

LAPORAN HASIL PRAKTIKUM 1
PEMROGRAMAN BERORIENTASI OBJEK LANJUT
“ (String) ”

Dosen Pengampu :
Sri Hartati Wijono, M.Kom.



Oleh

Nama : Maria Gresia Plena Br Purba

NIM : 235314094

Kelas : DP

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA

2024

A. TUJUAN

1. ...

B. PELAKSANAAN PRAKTIKUM

1. Soal Nomor 1

a. Soal

Ch9TestChar

b. Capture Code

```
1 package Modul1;
2
3 //maria gresia_235314094
4 public class Ch9TestChar {
5
6     public static void main(String[] args) {
7         // (int) 'X' mengubah karakter menjadi kode ASCII
8         System.out.println("Kode ASCII dari karakter X : " + (int) 'X');
9
10        // (char) mengubah angka 88 menjadi karakter
11        System.out.println("Karakter dengan kode ASCII 88 : " + (char) 88);
12
13        // memeriksa apakah String huruf kapital atau tidak
14        if (Character.isUpperCase(ch: 'a')) {
15            System.out.println("Huruf Besar");
16        } else {
17            // mengubah String menjadi huruf kapital dengan menggunakan toUpperCase
18            System.out.println("Huruf besarnya a adalah : " + Character.toUpperCase(ch: 'a'));
19        }
20
21        char angka = '8';
22        // mengubah karakter menjadi digit angka dengan menggunakan isDigit
23        if (Character.isDigit(ch: angka)) {
24            System.out.println("Angka " + angka);
25        }
26
27        char huruf = 'z';
28        // mengubah karakter menjadi huruf dengan menggunakan isLetter
29        if (Character.isLetter(ch: huruf)) {
30            System.out.println("Huruf " + huruf);
31        }
32    }
33 }
34 }
```

c. Capture Output

```
run:
Kode ASCII dari karakter X : 88
Karakter dengan kode ASCII 88 : X
Huruf besarnya a adalah : A
Angka 8
Huruf z
BUILD SUCCESSFUL (total time: 0 seconds)
```

d. Analisis

Mengubah karakter menjadi code ASCII dengan menggunakan “int”. Kemudian, mengubah suatu kode ASCII dengan menggunakan “char”. Terdapat perulangan if-else yang pertama memeriksa apakah ‘a’ huruf kapital atau tidak dengan menggunakan isUpperCase() pada Character. Jika tidak, maka akan menggunakan toUpperCase() pada Character untuk mengubah ‘a’ menjadi ‘A’. Kemudian, mengubah karakter menjadi angka dengan menggunakan isDigit() dan mengubah karakter menjadi huruf dengan menggunakan isLetter().

2. Soal Nomor 2

a. Soal

Ch9TestString

b. Capture Code

```
1 package Modul1;
2
3 //maria gresia_235314094
4 public class Ch9TestString {
5
6     public static void main(String[] args) {
7         String message = "Welcome to Java!";
8         String pesan = new String(original:"Welcome to this class!");
9         String s = new String(); //membuat objek String
10        s = "Welcome"; //mengisi objek s dengan kata "Welcome"
11
12        //menampilkan isi dari message, pesan, dan s
13        System.out.println(x: message);
14        System.out.println(x: pesan);
15        System.out.println(x: s);
16
17        //menampilkan karakter ke-4 dari s pada index ke-3
18        System.out.println("Huruf ke-3 dari string : " + s.charAt(index: 3));
19
20        //menampilkan huruf ke-5 hingga index ke-7 dengan menggunakan subString pada message
21        System.out.println("Huruf ke-5 hingga 7 dari string adalah : " + message.substring(beginIndex: 4, endIndex: 7));
22    }
23 }
```

c. Capture Output

```
run:
Welcome to Java!
Welcome to this class!
Welcome
Huruf ke-3 dari string : c
Huruf ke-5 hingga 7 dari string adalah : ome
BUILD SUCCESSFUL (total time: 0 seconds)
```

d. Analisis

Menampilkan isi dari message, pesan, dan s. kemudian menampilkan huruf ke-3 dari s dengan menggunakan charAt untuk menentukan index nya. Lalu, menampilkan huruf ke-5 hingga ke-7 dengan menggunakan substring pada message untuk menentukan index awal dan akhir.

3. Soal Nomor 3

a. Soal

Ch9CountVowels

b. Capture Code

```
5 //maria gresia_235314094
6 public class Ch9CountVowels {
7
8     public static void main(String[] args) {
9         Scanner scanner = new Scanner(System.in);
10        String name;
11        int numberOfCharacters, vowelCount = 0;
12        char letter;
13
14        System.out.print("What is your name? "); //meminta input
15        name = scanner.nextLine(); //menyimpan input ke dalam "name"
16
17        numberOfCharacters = name.length(); //menyimpan panjang dari nama ke dalam numberOfCharacters
18        System.out.println("name: " + name); //menampilkan isi dari "name"
19
20        for (int i = 0; i < numberOfCharacters; i++) { //memeriksa setiap karakter dalam "name"
21            letter = name.charAt(i); //mengambil karakter ke-i
22
23            //memeriksa apakah karakter tersebut merupakan huruf vokal atau tidak
24            if (letter == 'a' || letter == 'A'
25                || letter == 'e' || letter == 'E'
26                || letter == 'i' || letter == 'I'
27                || letter == 'o' || letter == 'O'
28                || letter == 'u' || letter == 'U') {
29                vowelCount++; //menjumlahkan setiap huruf vokal yang ada pada data name jika memenuhi syarat
30            }
31        }
32        System.out.println(name + ", your name has " + vowelCount + " vowels"); //menampilkan jumlah dari huruf vokal
33    }
34 }
```

c. Capture Output

```
run:
What is your name? maria grESIA pleNA BR PURBA
maria grESIA pleNA BR PURBA
maria grESIA pleNA BR PURBA, your name has 10 vowels
BUILD SUCCESSFUL (total time: 12 seconds)
```

d. Analisis

Meminta user untuk menginput nama kemudian menyimpannya ke dalam variabel `name`. Kemudian, menyimpan panjang dari nama ke dalam `numberOfCharacters` dan menampilkan isi dari variabel `"name"`. Terdapat perulangan `for` yang akan memeriksa setiap karakter dalam `"name"` untuk mengetahui apakah ada huruf vocal. Jika ada, maka huruf vocal tersebut akan dijumlahkan dan banyaknya huruf vocal tersebut akan ditampilkan.

4. Soal Nomor 4

a. Soal

Ch9CountVowels2

b. Capture Code

```
1 package Modul1;
2
3 import java.util.*;
4
5 //maria gresia_235314094
6 public class Ch9CountVowels2 {
7
8     public static void main(String[] args) {
9         Scanner scanner = new Scanner(System.in);
10        String name, nameUpper;
11        int numberOfCharacters, vowelCount = 0;
12        char letter;
13
14        System.out.print("What is your name?"); //meminta input
15        name = scanner.nextLine(); //menyimpan input ke dalam "name"
16
17        numberOfCharacters = name.length(); //menyimpan panjang dari nama ke dalam "numberOfCharacters"
18        nameUpper = name.toUpperCase(); //toUpperCase() digunakan untuk mengubah data menjadi kapital
19
20        for (int i = 0; i < numberOfCharacters; i++) { //memeriksa setiap karakter dalam "nameUpper"
21            letter = nameUpper.charAt(i); //mengambil karakter ke-i yang telah diubah menjadi huruf kapital
22            if (letter == 'A' //memeriksa apakah termasuk AIUEO
23                || letter == 'E'
24                || letter == 'I'
25                || letter == 'O'
26                || letter == 'U') {
27                vowelCount++; //menjumlahkan setiap huruf vokal yang ada pada data name jika memenuhi syarat
28            }
29        }
30        System.out.println(name + ", your name has " + vowelCount + " vowels"); //menampilkan jumlah dari huruf vokal
31    }
32 }
```

c. Capture Output

```
run:
What is your name?maria GresIA plena br puRBa
maria GresIA plena br puRBa, your name has 10 vowels
BUILD SUCCESSFUL (total time: 16 seconds)
```

d. Analisis

Meminta user untuk menginput nama dan menyimpannya dalam “name”. Kemudian, menyimpan panjang dari nama ke dalam numberOfCharacters dilanjutkan dengan mengubah huruf kecil menjadi huruf kapital dengan menggunakan toUpperCase() yang kemudian hasilnya akan disimpan kedalam variabel nameUpper.lalu, terdapat perulangan for yang berfungsi untuk memeriksa setiap karakter ke-i yang telah diubah menjadi huruf kapital semua apakah ada yang termasuk huruf vocal. Jika ada, maka akan dijumlahkan dan akan ditampilkan.

5. Soal Nomor 5

a. Soal

Ch9CountJava

b. Capture Code

```
1 package Modul1;
2
3 import java.util.*;
4
5 //maria gresia_235314094
6 public class Ch9CountJava {
7
8     public static void main(String[] args) {
9         Scanner scanner = new Scanner(System.in);
10
11         int javaCount = 0;
12         String word;
13
14         while (true) { //melakukan perulangan hingga user menginput "STOP"
15             System.out.print("Next word: "); //meminta input
16             word = scanner.next(); //menyimpan input ke dalam word
17
18             if (word.equals("STOP")) { //perulangan berhenti jika input sama dengan "STOP"
19                 break;
20             } else if (word.equalsIgnoreCase("Java")) { //jika input sama dengan java tanpa memperhatikan kapital
21                 javaCount++; //menjumlahkan setiap kata "Java" yang telah diinput
22             }
23         }
24         System.out.println("'Java' count: " + javaCount); //menampilkan jumlah kata "Java"
25     }
26 }
```

c. Capture Output

```
run:
Next word: java
Next word: wfejfb
Next word: Java
Next word: JAva
Next word: STOP
'Java' count: 3
BUILD SUCCESSFUL (total time: 22 seconds)
```

d. Analisis

Terdapat perulangan while yang akan terus berjalan. User diminta untuk menginput kata yang akan disimpan ke dalam variabel “word”. Terdapat if-else, jika user menginput “STOP” maka perulangan while akan berhenti. Jika tidak, maka program akan menjumlahkan berapa banyak inputan berupa kata “Java” dengan menghiraukan kapitalisasi huruf. Lalu, jumlah kata tersebut akan ditampilkan.

6. Soal Nomor 6

a. Soal

Ch9CountWords

b. Capture Code

```
1 package Modul1;
2
3 import java.util.*;
4 import javax.swing.JOptionPane;
5
6 //maria gresia_235314094
7 public class Ch9CountWords {
8
9     private static final char BLANK = ' ';
10
11     public static void main(String[] args) {
12         int index, wordCount, numberOfCharacters;
13
14         //meminta input yang akan ditampilkan dengan menggunakan JOptionPane.showInputDialog
15         String sentence = JOptionPane.showInputDialog(message: "Enter a sentence: ");
16         //menyimpan jumlah karakter "sentence" ke dalam numberOfCharacters
17         numberOfCharacters = sentence.length();
18
19         index = 0;
20         wordCount = 0;
21
22         //melakukan perulangan selama index kurang dari numberOfCharacters
23         while (index < numberOfCharacters) {
24             //jika karakter di index pada "sentence" adalah blank
25             while (index < numberOfCharacters && sentence.charAt(index) == BLANK) {
26                 index++; //maka akan menambahkan nilai index
27             }
28             //jika karakter di index pada "sentence" bukan blank
29             while (index < numberOfCharacters && sentence.charAt(index) != BLANK) {
30                 index++; //maka akan menambahkan nilai index
31             }
32             wordCount++; //menjumlahkan banyaknya kata yang ada
33         }
34         System.out.println("\nInput sentence: " + sentence); //menampilkan isi dari variabel sentence
35         System.out.println("    Word count: " + wordCount + " words"); //menampilkan jumlah dari banyak kata yang ada
36     }
37 }
```

c. Capture Output

run:

Input sentence: maria gresia

Word count: 2 words

BUILD SUCCESSFUL (total time: 5 seconds)

d. Analisis

Meminta user untuk menginput suatu kata yang kemudian akan disimpan ke dalam "sentence". Lalu, panjang dari "sentence" akan disimpan ke dalam "numberOfCharacters". Terdapat perulangan while selama index kurang dari "numberOfCharacters". Terdapat perulangan while di mana jika karakter di index pada "sentence" adalah blank, maka nilai dari index akan bertambah. Sedangkan, jika karakter di index pada "sentence" tidak blank, maka nilai dari index akan bertambah. Kemudian, menambahkan banyak kata yang disimpan ke dalam wordCount. Lalu, menampilkan "sentence" dan "wordCount".

7. Soal Nomor 7

a. Soal

Ch9ExtractWords

b. Capture Code

```
1 package Modul1;
2
3 import java.util.*;
4
5 //maria gresia_235314094
6 public class Ch9ExtractWords {
7
8     private static final char BLANK = ' ';
9
10    public static void main(String[] args) {
11        Scanner scanner = new Scanner(System.in);
12
13        int index, numberOfCharacters, beginIdx, endIdx;
14        String word, sentence;
15
16        System.out.print(s: "Input: "); //meminta input
17        sentence = scanner.nextLine(); //menyimpan input ke dalam "sentence"
18
19        //menyimpan jumlah karakter sentence ke dalam "numberOfCharacters"
20        numberOfCharacters = sentence.length();
21        index = 0;
22
23        //melakukan perulangan selama index kurang dari numberOfCharacters
24        while (index < numberOfCharacters) {
25            //jika karakter di index pada "sentence" adalah blank
26            while (index < numberOfCharacters && sentence.charAt(index) == BLANK) {
27                index++; //maka akan menambahkan nilai index
28            }
29            beginIdx = index; //menyimpan indeks awal kata
30        }
```



```

31 //jika karakter di index pada "sentence" bukan blank
32 while (index < numberOfCharacters && sentence.charAt(index) != BLANK) {
33     index++; //maka akan menambahkan nilai index
34 }
35
36 endIndex = index; //menyimpan indeks akhir kata
37 //jika beginIdx tidak sama dengan endIndex
38 if (beginIdx != endIndex) {
39     //mengambil kata pada sentence dari beginIdx hingga endIndex
40     word = sentence.substring(beginIdx, endIndex);
41     System.out.println(x: word); //menampilkan isi dari "word"
42 }
43 }
44 }
45 }

```

c. Capture Output

```

run:
Input: maria GRESIA plena
maria
GRESIA
plena
BUILD SUCCESSFUL (total time: 6 seconds)

```

d. Analisis

Meminta user untuk memberi inputan dan menyimpannya ke dalam “sentence”. Lalu, panjang dari “sentence” akan disimpan ke dalam “numberOfCharacters”. Terdapat perulangan while selama index kurang dari “numberOfCharacters”. Terdapat perulangan while di mana jika karakter di index pada “sentence” adalah blank, maka nilai dari index akan bertambah. Lalu, menyimpan index awal kata ke dalam “beginIdx”. Sedangkan, jika karakter di index pada “sentence” tidak blank, maka nilai dari index akan bertambah. Kemudian, menambahkan banyak kata yang disimpan ke dalam wordCount. Lalu, menyimpan index awal kata ke dalam “endIdx”. Kemudian, jika beginIdx tidak sama dengan endIndex maka akan mengambil kata dari posisi beginIdx hingga endIndex yang disimpan ke dalam “word”. Lalu, menampilkan hasil dari “word”.

8. Soal Nomor 8

a. Soal

Ch9MatchJavaIdentifier

b. Capture Code

```
1 package Modul1;
2
3 import java.util.*;
4
5 //maria gresia_235314094
6 public class Ch9MatchJavaIdentifier {
7
8     private static final String STOP = "STOP";
9     private static final String VALID = "Valid Java identifier";
10    private static final String INVALID = "Not a valid Java identifier";
11    private static final String VALID_IDENTIFIER_PATTERN = "[a-zA-Z][a-zA-Z0-9_]*"; //harus diawali dengan huruf
12
13    public static void main(String[] args) {
14        Scanner scanner = new Scanner(System.in);
15        String str, reply;
16
17        while (true) {
18            System.out.print("Identifier: "); //meminta input
19            str = scanner.next(); //menyimpan input ke dalam "str"
20
21            //jika input sama dengan STOP maka perulangan akan berhenti
22            if (str.equals(STOP)) {
23                break;
24            }
25
26            //harus diawali dengan huruf, lalu dilanjut dengan huruf, angka, _, atau $
27            if (str.matches("[a-zA-Z][a-zA-Z0-9_]*")) {
28                reply = VALID; //jika sesuai, maka akan menampilkan pesan pada VALID
29            } else {
30                reply = INVALID; //jika tidak sesuai, maka akan menampilkan pesan pada INVALID
31            }
32            System.out.println(str + ": " + reply + "\n"); //menampilkan pesan pada reply
33        }
34    }
35 }
```

c. Capture Output

```
run:
Identifier: java
java: Valid Java identifier

Identifier: efj23
efj23: Valid Java identifier

Identifier: led
led: Not a valid Java identifier

Identifier: STOP
BUILD SUCCESSFUL (total time: 47 seconds)
```

d. Analisis

Pada perulangan while, user diminta untuk menginputkan identifier dan data tersebut akan disimpan ke dalam variabel str. Jika input sama dengan STOP maka perulangan berhenti. Jika "str" sesuai dengan pola yang diinginkan, yaitu diawali dengan huruf lalu dapat dilanjutkan dengan huruf, angka, _, atau \$. Maka, nilai dari reply sama dengan valid. Jika tidak, maka, nilai dari reply sama dengan invalid. Lalu nilai dari str dan valid akan ditampilkan.

9. Soal Nomor 9

a. Soal

Ch9MatchPhoneNumber

b. Capture Code

```
1 package Modul1;
2
3 import java.util.*;
4
5 //maria gresia_235314094
6 public class Ch9MatchPhoneNumber {
7
8     private static final String STOP = "0";
9     private static final String VALID = "Valid phone number";
10    private static final String INVALID = "Not a valid phone number";
11    private static final String VALID_PHONE_PATTERN = "[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9]";
12
13    public static void main(String[] args) {
14        Scanner scanner = new Scanner(System.in);
15        String phoneStr, reply;
16
17        while (true) {
18            System.out.print("Phone#: "); //meminta input
19            phoneStr = scanner.next(); //menyimpan input ke dalam phoneStr
20
21            //jika user menginput "STOP" maka perulangan akan berhenti
22            if (phoneStr.equals(anObject: STOP)) {
23                break;
24            }
25            //memeriksa apakah "phoneStr" cocok dengan "VALID_PHONE_PATTERN"
26            if (phoneStr.matches(regex: VALID_PHONE_PATTERN)) {
27                reply = VALID; //jika sesuai, maka akan mencetak pesan pada valid
28            } else {
29                reply = INVALID; //jika tidak sesuai, maka akan mencetak pesan pada invalid
30            }
31            //menampilkan no pada "phoneStr" dan "reply"
32            System.out.println(phoneStr + ": " + reply + "\n");
33        }
34    }
35 }
```

c. Capture Output

```
run:
Phone#: 083840554803
083840554803: Not a valid phone number

Phone#: 8384-405-5480
8384-405-5480: Valid phone number
```

d. Analisis

Pada perulangan while, user diminta untuk menginputkan no hp dan data tersebut akan disimpan ke dalam variabel phoneStr. Jika user menginput “STOP” maka perulangan akan berhenti. Jika data dari phoneStr sesuai dengan pola yang diinginkan, yaitu 4 angka-3 angka-4 angka maka akan menampilkan pesan pada “VALID”. Jika tidak, maka akan menampilkan pesan pada “INVALID”. Lalu, diakhiri dengan menampilkan data no hp dengan pesan tersebut.

10. Soal Nomor 10

a. Soal

Ch9ReplaceAll

b. Capture Code

```
1 package Modul1;
2
3 import java.util.Scanner;
4
5 //maria gresia_235314094
6 public class Ch9ReplaceAll {
7
8     public static void main(String args[]) {
9         Scanner scanner = new Scanner(System.in);
10
11         System.out.print(s: "What is your name? "); //meminta input
12         String kalimat = scanner.nextLine(); //menyimpan input ke dalam "kalimat"
13         String diganti = kalimat.replaceAll(regex: "[aeiou]", replacement: "@"); //mengganti setiap huruf vokal dengan "@"
14
15         System.out.println("original string: " + kalimat); //menampilkan input "kalimat"
16         System.out.println("replaced string: " + diganti); //menampilkan string setelah huruf vokal diganti dengan "@"
17
18         String replaced = diganti.replaceAll(regex: "[0-9]", replacement: "*"); //mengganti angka dengan "*"
19         System.out.println("replaced number: " + replaced); //menampilkan kstring setelah angka diganti dengan "*"
20
21         String changed = diganti.replaceAll(regex: "\\s+", replacement: ""); //menghapus spasi
22         System.out.println("replaced white space: " + changed); //menampilkan string setelah spasi dihapus
23     }
24 }
```

c. Capture Output

```
run:
What is your name? maria Gresia PlenA br PurBA
original string: maria Gresia PlenA br PurBA
replaced string: m@r@@ Gr@s@@ Pl@nA br P@rBA
replaced number: m@r@@ Gr@s@@ Pl@nA br P@rBA
replaced white space: m@r@@Gr@s@@Pl@nAbrP@rBA
BUILD SUCCESSFUL (total time: 16 seconds)
```

d. Analisis

Meminta user untuk menginput nama yang akan disimpan ke dalam variabel “kalimat”. Kemudian mengganti setiap huruf voklat dengan “@” yang disimpan ke dalam variabel “diganti”. Dilanjut dengan menampilkan isi dari “kalimat” dan “diganti”. Lalu, menggunakan replaceAll() pada “diganti” untuk mengubah angka menjadi “*” yang kemudian ditampilkan setelah disimpan ke dalam variabel “replaced”. Menghapus semua spasi yang ada pada “diganti” dan menampilkannya setelah disimpan ke dalam variabel “changed”.

11. Soal Nomor 11

a. Soal

Ch9MatchJavaIdentifier2

b. Capture Code

```
1 package Modul1;
2
3 import java.util.Scanner;
4 import java.util.regex.Matcher;
5 import java.util.regex.Pattern;
6
7 //maria gresia_235314094
8 public class Ch9MatchJavaIdentifier2 {
9
10     private static final String STOP = "STOP";
11     private static final String VALID = "Valid Java identifier";
12     private static final String INVALID = "Not a valid Java identifier";
13     //pola diawali dengan huruf lalu dilanjut dengan huruf, angka, _ atau $
14     private static final String VALID_IDENTIFIER_PATTERN = "[a-zA-Z][a-zA-Z0-9_$]*";
15
16     public static void main(String[] args) {
17         Scanner scanner = new Scanner(System.in);
18         String str, reply;
19         Matcher matcher;
20
21         //mencocokkan input dengan pola pada "VALID_IDENTIFIER_PATTERN"
22         Pattern pattern = Pattern.compile(regex: VALID_IDENTIFIER_PATTERN);
23
24         while (true) {
25             System.out.print(s: "Identifier: "); //meminta input
26             str = scanner.next(); //menyimpan input ke dalam "str"
27
28             //jika input sama dengan STOP maka perulangan akan berhenti
29             if (str.equals(anObject: STOP)) {
30                 break;
31             }
32
33             //membuat objek untuk mencocokkan "str" dengan pola
34             matcher = pattern.matcher(input: str);
35
36             if (matcher.matches()) {
37                 reply = VALID; //jika sesuai, maka akan menampilkan pesan pada VALID
38             } else {
39                 reply = INVALID; //jika tidak sesuai, maka akan menampilkan pesan pada INVALID
40             }
41             //menampilkan hasil dari "str" dan "reply"
42             System.out.println(str + ": " + reply + "\n");
43         }
44     }
45 }
```

c. Capture Output

```
run:
Identifier: adhew3224
adhew3224: Valid Java identifier

Identifier: 23hdwf
23hdwf: Not a valid Java identifier

Identifier: STOP
BUILD SUCCESSFUL (total time: 28 seconds)
```

d. Analisis

Pada perulangan while, user diminta untuk menginputkan identifier dan data tersebut akan disimpan ke dalam variabel str. Jika input sama dengan STOP maka perulangan berhenti. Terdapat penggunaan pattern dan matcher untuk mencocokkan nilai dari str dengan polanya. Jika sesuai, maka akan menampilkan pesan pada “valid”. Jika tidak, maka akan menampilkan pesan pada “invalid”

12. Soal Nomor 12

a. Soal

Ch9PMCountJava

b. Capture Code

```
1 package Modul1;
2
3 import java.util.*;
4 import java.util.regex.*;
5
6 //maria gresia_235314094
7 public class Ch9PMCountJava {
8
9     public static void main(String[] args) {
10         String document = "Sistem manajemen belajar online di Universitas Sanata Dharma. "
11             + "\nmemiliki keamanan dan kehandalan yang sangat tinggi. "
12             + "\nAdministrator LMS ini dapat diakses dengan username: admin "
13             + "\ndan password: rahasia. Diharapkan semua staff agar merahasiakan "
14             + "\nusername dan password ini agar sistem tetap aman sentosa";
15         int wCount;
16         Matcher matcher;
17
18         //mencari berapa kali "username" muncul dengan menggunakan CASE_INSENSITIVE
19         //agar menghiraukan huruf kapital atau tidak
20         Pattern pattern = Pattern.compile(regex: "username", flags: Pattern.CASE_INSENSITIVE);
21         System.out.print("Sentence: " + document + "\n"); //menampilkan isi dari "document"
22
23         wCount = 0;
24         //membuat objek untuk memeriksa pola yang sama dengan "document"
25         matcher = pattern.matcher(input: document);
26
27         while (matcher.find()) { //mencari dan menghitung berapa kali pola muncul
28             wCount++; //menjumlahkan pola yang ditemukan
29         }
30         //menampilkan berapa banyak kata tersebut muncul
31         System.out.println("The word 'username' occurred " + wCount + " times.");
32     }
33 }
```

c. Capture Output

```
run:
Sentence: Sistem manajemen belajar online di Universitas Sanata Dharma
memiliki keamanan dan kehandalan yang sangat tinggi.
Administrator LMS ini dapat diakses dengan username: admin
dan password: rahasia. Diharapkan semua staff agar merahasiakan
username dan password ini agar sistem tetap aman sentosa
The word 'username' occurred 2 times.
BUILD SUCCESSFUL (total time: 0 seconds)
```

d. Analisis

Menyimpan teks ke dalam variabel “document” yang bertipe string. Kemudian mencari berapa banyak “username” muncul dalam teks tersebut dengan menggunakan compile. Kemudian, membuat objek untuk memeriksa pola yang sama dengan “document”. Lalu, mencari dan menghitung berapa kali pola itu muncul dan menjumlahkannya. Hasil penjumlahan tersebut akan ditampilkan dengan memanggil “wCount”.

13. Soal Nomor 13

a. Soal

Ch9PMCountJava2

b. Capture Code

```
1 package Modul1;
2
3 import java.util.regex.Matcher;
4 import java.util.regex.Pattern;
5
6 //maria gresia_235314094
7 public class Ch9PMCountJava2 {
8
9     public static void main(String[] args) {
10         String sentence, pattern;
11
12         sentence = "Sistem manajemen belajar online di Universitas Sanata Dharma "
13             + "memiliki keamanan dan kehandalan yang sangat tinggi. "
14             + "Administrator LMS ini dapat diakses dengan username: admin "
15             + "dan password: rahasia. Diharapkan semua staff agar merahasiakan "
16             + "username dan password ini agar sistem tetap aman sentosa";
17
18         //membuat pola untuk username dan password pada objek p1 dan p2
19         Pattern p1 = Pattern.compile(regex: ".username:\\s(\\w+)");
20         Pattern p2 = Pattern.compile(regex: ".password:\\s(\\w+)");
21
22         //membuat objek untuk memeriksa kesesuaian pola pada sentence
23         Matcher matcher1 = p1.matcher(input: sentence);
24         Matcher matcher2 = p2.matcher(input: sentence);
25
26         //mencari kecocokan pada p1 dalam sentence
27         while (matcher1.find()) {
28             //menampilkan yang sesuai dengan pola p1 dari objek matcher1
29             System.out.println(x: matcher1.group(group: 0));
30             matcher2.find();//mencari kecocokan pada p2 dalam sentence
31             //menampilkan yang sesuai dengan pola p2 dari objek matcher2
32             System.out.println(x: matcher2.group(group: 0));
33         }
34     }
35 }
```

c. Capture Output

```
run:
  username: admin
  password: rahasia
BUILD SUCCESSFUL (total time: 0 seconds)
```

d. Analisis

Menyimpan teks ke dalam variabel “sentence” yang bertipe string. Kemudian, membuat pola untuk username pada p1 dan password pada p2. Kemudian, membuat objek yang bernama matcher1 untuk p1 dan matcher2 untuk p2 untuk memeriksa kesesuaian pada “sentence”. Lalu, terdapat perulangan while yang digunakan untuk mencari kecocokan pada p1 dan p2 dengan menggunakan find() pada matcher1 dan matcher2. Kemudian, menampilkan hasilnya.

14. Soal Nomor 14

a. Soal

StringBufferInsert

b. Capture Code

```
1 package Modul1;
2
3 //maria gresia_235314094
4 public class StringBufferInsert {
5
6     public static void main(String[] args) {
7         Object objectRef = "hello";
8         String string = "goodbye";
9         char charArray[] = {'a', 'b', 'c', 'd', 'e', 'f'};
10        boolean booleanValue = true;
11        char characterValue = 'K';
12        int integerValue = 7;
13        long longValue = 10000000;
14        float floatValue = 2.5f;
15        double doubleValue = 33.333;
16
17        StringBuffer buffer = new StringBuffer();
18        //menyisipkan nilai dari objectRef
19        buffer.insert(offset: 0, obj:objectRef);
20        buffer.insert(offset: 0, str: " ");
21        //menyisipkan nilai dari string
22        buffer.insert(offset: 0, str:string);
23        buffer.insert(offset: 0, str: " ");
24        //menyisipkan nilai dari charArray
25        buffer.insert(offset: 0, str:charArray);
26        buffer.insert(offset: 0, str: " ");
27        //menyisipkan nilai dari charArray yang dimulai dari posisi ke-3 dan panjangnya 3 karakter
28        buffer.insert(index: 0, str:charArray, offset: 3, len:3);
29        buffer.insert(offset: 0, str: " ");
```



```

30 //menyisipkan nilai dari booleanValue
31 buffer.insert(offset: 0, b: booleanValue);
32 buffer.insert(offset: 0, str: " ");
33 //menyisipkan nilai dari characterValue
34 buffer.insert(offset: 0, c: characterValue);
35 buffer.insert(offset: 0, str: " ");
36 //menyisipkan nilai dari integerValue
37 buffer.insert(offset: 0, i: integerValue);
38 buffer.insert(offset: 0, str: " ");
39 //menyisipkan nilai dari longValue
40 buffer.insert(offset: 0, l: longValue);
41 buffer.insert(offset: 0, str: " ");
42 //menyisipkan nilai dari floatValue
43 buffer.insert(offset: 0, f: floatValue);
44 buffer.insert(offset: 0, str: " ");
45 //menyisipkan nilai dari doubleValue
46 buffer.insert(offset: 0, d: doubleValue);
47
48 System.out.printf(format: "buffer after inserts:\n%s\n\n", args: buffer.toString());
49 buffer.deleteCharAt(index: 10); //menghapus karakter pada index ke-10
50 buffer.delete(start: 2, end: 6); //menghapus karakter dari index ke-2 hingga index ke-5
51 System.out.printf(format: "buffer after deletes:\n%s\n", args: buffer.toString());
52 }
53 }

```

c. Capture Output

```

run:
buffer after inserts:
33.333 2.5 10000000 7 K true def abcdef goodbye hello

buffer after deletes:
33 2. 10000000 7 K true def abcdef goodbye hello
BUILD SUCCESSFUL (total time: 0 seconds)

```

d. Analisis

Menggunakan kelas StringBuffer untuk menyisipkan berbagai tipe data dengan insert(). Data yang ditambahkan, yaitu objek String, array karakter, boolean, karakter, integer, long, float, dan double. Setiap data dipisahkan oleh spasi dan semua data ditampilkan. Setelah itu, menampilkan hasil dari penghapusan karakter pada index ke-10 dan menghapus karakter dari index ke-2 hingga index ke-6.

15. Soal Nomor 15

a. Soal

StringBufferAppend

b. Capture Code

```
1 package Modul1;
2
3 //maria gresia_235314094
4 public class StringBufferAppend {
5
6     public static void main(String[] args) {
7         Object objectRef = "hello";
8         String string = "goodbye";
9         char charArray[] = {'a', 'b', 'c', 'd', 'e', 'f'};
10        boolean booleanValue = true;
11        char characterValue = 'Z';
12        int integerValue = 7;
13        long longValue = 10000000000L;
14        float floatValue = 2.5f;
15        double doubleValue = 33.333;
16
17        StringBuffer lastBuffer = new StringBuffer(str:"last StringBuffer");
18
19        StringBuffer buffer = new StringBuffer();
20
21        //menambahkan string dari objek
22        buffer.append(obj:objectRef);
23        buffer.append(str:"\n");
24        //menambahkan string ke StringBuffer
25        buffer.append(str:string);
26        buffer.append(str:"\n");
27        //menambahkan seluruh array karakter
28        buffer.append(str:charArray);
29
30        buffer.append(str:charArray);
31        buffer.append(str:"\n");
32        //menambahkan sebagian dari array karakter
33        buffer.append(str:charArray, offset: 0, len:3);
34        buffer.append(str:"\n");
35        //menambahkan nilai boolean
36        buffer.append(b: booleanValue);
37        buffer.append(str:"\n");
38        //menambahkan karakter
39        buffer.append(c: characterValue);
40        buffer.append(str:"\n");
41        //menambahkan nilai integer
42        buffer.append(i: integerValue);
43        buffer.append(str:"\n");
44        //menambahkan nilai long
45        buffer.append(lng: longValue);
46        buffer.append(str:"\n");
47        //menambahkan nilai float
48        buffer.append(f: floatValue);
49        buffer.append(str:"\n");
50        //menambahkan nilai double
51        buffer.append(d: doubleValue);
52        buffer.append(str:"\n");
53        //menambahkan seluruh isi dari StringBuffer
54        buffer.append(sb: lastBuffer);
55
56        //menampilkan isi dari objek buffer
57        System.out.printf(format: "buffer contains %s\n", args:buffer.toString());
58    }
59 }
```

c. Capture Output

```
run:
buffer contains hello
goodbye
abcdef
abc
true
Z
7
100000000000
2.5
33.333
last StringBuffer
BUILD SUCCESSFUL (total time: 0 seconds)
```

d. Analisis

Menggunakan kelas StringBuffer untuk menambahkan berbagai tipe data dengan append(). Data yang ditambahkan, yaitu objek String, array karakter, boolean, karakter, integer, long, float, dan double. Setiap data dipisahkan oleh baris baru. Setelah itu, isi dari StringBuffer akan ditampilkan dengan menggunakan toString()

16. Soal Nomor 16

a. Soal

TokenTest

b. Capture Code

```
1 package Modul1;
2
3 import java.util.Scanner;
4 import java.util.StringTokenizer;
5
6 //maria gresia_235314094
7 public class TokenTest {
8
9     public static void main(String[] args) {
10         Scanner scanner = new Scanner(System.in);
11
12         System.out.print(s: "Enter a sentence and press Enter : "); //meminta input
13         String sentence = scanner.nextLine(); //menyimpan input ke dalam "sentence"
14
15         //menghapus huruf vokal dan memisahkannya dari kata selanjutnya
16         StringTokenizer tokens = new StringTokenizer(str:sentence, delim: "[aeiou]");
17
18         //menampilkan jumlah kata yang terpisah
19         System.out.printf(format: "Number of elements: %d\nThe tokens are:\n", args: tokens.countTokens());
20
21         //menampilkan hasil token
22         while (tokens.hasMoreTokens()) {
23             System.out.println(x: tokens.nextToken());
24         }
25     }
26 }
```

c. Capture Output

```
run:
Enter a sentence and press Enter : maria gresia
Number of elements: 4
The tokens are:
m
r
gr
s
```

d. Analisis

Meminta user untuk menginput kalimat dan menyimpannya ke dalam variabel “sentence”. Kemudian menghapus huruf vocal dan memisahkannya dari karakter selanjutnya dengan menggunakan StringTokenizer yang disimpan ke dalam “tokens”. Lalu, menampilkan jumlah kata yang terpisah dengan menggunakan countTokens() pada tokens. Lalu, menampilkan hasil token satu per satu.

17. Soal Nomor 17

a. Soal

StringConstructors

b. Capture Code

```
1 package Modul1;
2
3 import java.util.*;
4
5 //maria gresia_235314094
6 public class StringConstructors {
7
8     public static void main(String[] args) {
9         char charArray[] = {'b', 'i', 'r', 't', 'h', ' ', ' ', 'd', 'a', 'y'};
10        String s = new String(charArray);
11        String s1 = new String();
12        String s2 = new String(s);
13        String s3 = new String(charArray);
14        String s4 = new String(charArray, 6, 3);
15
16        //menampilkan nilai dari s1, s2, s3, dan s4
17        System.out.println("s1 = " + s1);
18        System.out.println("s2 = " + s2);
19        System.out.println("s3 = " + s3);
20        System.out.println("s4 = " + s4);
21
22        //equals digunakan untuk membandingkan kata yang tidak mengabaikan kapitalisasi huruf
23        //atau harus sama persis
24
25        //jika s sama dengan "hello" dengan tidak mengabaikan kapitalisasi huruf
26        if (s.equals("hello")) {
27            //maka akan menampilkan pesan berikut
28            System.out.println("s is equals as \"hello\"");
```

```

29     } else {
30         //jika tidak, maka akan menampilkan pesan ini
31         System.out.println(x: "s is not equals as\"hello\"");
32     }
33
34     //equalsIgnoreCase digunakan untuk membandingkan kata yang mengabaikan kapitalisasi huruf
35     //atau tidak harus sama persis
36
37     //jika s sama dengan "HELLO" dengan mengabaikan kapitalisasi huruf
38     if (s.equalsIgnoreCase(anotherString: "HELLO")) {
39         //maka akan menampilkan pesan ini
40         System.out.println(x: "s is equals as\"hello\"");
41     } else {
42         //jika tidak, maka akan menampilkan pesan ini
43         System.out.println(x: "s is not equals as\"hello\"");
44     }
45
46     //jika s3 mengandung "day"
47     if (s3.contains(s: "day")) {
48         //maka akan menampilkan pesan ini
49         System.out.println(x: "s3 contains\"day\"");
50     } else {
51         //jika tidak, maka akan menampilkan pesan ini
52         System.out.println(x: "s3 do not contains\"day\"");
53     }
54
55     //concat digunakan untuk menggabungkan dua string

```

```

56
57     //menggabungkan s dengan s3 dan disimpan ke dalam s5
58     String s5 = s.concat(str: " ") + s3;
59     //menampilkan hasil penggabungan dari s5
60     System.out.println(x: s5);
61
62     //replaceAll digunakan untuk mengganti suatu kata dengan kata baru
63
64     //mengganti "hello" dengan "happy" pada s5 dan disimpan ke dalam s6
65     String s6 = s5.replaceAll(regex: "hello", replacement: "happy");
66     //menampilkan hasil penggantian dari s6
67     System.out.println(x: s6);
68
69     //membuat array yang bernama s7 dengan panjang 3
70     String[] s7 = new String[3];
71     //memecah s6 dengan spasi
72     s7 = s6.split(regex: " ");
73
74     //melakukan perulangan dengan i kurang dari panjang s7
75     //menampilkan setiap elemennya
76     for (int i = 0; i < s7.length; i++) {
77         System.out.println(s7[i]);
78     }
79
80     //memisahkan s6 berdasarkan spasi dengan menggunakan StringTokenizer
81     StringTokenizer tokens = new StringTokenizer(str: s6);
82
83     //menampilkan jumlah token dari hasil pemisahan
84     System.out.println("Number of Token = " + tokens.countTokens());

```

```

85
86     //menampilkan setiap token selama masih ada sisa
87     while (tokens.hasMoreElements()) {
88         System.out.println(x: tokens.nextToken());
89     }
90 }
91 }

```

c. Capture Output

```
run:
s1 =
s12 = hello
s3 = birth day
s4 = day
s is equals as"hello"
s is equals as"hello"
s3 contains"day"
hello birth day
happy birth day
happy
birth
day
Number of Token = 3
happy
birth
day
BUILD SUCCESSFUL (total time: 0 seconds)
```

d. Analisis

Menampilkan nilai dari s1, s2, s3, dan s4. Kemudian, membandingkan nilai s menggunakan equals yang tidak mengabaikan kapitalisasi dan equalsIgnoreCase yang mengabaikannya. Jika s3 mengandung “day” program akan mencetak pesan tertentu. Kemudian ,program menggabungkan s dengan s3, menyimpan hasilnya ke s5. Lalu, mengganti "hello" dengan "happy" dalam s5 dan menyimpan hasilnya ke s6 menggunakan replaceAll(). Program memecah s6 menjadi array s7 dengan spasi. Kemudian terdapat perulangan for untuk menampilkan setiap elemennya. String s6 dipisah dengan menggunakan StringTokenizer menampilkan jumlah token, dan menampilkan setiap token yang tersisa.

18. Soal Nomor 18

a. Soal

RegexSubstitution

b. Capture Code

```
1 package Modul1;
2
3 //maria gresia_235314094
4 public class RegexSubstitution {
5
6     public static void main(String[] args) {
7         String firstString = "Every day is sunday";
8         String secondString = "Vini pergi ke pasar 3 jam lalu, bersama 4 temannya";
9
10        System.out.printf(format: "Original String 1: %s\n", args: firstString);
11
12        //mengganti kata "day" dengan "week there"
13        firstString = firstString.replaceAll(regex: "\\bday\\b", replacement: "week there");
14        System.out.printf(format: "Original string 1 become : %s\n", args: firstString);
15
16        System.out.printf(format: "Original String 2: %s\n", args: secondString);
17
18        //mengganti setiap huruf "a" dengan "o"
19        secondString = secondString.replaceAll(regex: "[a]", replacement: "o");
20        System.out.printf(format: "Original String 2: %s\n", args: secondString);
21
22        //mengganti setiap kata dengan "word"
23        System.out.printf(format: "Every word replaced by \\\"word\\\": %s\n",
24                          args: firstString.replaceAll(regex: "\\W+", replacement: "word"));
25
26        //mengganti angka pertama dengan "digit"
27        secondString = secondString.replaceFirst(regex: "\\d", replacement: "digit");
28        System.out.printf(format: "Replaced second string: %s\n", args: secondString);
29
30        //memisahkan kalimat yang memiliki koma menjadi kalimat yang terpisah
31        String output = "String is split at commas: [";
32
33        //memisahkan kata berdasarkan koma yang diikuti oleh spasi
34        String[] results = secondString.split(regex: ",", \\s*");
35
36        for (String string : results) {
37            output += "\\n" + string + "\\n, ";
38        }
39        //menampilkan hasil pemisahan
40        System.out.println("Hasil split : " + output);
41
42        //menghapus koma dan spasi terakhir, lalu memberikan "]" di akhir
43        output = output.substring(beginIndex: 0, output.length() - 2) + "]";
44        //menampilkan hasil pemisahan
45        System.out.println(x: output);
46    }
47 }
```

c. Capture Output

run:

```
Original String 1: Every day is sunday
Original string 1 become : Every week there is sunday
Original String 2: Vini pergi ke pasar 3 jam lalu, bersama 4 temannya
Original String 2: Vini pergi ke posor 3 jom lolu, bersomo 4 temonnyo
Every word replaced by "word": word word word word word
Replaced second string: Vini pergi ke posor digit jom lolu, bersomo 4 temonnyo
Hasil split : String is split at commas: ["Vini pergi ke posor digit jom lolu", "bersomo 4 temonnyo",
String is split at commas: ["Vini pergi ke posor digit jom lolu", "bersomo 4 temonnyo"]
BUILD SUCCESSFUL (total time: 0 seconds)
```

d. Analisis

Menyimpan teks ke dalam variabel “firstString” dan “secondString”. Menampilkan teks asli dari “firstString”. Mengganti kata "day" dengan "week there" pada “firstString” lalu menampilkannya. Menampilkan teks asli dari “secondString”. Mengganti setiap huruf "a" dengan "o" pada “secondString” lalu menampilkannya. Mengganti setiap kata pada “firstString” dengan “word” lalu menampilkannya. Dilanjut dengan mengganti angka pertama dengan "digit" pada “secondString” lalu menampilkannya. Kemudian memisahkan koma kalimat yang memiliki koma menjadi kalimat yang terpisah dan memisahkan kata berdasarkan koma yang diikuti oleh spasi. Kemudian menampilkan hasil pemisahannya yang dilanjutkan dengan menampilkan hasil penghapusan koma dan spasi terakhir, lalu memberikan "]" di akhir.

19. Soal Nomor 19

a. Soal

Masukan berupa angka 5 – 10 karakter.

b. Capture Code

```
1 package Modul1;
2
3 import java.util.regex.*;
4 import java.util.Scanner;
5
6 //maria gresia_235314094
7 public class RegularExpression {
8
9     public static void main(String[] args) {
10         Scanner in = new Scanner(System.in);
11
12         //soal no 4a
13         System.out.print("Masukkan angka = "); //meminta input
14         String masukan = in.next(); //menyimpan input ke dalam "masukan"
15
16         //memeriksa apakah masukan terdiri dari 5 hingga 10 digit angka
17         boolean valid = masukan.matches(regex: "\\d{5,10}");
18         if (valid) {
19             System.out.println("sesuai");
20         } else {
21             System.out.println("tidak sesuai");
22         }
23     }
24 }
```

c. Capture Output

```
run:
Masukkan angka = 2587564
sesuai
BUILD SUCCESSFUL (total time: 5 seconds)
```


d. Analisis

Meminta user untuk menginput kata dan menyimpannya ke dalam “masukan”. Mencocokkan apakah “masukan” sesuai dengan pola yang diinginkan, yaitu masukan berupa angka sebanyak 5 hingga 10 karakter. Lalu, hasilnya akan disimpan ke dalam variabel “valid”. Jika sesuai, maka akan menampilkan pesan sesuai dan jika tidak, maka akan menampilkan pesan tidak sesuai.

20. Soal Nomor 20

a. Soal

Masukan berakhiran ‘*’ atau ‘#’, diawali dengan angka, karakter berikutnya berupa minimal satu huruf besar dan berikutnya huruf kecil atau huruf besar, panjang minimal 8 karakter.

b. Capture Code

```
1 package Modul1;
2
3 import java.util.regex.*;
4 import java.util.Scanner;
5
6 //maria gresia_235314094
7 public class RegularExpression1 {
8
9     public static void main(String[] args) {
10         Scanner in = new Scanner(System.in);
11
12         //soal no 4b
13         System.out.print(s: "Masukkan kata = "); //meminta input
14         String masukan1 = in.next(); //menyimpan input ke dalam "masukan1"
15
16         //memeriksa apakah masukan1 sesuai dengan format, yaitu diawali dengan angka
17         //kemudian dilanjutkan dengan minimal satu huruf besar dan selanjutnya
18         //huruf kecil atau besar minimal 8 karakter. lalu, diakhiri dengan "*" atau "#"
19         boolean isValid = masukan1.matches(regex: "\\d[A-Z][A-Za-z]{6,}[*#]");
20         if (isValid) {
21             System.out.println(x: "sesuai");
22         } else {
23             System.out.println(x: "tidak sesuai");
24         }
25     }
26 }
```

c. Capture Output

```
run:
Masukkan kata = 3Gfwerwq*
sesuai
BUILD SUCCESSFUL (total time: 11 seconds)
```

d. Analisis

Meminta user untuk menginput kata dan menyimpannya ke dalam “masukan1”. Mencocokkan apakah “masukan1” sesuai dengan pola yang diinginkan, yaitu diawali dengan angka dan dilanjut dengan minimal satu huruf kapital. Kemudian, huruf kecil atau huruf besar dengan panjang minimal 8 karakter. Lalu, hasilnya akan disimpan ke dalam variabel “isValid”. Jika sesuai, maka akan menampilkan pesan sesuai dan jika tidak, maka akan menampilkan pesan tidak sesuai.

21. Soal Nomor 21

a. Soal

Masukan berupa huruf kecil atau huruf besar dan tidak boleh mengandung angka.

b. Capture Code

```
1 package Modul1;
2
3 import java.util.regex.*;
4 import java.util.Scanner;
5
6 //maria gresia_235314094
7 public class RegularExpression2 {
8
9     public static void main(String[] args) {
10         Scanner in = new Scanner(System.in);
11
12         //soal no 4c
13         System.out.print("Masukkan kata = "); //meminta input
14         String masukan2 = in.next(); //menyimpan input ke dalam "masukan2"
15
16         //memeriksa apakah masukan2 terdiri dari huruf kecil atau besar
17         //dan tidak boleh mengandung angka
18         boolean benar = masukan2.matches(regex: "[a-zA-Z]+");
19         if (benar) {
20             System.out.println("sesuai");
21         } else {
22             System.out.println("tidak sesuai");
23         }
24     }
25 }
```

c. Capture Output

```
run:
Masukkan kata = fkwfJgfi
sesuai
BUILD SUCCESSFUL (total time: 3 seconds)
```

d. Analisis

Meminta user untuk menginput kata dan menyimpannya ke dalam “masuk2”. Mencocokkan apakah “masuk2” sesuai dengan pola yang diinginkan, yaitu hanya terdiri dari huruf kecil atau huruf besar saja. Lalu, hasilnya akan disimpan ke dalam variabel “benar”. Jika sesuai, maka akan menampilkan pesan sesuai dan jika tidak, maka akan menampilkan pesan tidak sesuai.

22. Soal Nomor 22

a. Soal

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

b. Capture Code

```
1 package Modul1;
2
3 import java.util.regex.*;
4 import java.util.Scanner;
5
6 //maria gresia 235314094
7 public class RegularExpression11 {
8
9     public static void main(String[] args) {
10         Scanner in = new Scanner(System.in);
11
12         //soal no 4d
13         System.out.println();
14         String teks = "<html lang=id><head><meta charset=UTF-8> "
15             + "<meta name=viewport content=width=device-width, "
16             + "initial-scale=1.0> <h1>Selamat Datang di Situs Kami!</h1> \n</header> <main> "
17             + "<p>Ini adalah paragraf pertama dari halaman web "
18             + "contoh ini. \nHTML adalah bahasa yang digunakan untuk membuat struktur halaman web.</p> "
19             + "<p> \nAnda bisa belajar lebih lanjut tentang HTML di <a href=https://www.htmlschools.com \" "
20             + "target=_blank\">HTMLSchools</a>.</p> </main><footer> <p> \n2024 "
21             + "Contoh Halaman Web. Semua hak cipta dilindungi.</p></footer></body></html>";
22
23         //menghapus semua tag HTML
24         String teksBaru = teks.replaceAll(regex: "<[^>]*>", replacement:""); //menghapus tag HTML
25
26         //menampilkan hasil teks setelah penghapusan tag HTML
27         System.out.println(*: teksBaru);
28     }
29 }
```

c. Capture Output

run:

```
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.
BUILD SUCCESSFUL (total time: 0 seconds)
```

d. Analisis

Menghapus semua tag HTML dengan menggunakan `replaceAll()` yang disimpan ke dalam `teksBaru`. Lalu, menampilkannya.

C. DAFTAR PUSTAKA