasil Run dari Program 9

I.9.C. Analisa

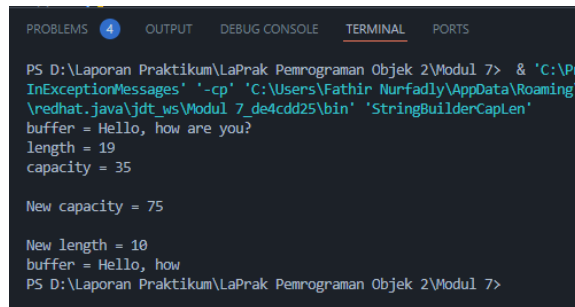
Program di atas mengilustrasikan penggunaan konstruktor-konstruktor yang tersedia dalam kelas `StringBuilder` di Java untuk membuat objek `StringBuilder` dengan cara yang berbeda. Pertama, objek `StringBuilder` 'buffer1' dibuat tanpa argumen, sehingga memiliki kapasitas awal default. Kemudian, objek `StringBuilder` 'buffer2' dibuat dengan argumen kapasitas awal 10. Terakhir, objek `StringBuilder` 'buffer3' dibuat dengan argumen string "hello", yang menginisialisasi konten `StringBuilder` dengan string tersebut dan kapasitas yang cukup untuk menampung string tersebut. Setelah itu, program mencetak nilai dari ketiga objek `StringBuilder` tersebut.

I.10 Program 10

I.10.A. Source Code

```
public class StringBuilderCapLen {  
    public static void main(String[] args) {  
        StringBuilder buffer = new StringBuilder("Hello, how are  
you?");  
  
        System.out.printf("buffer = %s\nlength = %d\ncapacity =  
%d\n\n", buffer.toString(), buffer.length(), buffer.capacity());  
  
        buffer.ensureCapacity(75);  
  
        System.out.printf("New      capacity      =      %d\n\n",  
buffer.capacity());  
  
        buffer.setLength(10);  
  
        System.out.printf("New  length  =  %d\nbuffer  =  %s\n",  
buffer.length(), buffer.toString());  
    }  
}
```

I.10.B. Screenshot



```
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7> & "C:\Pr  
InExceptionMessages" "-cp" "C:\Users\Fathir NurFadly\AppData\Roaming\  
\redhat.java\jdt_ws\Modul 7_de4cdd25\bin" "StringBuilderCapLen"  
buffer = Hello, how are you?  
length = 19  
capacity = 35  
  
New capacity = 75  
  
New length = 10  
buffer = Hello, how  
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>
```

Gambar 10 Hasil Run dari Program 10

I.10.C. Analisa

Program di atas mengilustrasikan penggunaan metode 'length()', 'capacity()', 'ensureCapacity()', dan 'setLength()' dalam kelas StringBuilder di Java untuk mengelola panjang dan kapasitas StringBuilder. Pertama, objek StringBuilder 'buffer' dibuat dengan string "Hello, how are you?". Kemudian, program mencetak nilai dari StringBuilder tersebut, yaitu panjangnya dan kapasitasnya. Selanjutnya, metode 'ensureCapacity()' digunakan untuk memastikan bahwa kapasitas StringBuilder cukup besar untuk menampung sejumlah karakter tertentu (dalam kasus ini, 75 karakter). Program mencetak kapasitas baru StringBuilder setelah penyesuaian. Terakhir, metode 'setLength()' digunakan untuk mengatur ulang panjang StringBuilder menjadi 10 karakter, dan program mencetak panjang dan isi StringBuilder setelah pengaturan ulang.

I.11 Program 11

I.11.A. Source Code

```
public class StringBuilderChars {  
    public static void main(String[] args) {  
        StringBuilder buffer = new StringBuilder("hello there");  
  
        System.out.printf("buffer = %s\n", buffer.toString());  
  
        System.out.printf("Character at 0: %s\nCharacter at 4:  
%s\n\n", buffer.charAt(0), buffer.charAt(4));  
    }  
}
```



```

char[] charArray = new char[buffer.length()];
buffer.getChars(0, buffer.length(), charArray, 0);
System.out.print("The characters are: ");
for (char character : charArray) {
    System.out.print(character);
}

buffer.setCharAt(0, 'H');
buffer.setCharAt(6, 'T');

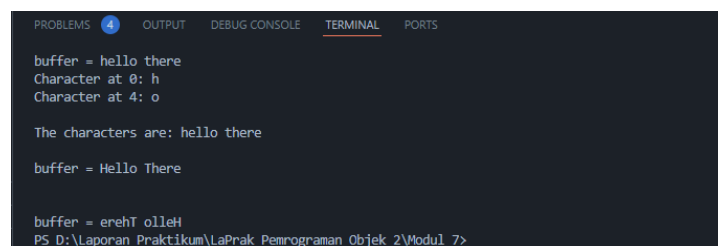
System.out.printf("\n\nbuffer          =          %s\n",
buffer.toString());

buffer.reverse();

System.out.printf("\n\nbuffer          =          %s\n",
buffer.toString());
}
}

```

I.11.B. Screenshot



```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
buffer = hello there
Character at 0: h
Character at 4: o

The characters are: hello there

buffer = Hello There

buffer = ereht olleH
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>

```

Gambar 11 Hasil Run dari Program 11

I.11.C. Analisa

Program di atas menunjukkan berbagai operasi yang dapat dilakukan pada objek `StringBuilder` di Java untuk memanipulasi karakter-karakter di dalamnya. Pertama, objek `StringBuilder` 'buffer' dibuat dengan string "hello there". Kemudian, program mencetak nilai dari `StringBuilder` tersebut, serta karakter pada indeks 0 dan 4 menggunakan metode '`charAt()`'.

Selanjutnya, sebuah array karakter 'charArray' dibuat dengan panjang yang sama dengan panjang StringBuilder, dan karakter-karakter dari StringBuilder disalin ke dalam array tersebut menggunakan metode 'getChars()'. Program kemudian mencetak karakter-karakter dalam array tersebut. Setelah itu, karakter pertama di StringBuilder diubah menjadi 'H' dan karakter keenam diubah menjadi 'T' menggunakan metode 'setCharAt()'. Program mencetak nilai StringBuilder setelah perubahan karakter. Terakhir, karakter-karakter dalam StringBuilder dibalik menggunakan metode 'reverse()', dan hasilnya dicetak.

I.12 Program 12

I.12.A. Source Code

```
public class StringBuilderAppend {  
    public static void main(String[] args) {  
        Object objectRef = "hello";  
  
        String string = "goodbye";  
  
        char[] charArray = {'a', 'b', 'c', 'd', 'e', 'f'};  
        boolean booleanValue = true;  
  
        char characterValue = 'Z';  
        int integerValue = 7;  
  
        long longValue = 10000000000L;  
        float floatValue = 2.5f;  
        double doubleValue = 33.333;  
    }  
}
```

```
StringBuilder lastBuffer = new StringBuilder("last
buffer");

StringBuilder buffer = new StringBuilder();

buffer.append(objectRef)
        .append("\n")
        .append(string)
        .append("\n")
        .append(charArray)
        .append("\n")
        .append(charArray, 0, 3)
        .append("\n")
        .append(booleanValue)
        .append("\n")
        .append(characterValue)
        .append("\n")
        .append(integerValue)
        .append("\n")
        .append(longValue)
        .append("\n")
        .append(floatValue)
        .append("\n")
        .append(doubleValue)
        .append("\n")
        .append(lastBuffer);
```

```

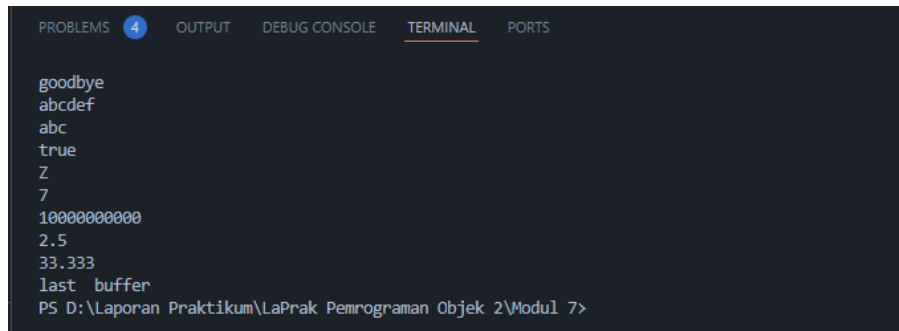
        System.out.printf("buffer                contains%n%s%n",
buffer.toString());

    }

}

```

I.12.B. Screenshot



Gambar 12 Hasil Run dari Program 12

I.12.C. Analisa

Program di atas menunjukkan penggunaan metode 'append()' dalam kelas `StringBuilder` di Java untuk menggabungkan berbagai jenis data ke dalam objek `StringBuilder`. Pertama, beberapa variabel dengan tipe data berbeda seperti string, array karakter, boolean, karakter tunggal, integer, long, float, dan double dideklarasikan dengan nilai-nilai tertentu. Kemudian, objek `StringBuilder` 'buffer' dibuat tanpa argumen. Program menggunakan metode 'append()' untuk menambahkan nilai-nilai dari variabel-variabel tersebut ke dalam `StringBuilder` 'buffer' secara berurutan, kemudian mencetak nilai `StringBuilder` tersebut. Metode 'append()' digunakan untuk menggabungkan berbagai jenis data ke dalam objek `StringBuilder`, termasuk string, array karakter, nilai boolean, karakter tunggal, dan nilai numerik.

I.13 Program 13

I.13.A. Source Code

```

public class StringBuilderInsertDelete {

    public static void main(String[] args) {

        Object objectRef = "hello";

        String string = "goodbye";
    }
}

```

```
char[] charArray = {'a', 'b', 'c', 'd', 'e', 'f'};

boolean booleanValue = true;

char characterValue = 'Z';

int integerValue = 7;

long longValue = 10000000000L;

float floatValue = 2.5f;

double doubleValue = 33.333;


StringBuilder buffer = new StringBuilder();


buffer.insert(0, objectRef);
buffer.insert(0, " ");
buffer.insert(0, string);
buffer.insert(0, " ");
buffer.insert(0, charArray);
buffer.insert(0, " ");
buffer.insert(0, charArray, 3, 3);
buffer.insert(0, " ");
buffer.insert(0, booleanValue);
buffer.insert(0, " ");
buffer.insert(0, characterValue);
buffer.insert(0, " ");
buffer.insert(0, integerValue);
buffer.insert(0, " ");
buffer.insert(0, longValue);
buffer.insert(0, " ");
```

```

        buffer.insert(0, floatValue);

        buffer.insert(0, " ");

        buffer.insert(0, doubleValue);

        System.out.printf("buffer after inserts:\n%s\n\n",
buffer.toString());

        buffer.deleteCharAt(10);

        buffer.delete(2, 6);

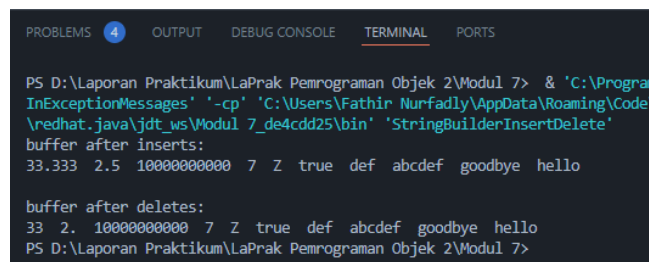
        System.out.printf("buffer after deletes:\n%s\n",
buffer.toString());

    }

}

```

I.13.B. Screenshot



```

PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7> & 'C:\Program
InExceptionMessages' '-cp' 'C:\Users\Fathir Nurfadly\AppData\Roaming\Code\
\redhat.java\jdt_ws\Modul 7_de4cdd25\bin' 'StringBuilderInsertDelete'
buffer after inserts:
33.333 2.5 10000000000 7 Z true def abcdef goodbye hello

buffer after deletes:
33 2. 10000000000 7 Z true def abcdef goodbye hello
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>

```

Gambar 13 Hasil Run dari Program 13

I.13.C. Analisa

Program di atas menunjukkan penggunaan metode 'insert()' dan 'delete()' dalam kelas StringBuilder di Java untuk menyisipkan dan menghapus karakter atau substring dari objek StringBuilder. Pertama, beberapa variabel dengan tipe data berbeda seperti string, array karakter, boolean, karakter tunggal, integer, long, float, dan double dideklarasikan dengan nilai-nilai tertentu. Kemudian, objek StringBuilder 'buffer' dibuat tanpa argumen. Program menggunakan metode 'insert()' untuk menyisipkan nilai-nilai dari variabel-variabel tersebut ke dalam StringBuilder 'buffer' pada indeks tertentu, dan menggunakan metode 'delete()' untuk menghapus karakter pada indeks tertentu atau rentang indeks. Setelah itu, program mencetak nilai StringBuilder tersebut setelah operasi penyisipan dan penghapusan. Metode 'insert()'

digunakan untuk menyisipkan nilai-nilai dari berbagai jenis data ke dalam objek `StringBuilder`, sementara metode `'delete()'` digunakan untuk menghapus karakter atau substring dari objek `StringBuilder`.

I.14 Program 14

I.14.A. Source Code

```
public class ValidateInput {

    //validasi firstName
    public static boolean validateFirstName(String firstName){
        return firstName.matches("[A-Z][a-zA-Z]*");
    }

    //validasi lastName
    public static boolean validateLastName(String lastName){
        return lastName.matches("[a-zA-z]+(['-][a-zA-Z]+)*");
    }

    //validasi alamat
    public static boolean validateAddress(String address){
        return address.matches("\\d+\\s+([a-zA-Z]+|[a-zA-Z]+\\s[a-zA-Z]+)");
    }

    //validasi kota
```

```

        public static boolean validateCity(String city) {
            return city.matches("([a-zA-Z]+|[a-zA-Z]+\s[a-zA-Z]+)");
        }

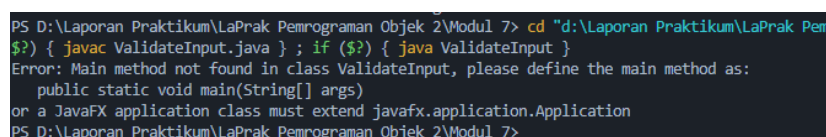
        //validasi negara
        public static boolean validateState(String state) {
            return state.matches("([a-zA-Z]+|[a-zA-Z]+\s[a-zA-Z]+)");
        }

        //validasi kodepos
        public static boolean validateZip(String zip) {
            return zip.matches("\\d{5}");
        }

        //validasi telephone
        public static boolean validatePhone(String phone) {
            return phone.matches("[1-9]\\d{2}-[1-9]\\d{2}-\\d{4}");
        }
    }
}

```

I.14.B. Screenshot



```

PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7> cd "d:\Laporan Praktikum\LaPrak Pem
$?) { javac ValidateInput.java } ; if ($?) { java ValidateInput }
Error: Main method not found in class ValidateInput, please define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>

```

Gambar 14 Hasil Screenshot dari Program 14

I.14.C. Analisa

Program ini merupakan kumpulan metode validasi input untuk data personal seperti nama depan, nama belakang, alamat, kota, negara, kode pos, dan nomor telepon. Setiap metode menggunakan ekspresi reguler untuk memeriksa apakah input sesuai dengan format yang

diharapkan. Misalnya, metode 'validateFirstName' memeriksa apakah nama depan dimulai dengan huruf kapital diikuti oleh huruf kecil atau tidak, sementara metode 'validatePhone' memeriksa apakah nomor telepon memiliki format yang benar. Dengan menggunakan metode-metode ini, pengguna dapat memastikan bahwa data yang dimasukkan sesuai dengan aturan yang ditetapkan sebelumnya.

I.15 Program 15

I.15.A. Source Code

```
import java.util.Scanner;

public class Validate {

    public static void main(String[] args) {

        //dapatkan input dari user

        Scanner scanner = new Scanner(System.in);

        System.out.println("Please enter first name:"); String
firstName = scanner.nextLine();

        System.out.println("Please enter last name:"); String
lastName = scanner.nextLine();

        System.out.println("Please enter address:");

        String address = scanner.nextLine();

        System.out.println("Please enter city:");

        String city = scanner.nextLine();

        System.out.println("Please enter state:");

        String state = scanner.nextLine();

        System.out.println("Please enter zip:");
```

```
String zip = scanner.nextLine();
System.out.println("Please enter phone:");
String phone = scanner.nextLine();
//validasi masukan user dan tampil pesan error
System.out.println("\nValidate Result:");

if (!ValidateInput.validateFirstName(firstName))
    System.out.println("Invalid first name");

else if (!ValidateInput.validateLastName(lastName))
    System.out.println("Invalid last name");

else if (!ValidateInput.validateAddress(address))
    System.out.println("Invalid address");

else if (!ValidateInput.validateCity(city))
    System.out.println("Invalid city");

else if (!ValidateInput.validateState(state))
    System.out.println("Invalid state");

else if (!ValidateInput.validateZip(zip))
    System.out.println("Invalid zip code");

else if (!ValidateInput.validatePhone(phone))
```

```

        System.out.println("Invalid phone number");

    else

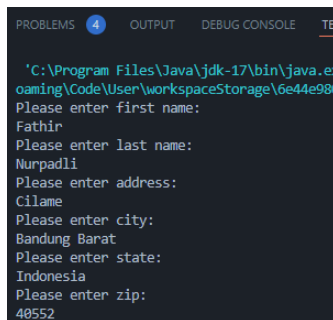
        System.out.println("Valid input. Thank you.");

    }

}

```

I.15.B. Screenshot



```

'C:\Program Files\Java\jdk-17\bin\java.exe
oaming\Code\User\workspaceStorage\6e44e988
Please enter first name:
Fathir
Please enter last name:
Nurpadli
Please enter address:
Cilame
Please enter city:
Bandung Barat
Please enter state:
Indonesia
Please enter zip:
40552

```

Gambar 15 Hasil Run dari Program 15

I.15.C. Analisa

Program di atas merupakan aplikasi sederhana yang meminta pengguna untuk memasukkan data personal seperti nama depan, nama belakang, alamat, kota, negara bagian, kode pos, dan nomor telepon menggunakan objek Scanner. Setelah pengguna memasukkan data, program memvalidasi setiap input menggunakan metode-metode validasi yang didefinisikan dalam kelas ValidateInput. Jika salah satu input tidak valid, program akan mencetak pesan kesalahan yang sesuai. Jika semua input valid, program akan mencetak "Valid input. Thank you.".

BAB II. TUGAS PRAKTIKUM

II.1 Tugas 1

II.1.A. Source Code

```
import java.util.Calendar;

import java.util.Date;

public class CompareTo {

    public static void main(String[] args) {

        Date date = new Date();

        Calendar calendar = Calendar.getInstance();

        calendar.set(2003, Calendar.OCTOBER, 18);

        Date date2 = calendar.getTime();

        int comparison = date.compareTo(date2);

        int comparison2 = date2.compareTo(date);

        int comparison3 = date.compareTo(date); // Ini akan
selalu mengembalikan 0 karena membandingkan objek dengan dirinya
sendiri.

        System.out.println("Comparison Result: " + comparison);

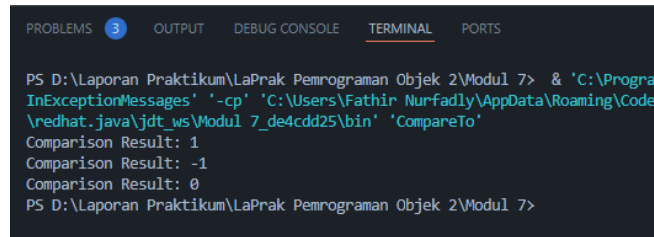
        System.out.println("Comparison      Result:      "      +
comparison2);

        System.out.println("Comparison      Result:      "      +
comparison3);

    }
}
```

```
}
```

II.1.B. Screenshot



```
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7> & 'C:\Program Files\Java\jdk-8.0.602\bin\java.exe' -cp 'C:\Users\Fathir Nurfadly\AppData\Roaming\Code\redhat.java\jdt_ws\Modul 7_de4cdd25\bin' 'CompareTo'
Comparison Result: 1
Comparison Result: -1
Comparison Result: 0
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>
```

Gambar 16 Hasil Run dari Tugas 1

II.1.C. Analisa

Program di atas membandingkan dua objek `Date` menggunakan metode `compareTo()`, yang mengembalikan nilai integer yang menunjukkan hubungan relatif antara dua tanggal. Pertama, objek `Date` saat ini diperoleh menggunakan konstruktor `Date()` yang akan merepresentasikan waktu saat ini. Kemudian, objek `Calendar` diperoleh menggunakan `Calendar.getInstance()` dan diatur tanggalnya menjadi 18 Oktober 2003 menggunakan metode `calendar.set()`. Setelah itu, objek `Date` kedua diperoleh dari `Calendar` menggunakan `calendar.getTime()`. Tiga perbandingan dilakukan antara objek-objek `Date`: `date.compareTo(date2)`, `date2.compareTo(date)`, dan `date.compareTo(date)` yang akan selalu mengembalikan 0 karena membandingkan objek dengan dirinya sendiri. Hasil perbandingan dicetak ke layar menggunakan `System.out.println()`.

II.2 Tugas 2

II.2.A. Source Code

```
import java.util.Scanner;

public class Whitespace {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.println("Masukkan String:");
```

```

        String input = in.nextLine();

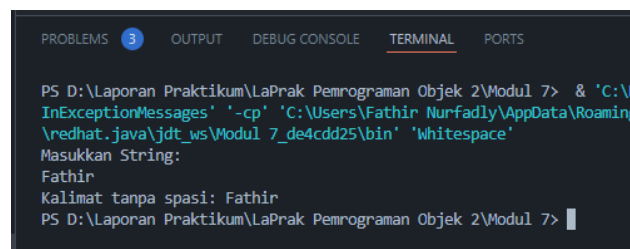
        // Menghapus semua spasi dari input menggunakan metode
replaceAll() dengan ekspresi reguler.

        String tanpaSpasi = input.replaceAll("\\s", "");

        System.out.println("Kalimat tanpa spasi: " +
tanpaSpasi);
    }
}

```

II.2.B. Screenshot



Gambar 17 Hasil Run dari Tugas 2

II.2.C. Analisa

Program di atas merupakan sebuah program sederhana yang meminta pengguna untuk memasukkan sebuah string melalui keyboard menggunakan objek Scanner. Setelah pengguna memasukkan string, program menggunakan metode `replaceAll()` untuk menghapus semua spasi dari string input menggunakan ekspresi reguler `"\s"`. Hasilnya disimpan dalam variabel `tanpaSpasi`. Akhirnya, program mencetak string tanpa spasi ke layar.

II.3 Tugas 3

II.3.A. Source Code

```

import java.util.regex.Matcher;

import java.util.regex.Pattern;

```

```

public class DuplicateWords {
    public static void main(String[] args) {
        String text = "Ini adalah sebuah tes tes untuk menemukan
kata yang duplikat duplikat.";

        // Definisikan pola regex untuk mencari kata yang
duplikat

        String regex = "\\b(\\w+)\\b.*\\b\\1\\b";

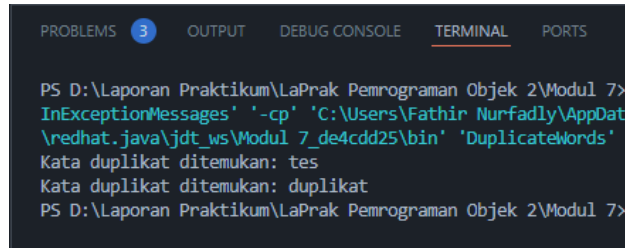
        // Buat pola regex dari string
        Pattern pattern = Pattern.compile(regex);

        // Buat objek Matcher
        Matcher matcher = pattern.matcher(text);

        // Temukan kata yang duplikat dan cetak hasilnya
        while (matcher.find()) {
            System.out.println("Kata duplikat ditemukan: " +
matcher.group(1));
        }
    }
}

```

II.3.B. Screenshot



```
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>
InExceptionHandlerMessages' -cp' 'C:\Users\Fathir Nurfadly\AppData
\redhat.java\jdt_ws\Modul 7_de4cdd25\bin' 'DuplicateWords'
Kata duplikat ditemukan: tes
Kata duplikat ditemukan: duplikat
PS D:\Laporan Praktikum\LaPrak Pemrograman Objek 2\Modul 7>
```

Gambar 18 Hasil Run dari Tugas 3

II.3.C. Analisa

Program di atas menggunakan ekspresi reguler untuk mencari kata yang duplikat dalam sebuah string. Pertama, sebuah string teks diinisialisasi dengan kalimat yang berisi kata-kata duplikat. Selanjutnya, sebuah pola regex ``\b(\w+)\b.*\b\1\b`` didefinisikan untuk mencari kata-kata yang duplikat. Pola ini mencocokkan kata-kata yang diapit oleh batas kata (``\b``) yang kemudian diikuti oleh nol atau lebih karakter (``.*``) dan kemudian diikuti lagi oleh kata yang sama yang telah ditemukan sebelumnya (``\1``). Selanjutnya, sebuah objek ``Pattern`` dibuat menggunakan pola regex tersebut. Objek ``Matcher`` kemudian dibuat dari string teks dan pola regex. Selanjutnya, program menggunakan metode ``find()`` dari objek ``Matcher`` untuk menemukan kata-kata yang cocok dengan pola regex. Setiap kali kata yang duplikat ditemukan, program mencetaknya ke layar.

BAB III. KESIMPULAN

String, karakter, dan regular expression adalah bagian yang sangat penting dalam pemrograman karena memungkinkan pengembang untuk memanipulasi dan memproses teks dengan lebih efisien dan fleksibel. String digunakan untuk merepresentasikan urutan karakter, sedangkan karakter adalah unit dasar dari teks. Penggunaan string memungkinkan pengolahan teks yang kompleks seperti pencarian, manipulasi, dan pemformatan. Regular expression merupakan alat yang kuat untuk pencarian dan penggantian pola dalam teks, memungkinkan pengguna untuk menentukan pola pencarian yang kompleks dan fleksibel. Dengan memahami konsep ini, pengembang dapat mengembangkan aplikasi yang lebih canggih dan efisien dalam pengolahan teks, pengambilan data, validasi input, dan banyak lagi.

