**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**

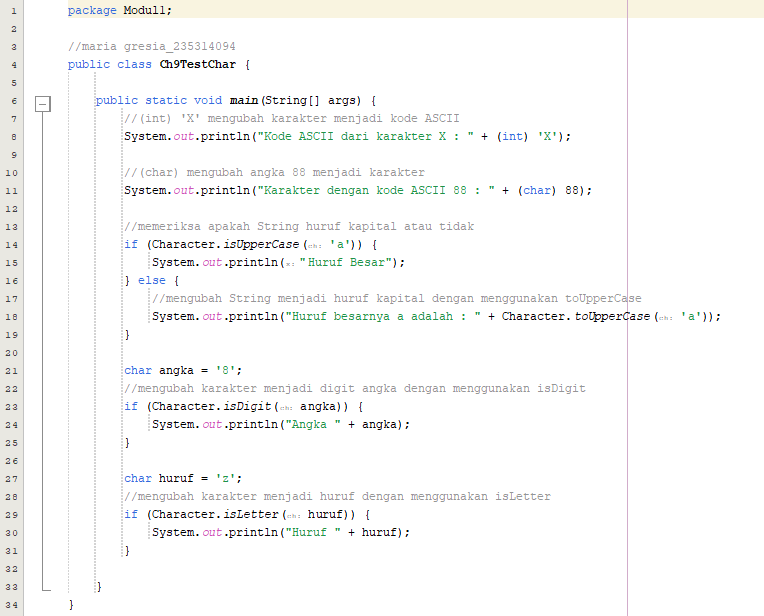
**YOGYAKARTA**

**2024**

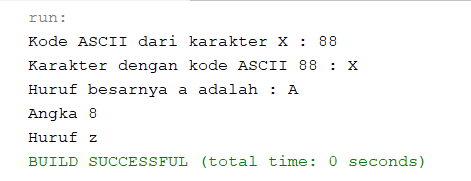
1. **TUJUAN**
2. …
3. **PELAKSANAAN PRAKTIKUM**
   1. **Soal Nomor 1**
4. **Soal**

Ch9TestChar

1. **Capture Code**

****

1. **Capture Output**



1. **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().

* 1. **Soal Nomor 2**

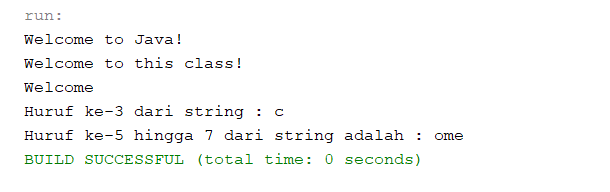
1. **Soal**

Ch9TestString

1. **Capture Code**

****

1. **Capture Output**

****

1. **Analisis**

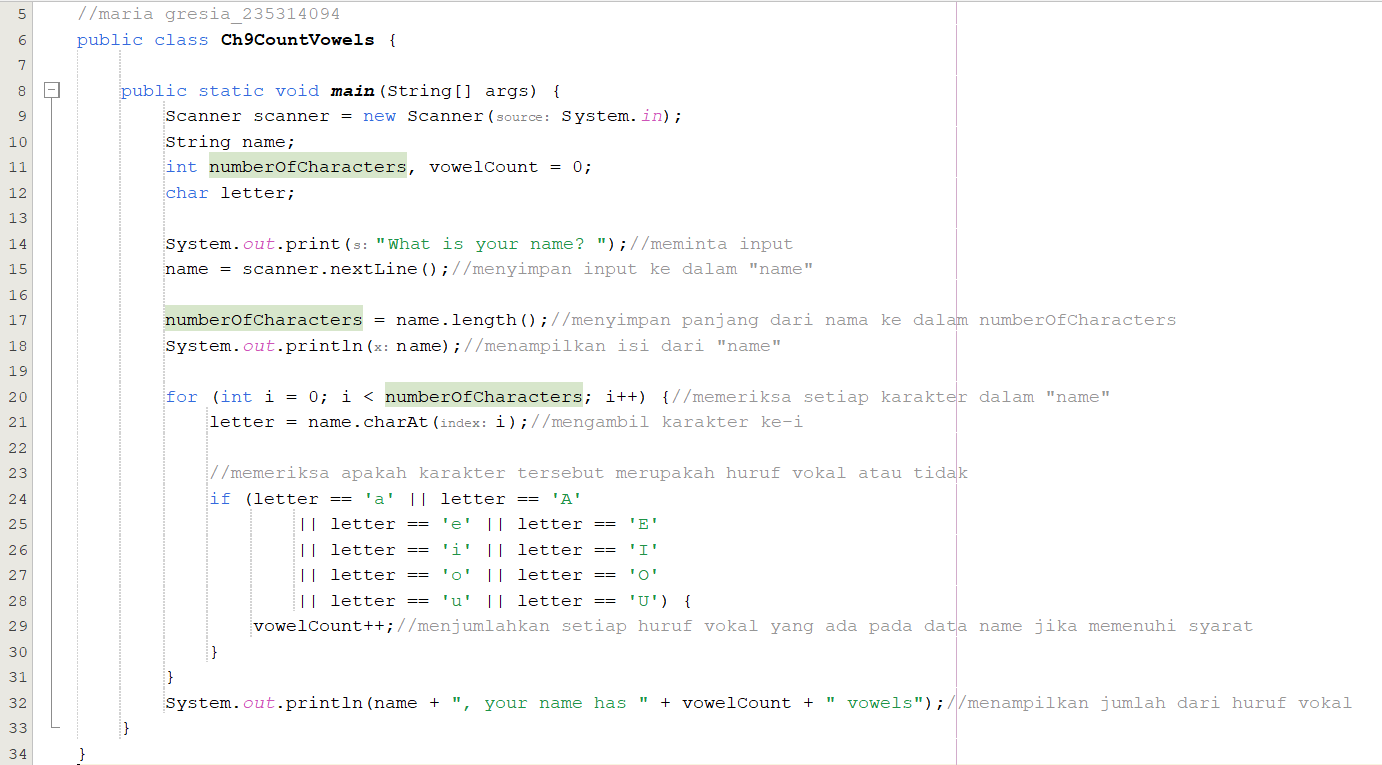
Menampilkan isi dari message, pesan, dan s. kemudian menampilkan huruf ke-3 dari s dengan menggunakan charAt untuk menentukan index nya. Lalu, menampilkan huruf ke-5 hingga ke-7 dengan menggunakan substring pada message untuk menentukan index awal dan akhir.

* 1. **Soal Nomor 3**

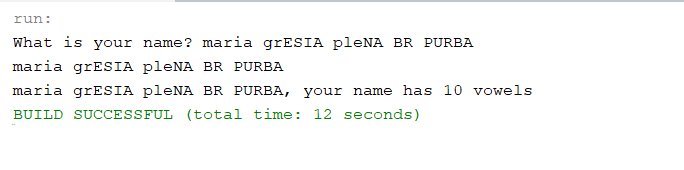
1. **Soal**

Ch9CountVowels

1. **Capture Code**



1. **Capture Output**

****

1. **Analisis**

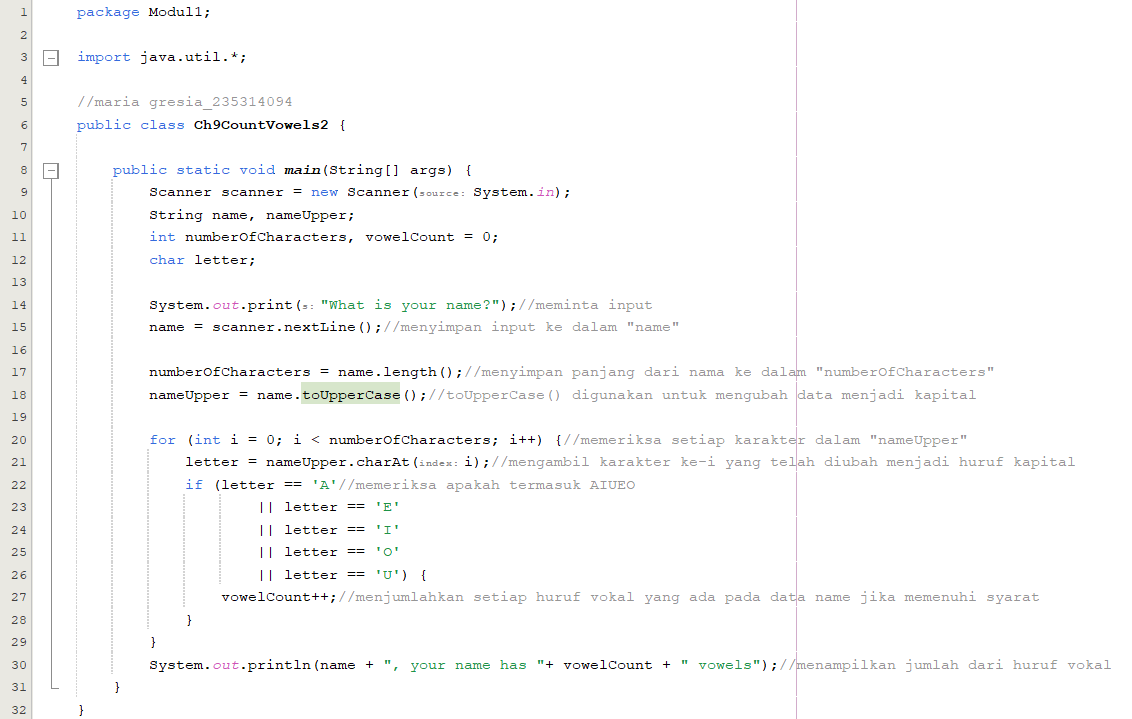
Meminta user untuk menginput nama kemudian menyimpannya ke dalam variabel name. Kemudian, menyimpan panjang dari nama ke dalam numberOfCharacters dan menampilkan isi dari variabel “name”. Terdapat perulangan for yang akan memeriksa setiap karakter dalam “name” untuk mengetahui apakah ada huruf vocal.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

* 1. **Soal Nomor 4**

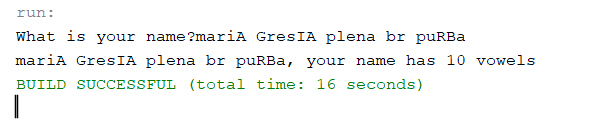
1. **Soal**

Ch9CountVowels2

1. **Capture Code**



1. **Capture Output**

****

1. **Analisis**

Meminta user untuk menginput nama dan menyimpannya dalam “name”. Kemudian, menyimpan panjang dari nama ke dalam numberOfCharacters dilanjut 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.

* 1. **Soal Nomor 5**

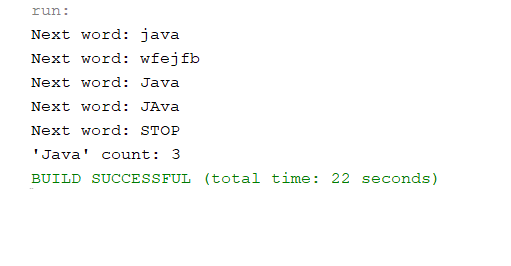
1. **Soal**

Ch9CountJava

1. **Capture Code**



1. **Capture Output**

****

1. **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.

* 1. **Soal Nomor 6**

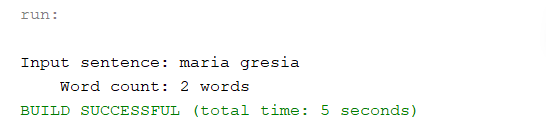
1. **Soal**

Ch9CountWords

1. **Capture Code**



1. **Capture Output**



1. **Analisis**

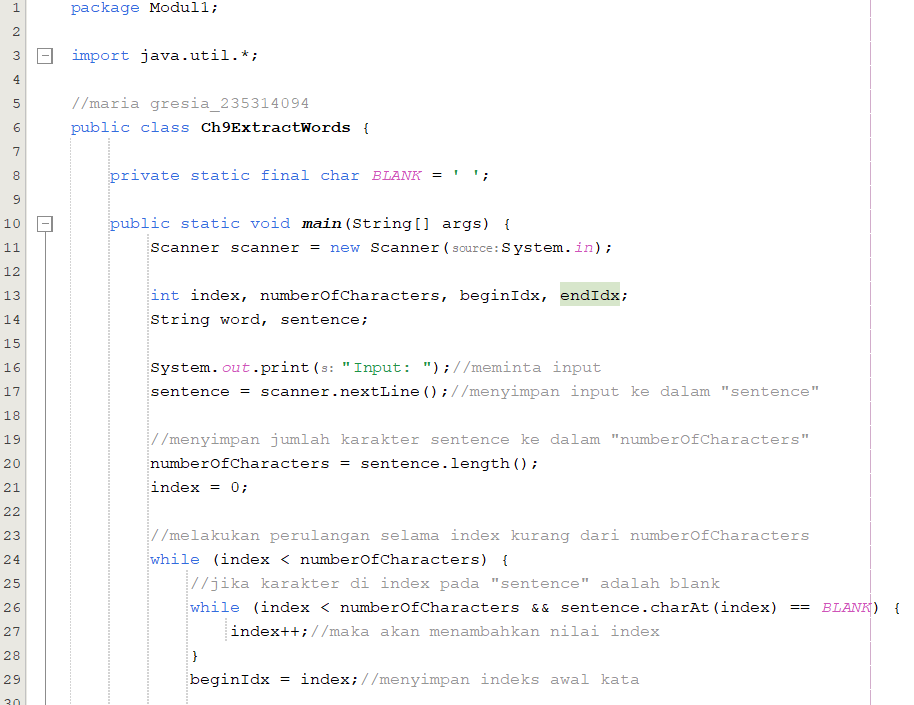
Meminta user untuk menginput suatu kata yang kemudian akan disimpan ke dalam “sentence”. Lalu, panjang dari “sentence” akan disimpan ke dalam “numberOfCharacters”. Terdapat perulangan while selama index kurang dari “numberOfCharacters”. Terdapat perulangan while di mana jika karakter di index pada “sentence” 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”.

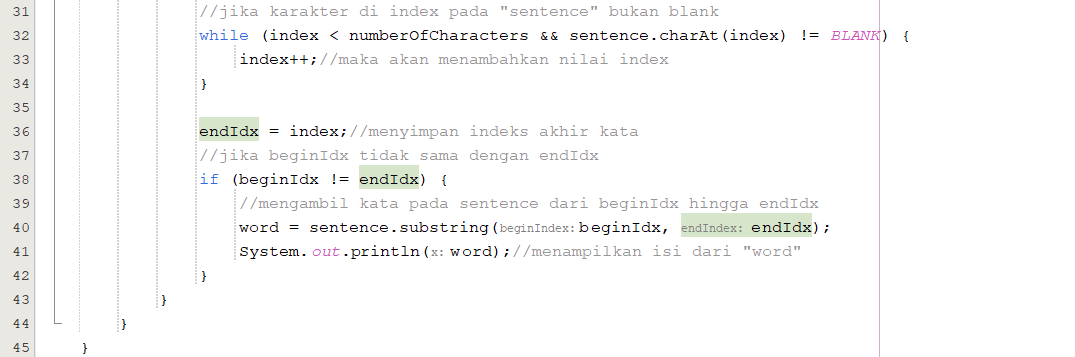
* 1. **Soal Nomor 7**

1. **Soal**

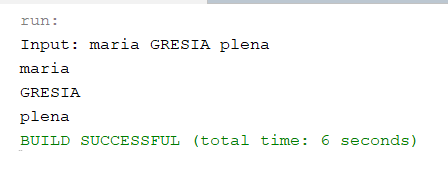
Ch9ExtractWords

1. **Capture Code**





1. **Capture Output**



1. **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 endIdx maka akan mengambil kata dari posisi beginIdx hingga endIdx yang disimpan ke dalam “word”. Lalu, menampilkan hasil dari “word”.

* 1. **Soal Nomor 8**

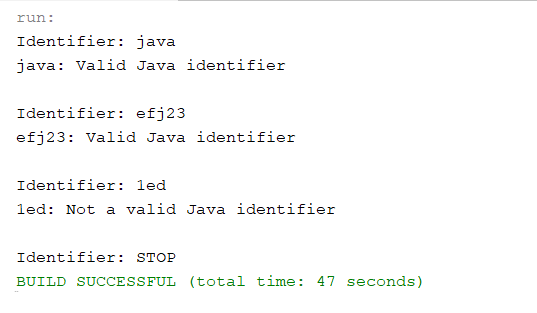
1. **Soal**

Ch9MatchJavaIdentifier

1. **Capture Code**



1. **Capture Output**



1. **Analisis**

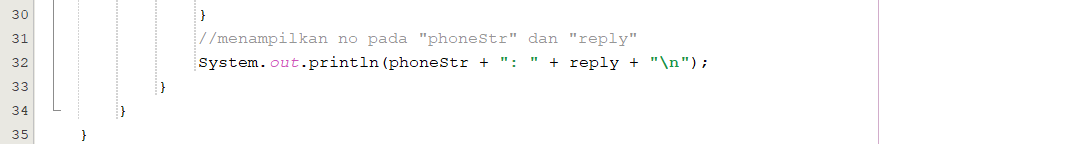
Pada perulangan while, user diminta untuk menginputkan identifier dan data tersebut akan disimpan ke dalam variabel str. Jika input sama dengan STOP maka perulangan berhenti. Jika “str” sesuai dengan pola yang diinginkan, yaitu diawali dengan huruf lalu dapat dilanjutkan dengan huruf, angka, \_, atau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

* 1. **Soal Nomor 9**
     1. **Soal**

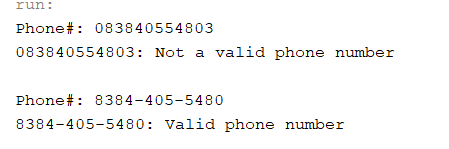
Ch9MatchPhoneNumber

* + 1. **Capture Code**





* + 1. **Capture Output**



* + 1. **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.

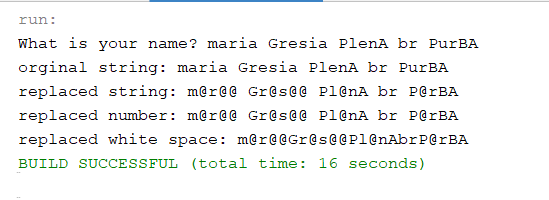
* 1. **Soal Nomor 10**
     1. **Soal**

Ch9ReplaceAll

* + 1. **Capture Code**



* + 1. **Capture Output**



* + 1. **Analisis**

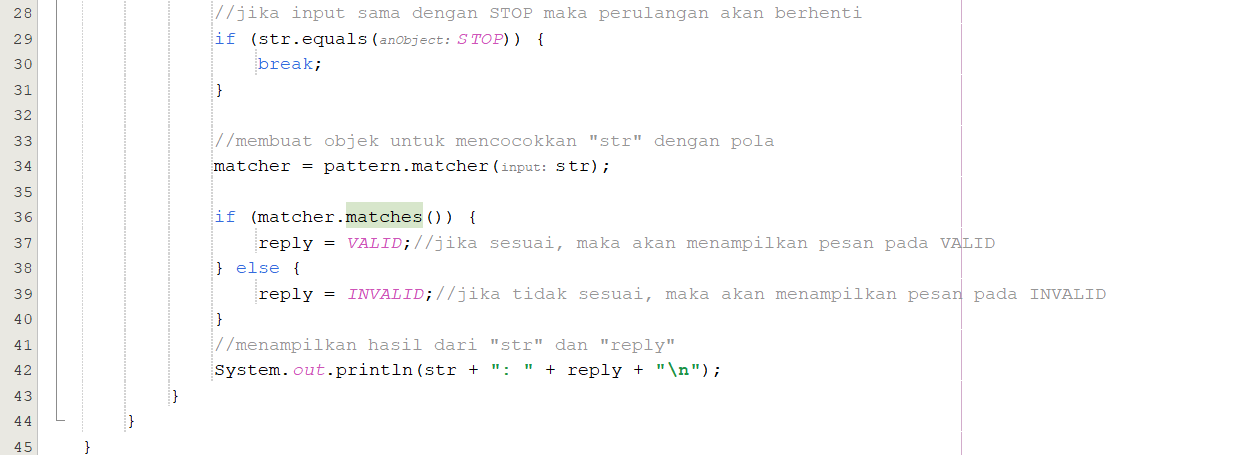
Meminta user untuk menginput nama yang akan disimpan ke dalam variabel “kalimat”. Kemudian mengganti setiap huruf voklat dengan “@” yang disimpan ke dalam variabel “diganti”. Dilanjut dengan menampilkan isi dari “kalimat” dan “diganti”. Lalu, menggunakan replaceAll() pada “diganti” untuk mengubah angka 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”.

* 1. **Soal Nomor 11**
     1. **Soal**

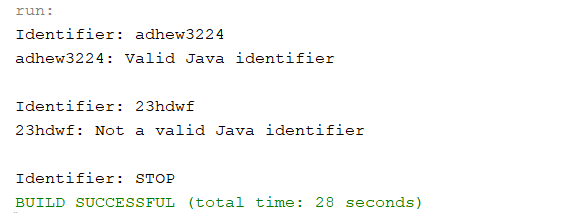
Ch9MatchJavaIdentifier2

* + 1. **Capture Code**





* + 1. **Capture Output**



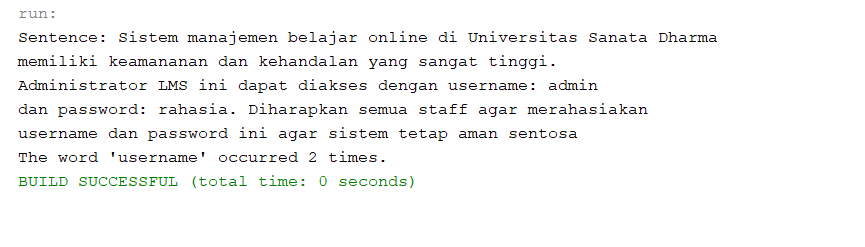
* + 1. **Analisis**

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 maka akan menampilkan pesan pada “valid”. Jika tidak, maka akan menampilkan pesan pada “invalid”

* 1. **Soal Nomor 12**
     1. **Soal**

Ch9PMCountJava

* + 1. **Capture Code**
    2. **Capture Output**



* + 1. **Analisis**

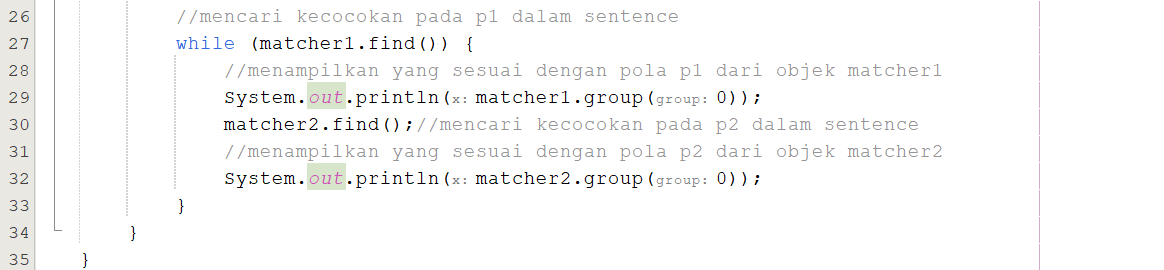
Menyimpan teks ke dalam variabel “document” yang bertipe string. Kemudian mencari berapa banyak “username” muncul dalam teks tersebut dengan menggunakan compile. Kemudian, membuat objek untik memeriksa pola yang sama dengan “document”. Lalu, mencari dan menghitung berapa kali pola itu muncul dan menjumlahkannya.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

* 1. **Soal Nomor 13**
     1. **Soal**

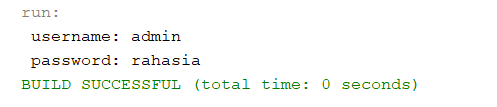
Ch9PMCountJava2

* + 1. **Capture Code**





* + 1. **Capture Output**



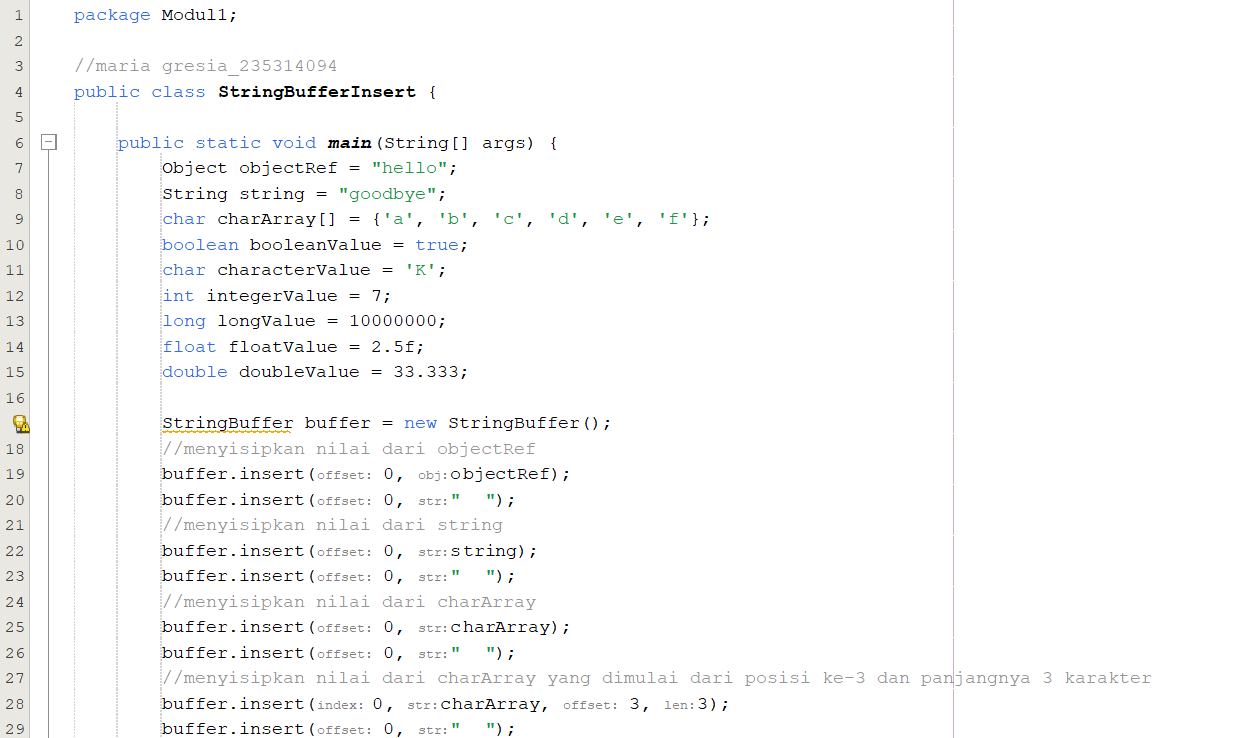
* + 1. **Analisis**

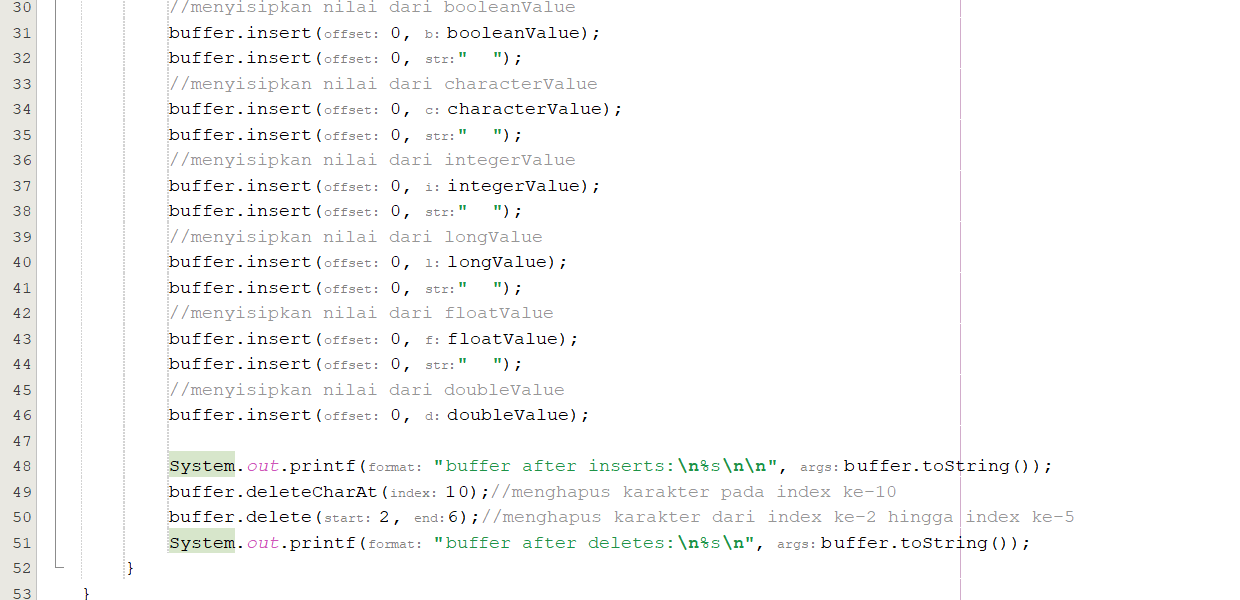
Menyimpan teks ke dalam variabel “sentence” yang bertipe string. Kemudian, membuat pola untuk username pada p1 dan password pada p2. Kemudian, membuat objek yang bernama matcher1 untuk p1 dan matcher2 untuk p2untuk memeriksa kesesuaian pada “sentence”. Lalu, terdapat perulangan while yang digunakan untuk mencari kecocokan pada p1 dan p2 dengan menggunakan find() pada matcher1 dan matcher2. Kemudian, menampilkan hasilnya.

* 1. **Soal Nomor 14**
     1. **Soal**

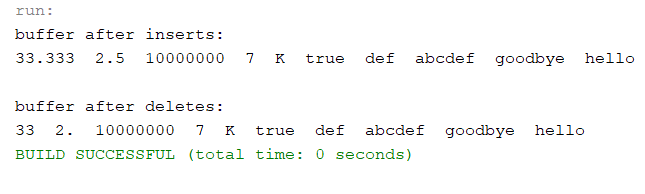
StringBufferInsert

* + 1. **Capture Code**





* + 1. **Capture Output**



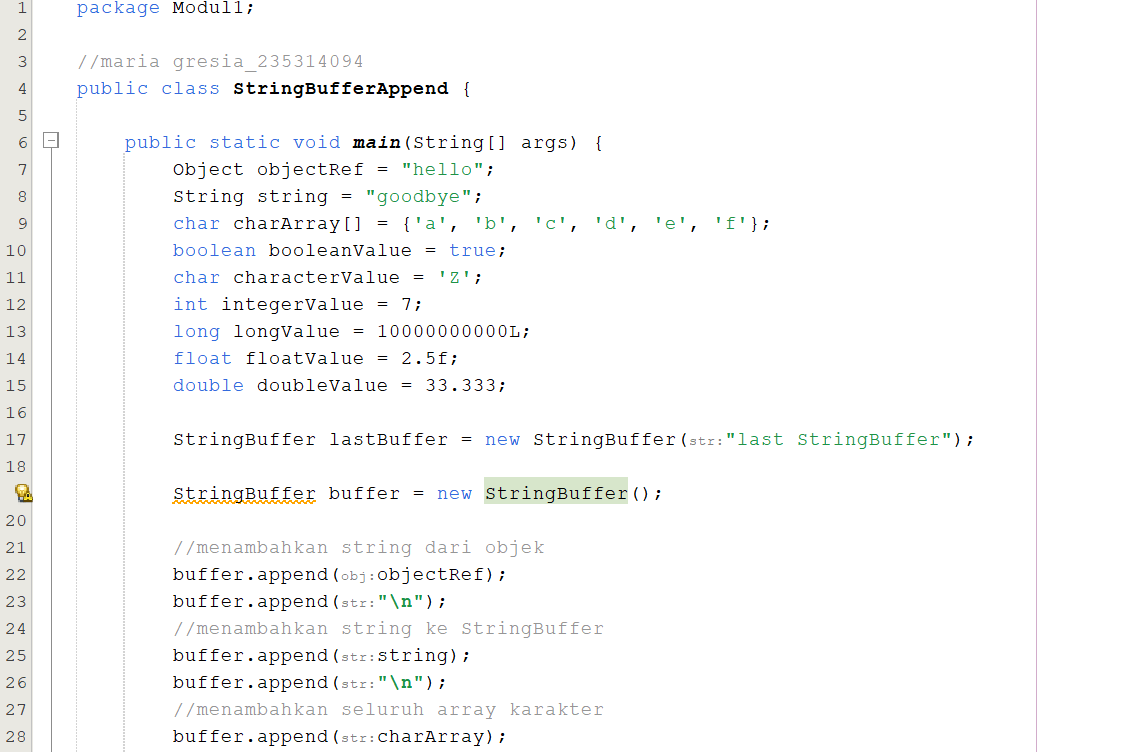
* + 1. **Analisis**

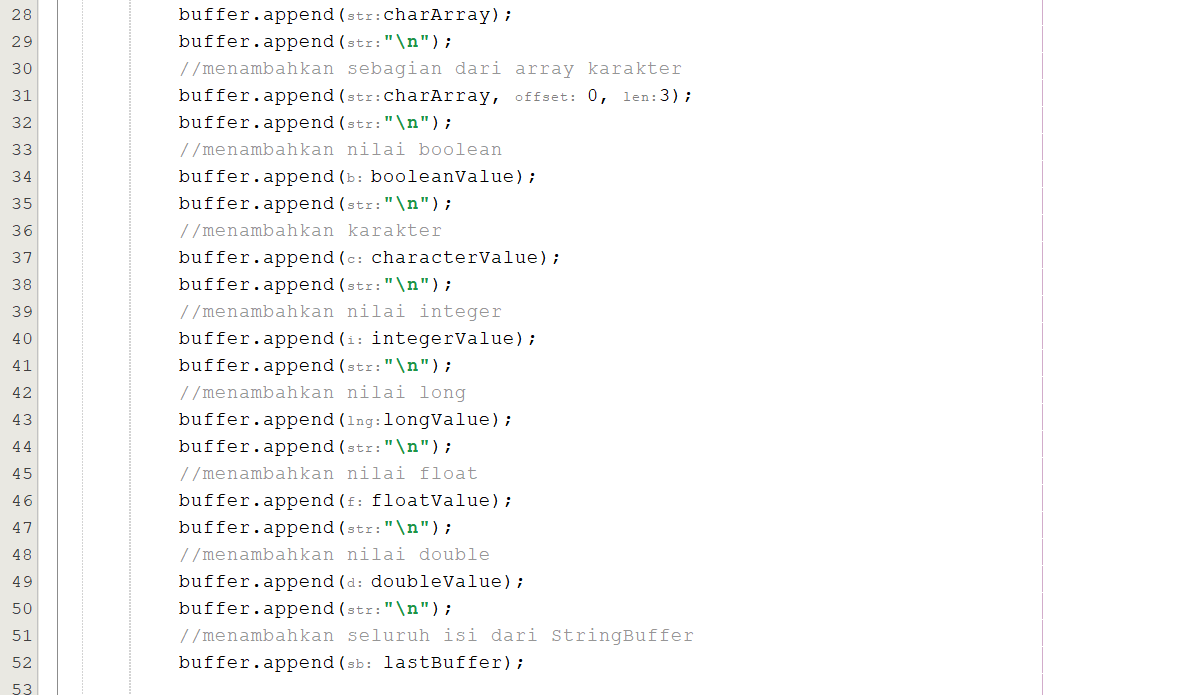
Menggunakan kelas StringBuffer untuk menyisipkan berbagai tipe data dengan insert(). Data yang ditambahkan, yaitu objek String, array karakter, boolean, karakter, integer, long, float,dan double. Setiap data dipisahkan oleh spasi dan semua data ditampilkan. Setelah itu, menampilkan hasil dari penghapusan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

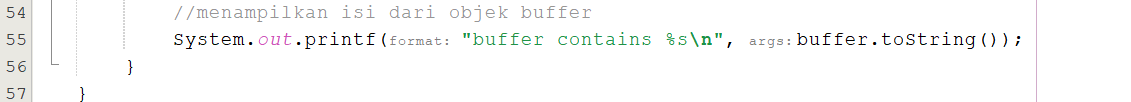
* 1. **Soal Nomor 15**
     1. **Soal**

StringBufferAppend

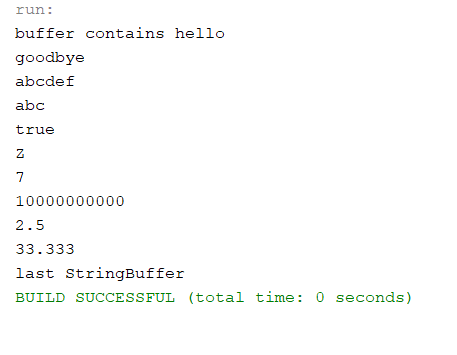
* + 1. **Capture Code**







* + 1. **Capture Output**



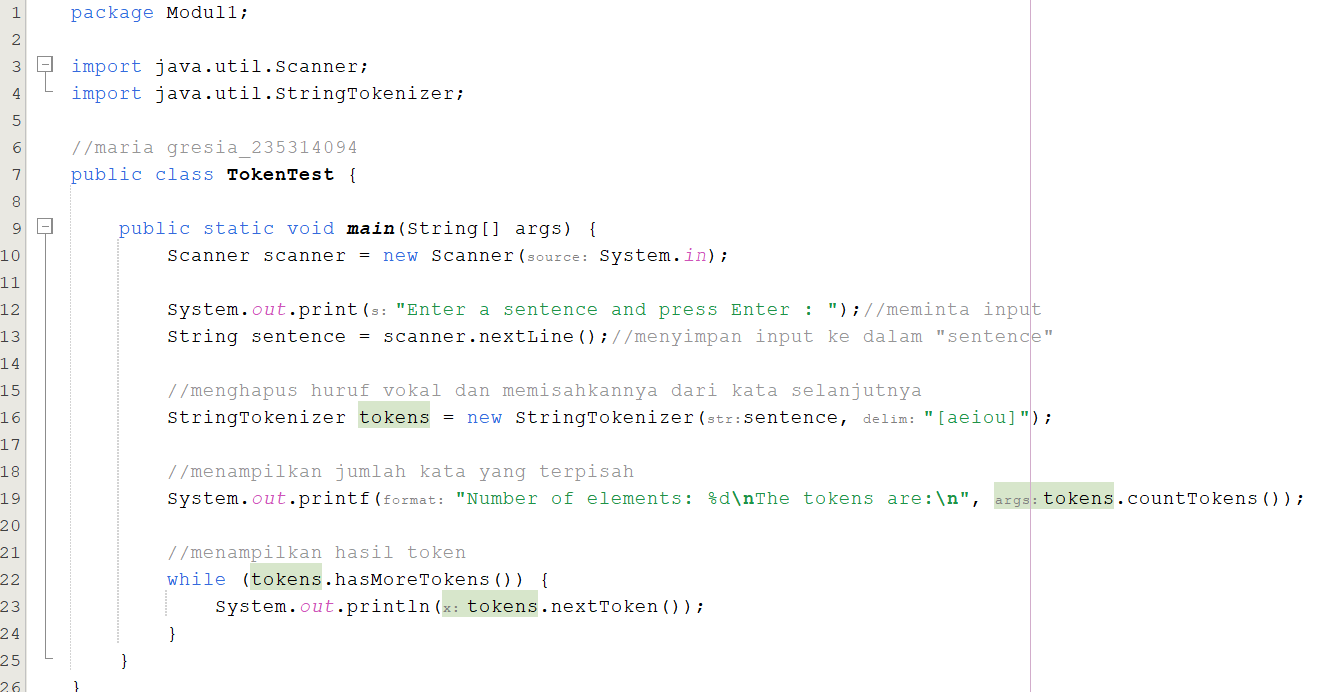
* + 1. **Analisis**

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()

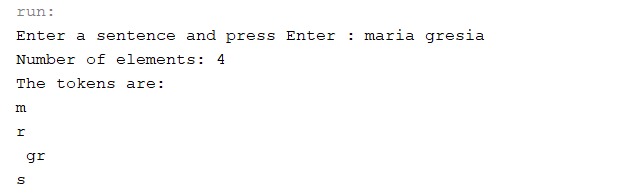
* 1. **Soal Nomor 16**
     1. **Soal**

TokenTest

* + 1. **Capture Code**



* + 1. **Capture Output**



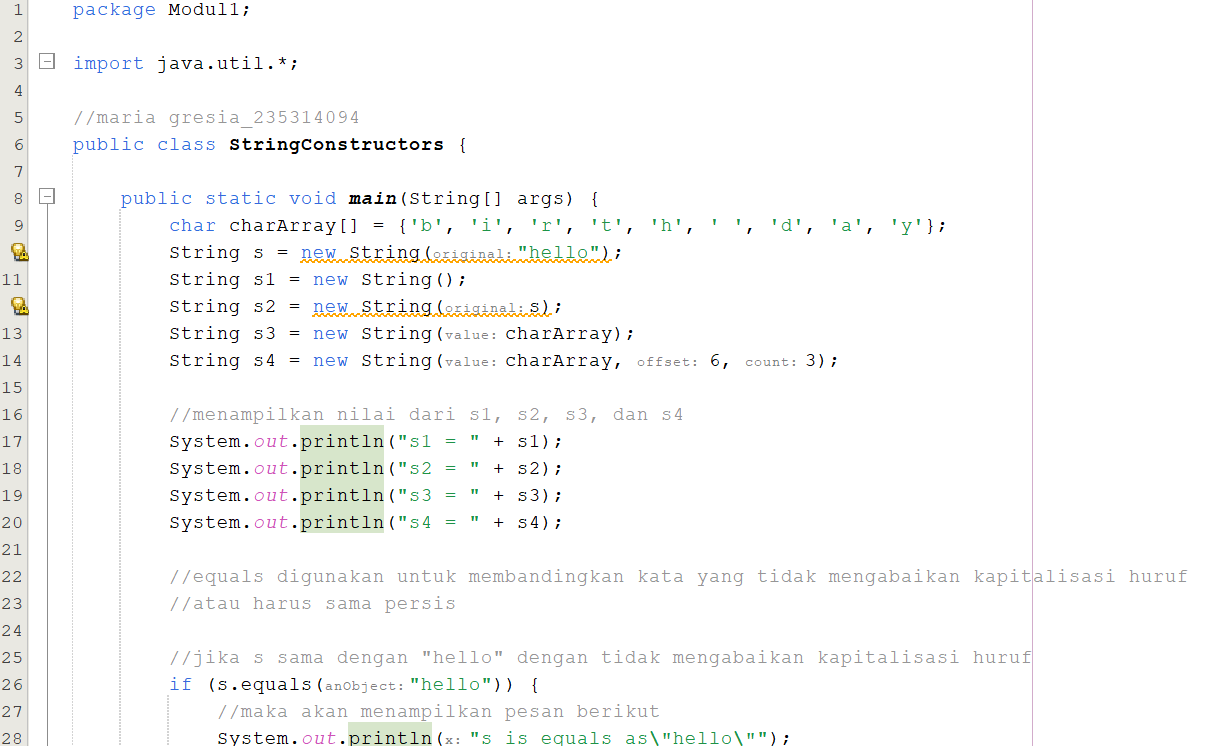
* + 1. **Analisis**

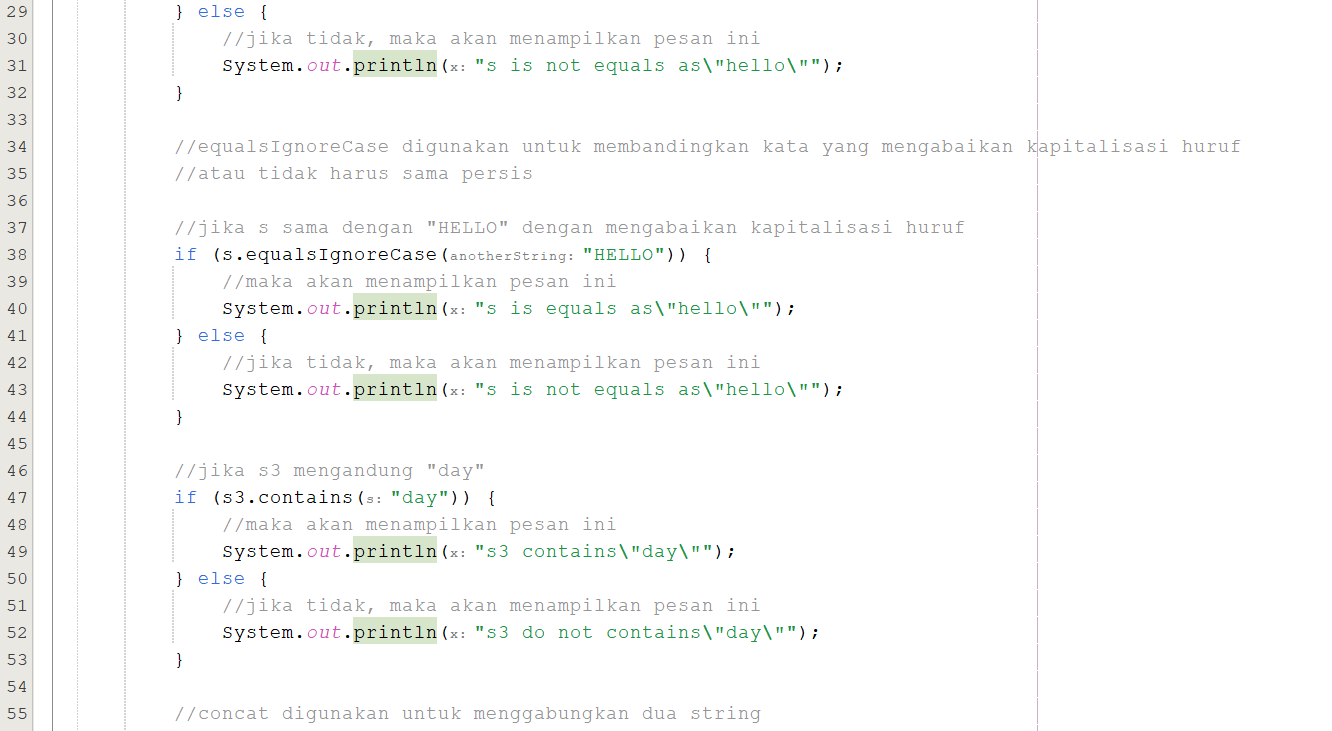
Meminta user untuk menginput kalimat dan menyimpannya ke dalam variabel “sentence”. Kemudian menghapus huruf vocal dan memisahkannya dari karakter selanjutnya dengan menggunakan StringTokenizer yang disimpan ke dalam “tokens”. Lalu, menampilkan jumlah kata yang terpisah dengan menggunakan countTokens()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

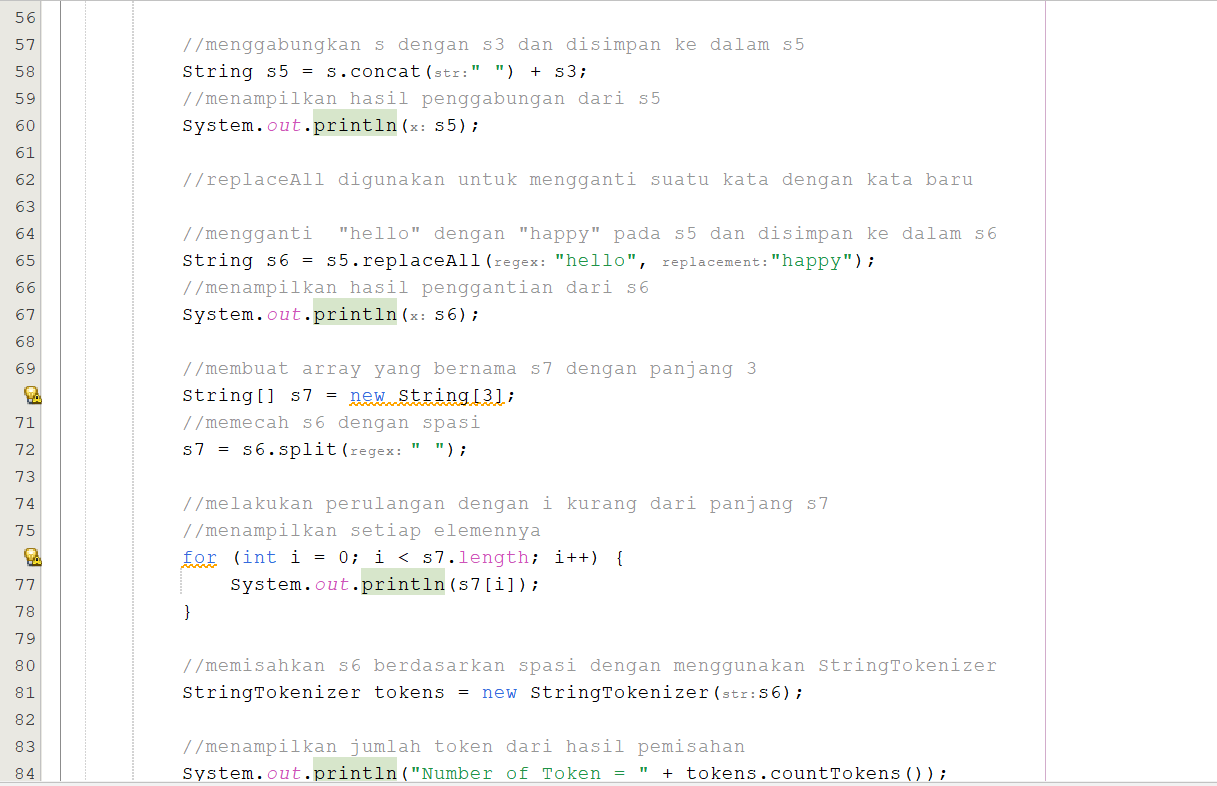
* 1. **Soal Nomor 17**
     1. **Soal**

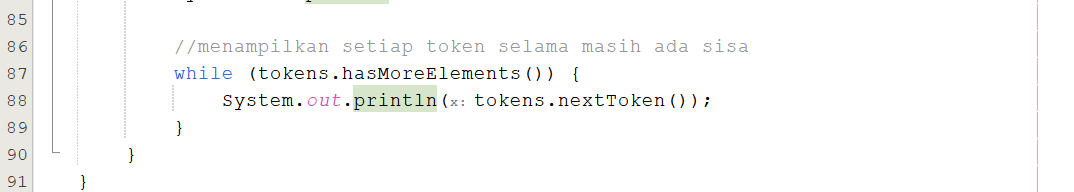
StringConstructors

* + 1. **Capture Code**

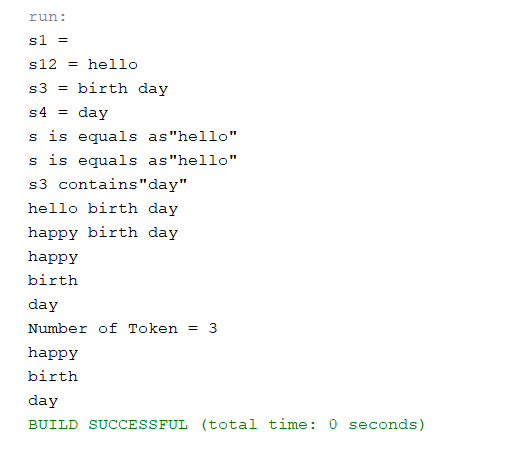








* + 1. **Capture Output**



* + 1. **Analisis**

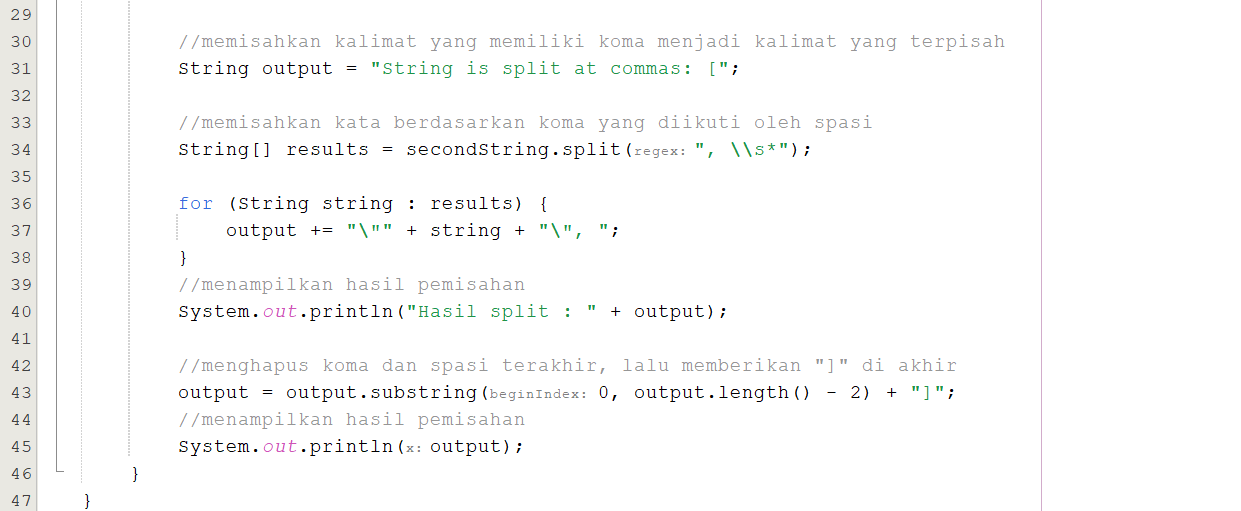
Menampilkan nilai dari s1, s2, s3, dan s4. Kemudian, membandingkan nilai s menggunakan equals yang tidak mengabaikan kapitalisasi dan equalsIgnoreCase yang mengabaikannya. Jika s3 mengandung “day” program akan mencetak pesan tertentu. Kemudian ,program menggabungkan s dengan s3, menyimpan hasilnya ke s5. Lalu, mengganti "hello" dengan "happy" dalam s5 dan menyimpan hasilnya ke s6 memggunakan 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.

* 1. **Soal Nomor 18**
     1. **Soal**

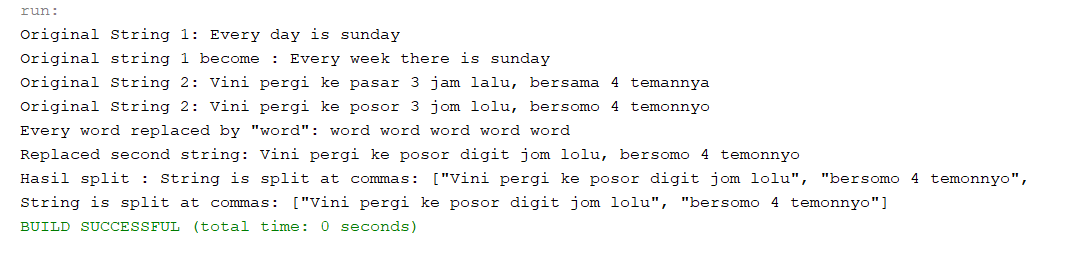
RegexSubstitution

* + 1. **Capture Code**





* + 1. **Capture Output**



* + 1. **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.

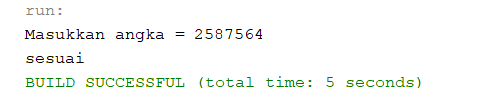
* 1. **Soal Nomor 19**
     1. **Soal**

Masukan berupa angka 5 – 10 karakter.

* + 1. **Capture Code**



* + 1. **Capture Output**



* + 1. **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.

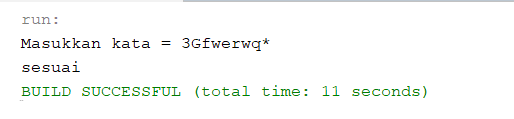
* 1. **Soal Nomor 20**
     1. **Soal**

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

* + 1. **Capture Code**



* + 1. **Capture Output**



* + 1. **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.

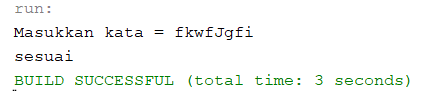
* 1. **Soal Nomor 21**
     1. **Soal**

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

* + 1. **Capture Code**



* + 1. **Capture Output**

****

* + 1. **Analisis**

Meminta user untuk menginput kata dan menyimpannya ke dalam “masukan2”. Mencocokkan apakah “masukan2” 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.

* 1. **Soal Nomor 22**
     1. **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 :

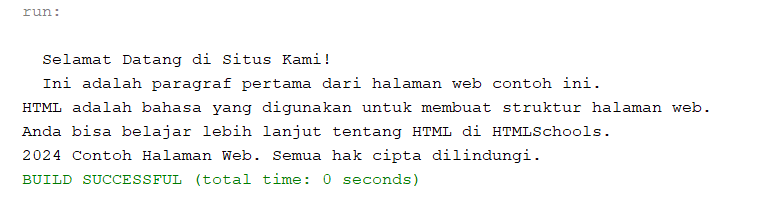
Selamat Datang di Situs Kami!

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

* + 1. **Capture Code**



* + 1. **Capture Output**



* + 1. **Analisis**

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

1. **DAFTAR PUSTAKA**