**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN**

**Pertemuan Ke – 5**



**DISUSUN OLEH :**

**HELDA LUDYA SAFITRI**

**175410186**

**TEKNIK INFORMATIKA**

**STMIK AKAKOM**

**YOGYAKARTA**

**2017**

**PERTEMUAN KE-5**

**THE STRING CLASS, THE RANDOM CLASS AND THE MATH CLASS**

1. **TUJUAN**
2. Mahasiswa dapat menggunakan method yang ada pada kelas String.
3. Mahasiswa dapat menjelaskan dan menggunakan penggunakan bilangan acak pada program.
4. Mahasiswa dapat menjelaskan dan menggunakan metode pada kelas Math.
5. **DASAR TEORI**

String Class merupakan salah satu kelas dasar yang disediakan oleh Java untuk manipulasi karakter. String Class berfungsi untuk mendefinisikan string yang constant (tidak bisa berubah). Ada beberapa String Class pada Java yang fungsinya disesuaikan dengan program yang telah dibuat.

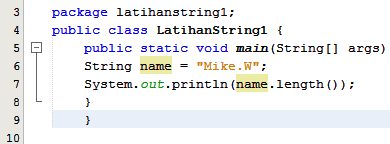
Random class merupakan salah satu kelas dasar yang disediakan oleh Java, seperti String Class, tetapi random class membutuhkan instansiasi.

Math class merupakan salah satu kelas yang disediakan Java, berfungsi untuk menghitung operasi aritmatika.

1. **PEMBAHASAN PRAKTIK**

**Praktik 1 – String Class Methods**

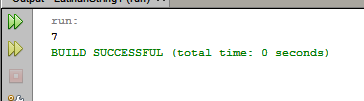
1. Menggunakan metode length()



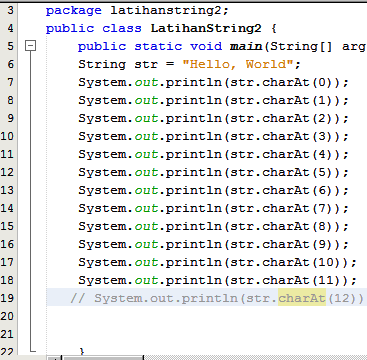
* Baris 4 merupakan nama class dari program yang dibuat yaitu LatihanString1.
* Baris 5 merupakan method yang digunakan untuk mengawali pembuatan program,dimana class method tersebut bisa dipanggil baik di dalam maupun di luar class.
* Baris 6 merupakan sebuah variable dengan nama name dan tipe data String.
* Baris 7berfungsi untuk menampilkan variable name yang ada pada baris sebelumnya, dimana:

**name.length()**merupakan method yang digunakan untuk menghitung jumlah karekter nilai variable name dari tipe data String.

Hasil Output :



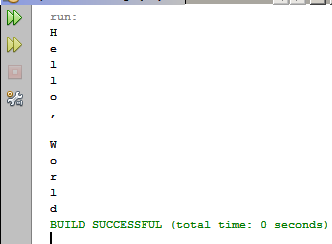
1. Mengakses tiap karakter pada sebuah String



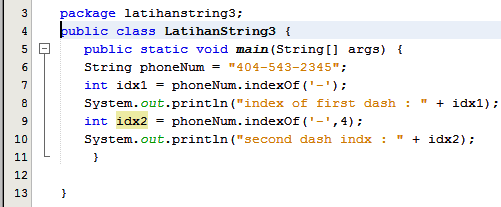
* Baris 4 merupakan nama class dari program yang dibuat yaitu LatihanString2.
* Baris 5 merupakan method yang digunakan untuk mengawali pembuatan program,dimana class method ini bisa dipanggil baik di dalam maupun di luar class.
* Baris 6 merupakan sebuah variable dengan namastr dan tipe data String.
* Baris 7-19 digunakan untuk menampilkan variable str untuk index ke-0 sampai ke-12 dimana :

**Str.charAt()**berfungsi untuk mengakses karakter di dalam tipe data String pada index masing-masing nilai dari variable. Misal Str.charAt(0) yaitu huruf “**h**” karena no index dimulai dari 0 (nol).

Hasil Output :



1. Mengambil nilai index dari karakter padaString.



* Baris 4 merupakan nama class dari program yang dibuat yaitu LatihanString3.
* Baris 5 merupakan method yang digunakan untuk mengawali pembuatan program dimana class method ini bisa dipanggil baik di dalam maupun di luar class.
* Baris 6 merupakan sebuah variable dengan nama phoneNum dan tipe datanya String.
* Baris 7 digunakan untuk mencari urutan dari (‘-‘) pada nilai dari variable phoneNum dimana :

**int idx1=phoneNum.indexOf(‘-‘)**artinya kita akan mencari urutan/index keberapa tanda ini (‘-‘) dari konstanta yang ada pada variable tersebut.

* Baris 8berfungsi untuk menampilkan variable idx1.
* Baris 9 sama dengan baris ke 7 hanya saja index yang dicari mulai dari urutan ke 4,

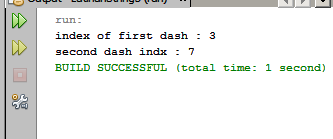
dimana :

**int idx2=phoneNum.indexOf(‘-‘,4)**artinya mencari urutan/index keberapa dari

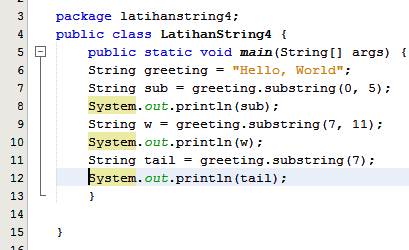
(‘-‘) dan mulai di hitung dari urutan index ke-4.

* Baris 10 digunakan untuk menampilkan variable idx2.

Hasil Output :



1. Mengganti substring dari sebuah String



* Baris 4 merupakan nama class dari program yang dibuat yaitu LatihanString4.
* Baris 5 merupakan method yang digunakan untuk mengawali pembuatan program dimana class method ini bisa dipanggil baik di dalam maupun di luar class.
* Baris 6 merupakan sebuah variable dengan nama greeting dan tipe data String.
* Baris 7,9 dan 11 digunakan untuk memecah karakter pada nilai variable greetingdimana :

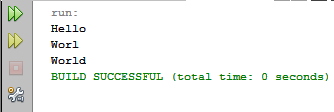
**String sub=greeting substring(0,5)** digunakan untuk memecah string dari index ke-0 sampai index ke-5.

**String w=greeting substring(7,11)** digunakan untuk memecah string dari index ke-7 sampai index ke-11.

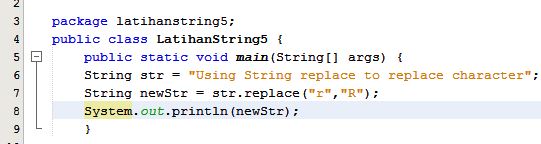
**String tail=greeting substring(7)** digunakan untuk memecah string dari index ke-7 sampai index terakhir.

* Baris 8,10 dan 12 digunakan untuk menampilkan variable dari masing-masing baris yaitu sub,w dan tail.

Hasil Output :



1. Mengganti karakter pada String

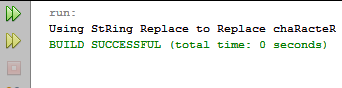


* Baris 4 merupakan nama class dari program yang dibuat yaitu LatihanString5.
* Baris 5 merupakan method yang digunakan untuk mengawali pembuatan program dimana class method ini bisa dipanggil baik di dalam maupun di luar class.
* Baris 6 merupakan sebuah variable dengan nama str dan tipe datanya String.
* Baris 7 digunakan untuk mengganti model karakter pada nilai variable str dimana :

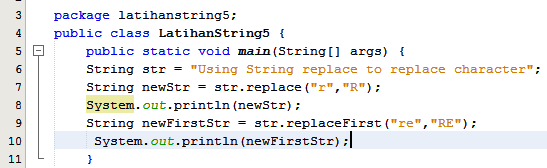
**String newStr=str.replace(“r”,”R”)** ini digunakan untuk mengganti semua huruf “r” menjadi huruf “R”.

* Baris 5 digunakan untuk menampilkan variable newStr.

Hasil Output :



1. Metode dengan parameter : mengganti tipmenjadi 10%

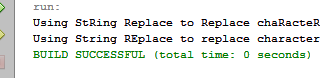


* Baris 4 merupakan nama class dari program yang dibuat yaitu LatihanString5.
* Baris 5 merupakan method yang digunakan untuk mengawali pembuatan program dimana class method ini bisa dipanggil baik di dalam maupun di luar class.
* Baris 9 digunakan untuk mengganti karakter yang ada di awal pernyataan, dimana:

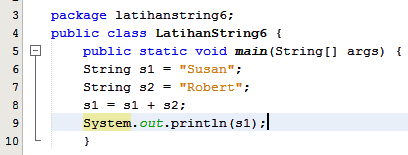
**String newFirstStr=str.replaceFirst(“re”,”RE”)** ini digunakan untuk mengganti karakter (“re”) menjadi (“RE”) yang diganti hanya karakter yang ada di awal pernyataan saja.

* Baris 7 digunakan untuk menampilkan variable newFirstStr.

Hasil Output :

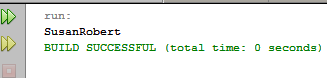


1. Mendeklarasikan dan membuat sebuah String

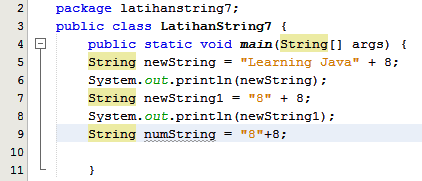


* Baris 6 merupakan sebuah variable dengan nama s1 dan tipe dataString.
* Baris 7 merupakan sebuah variable dengan nama s2 dan type datanya String**.**
* Baris 8 digunakan untuk menggabungkan variable konstanta s1 dengan s2didalam variable s1.
* Baris 9 digunakan untuk menampilkan variable s1.

Hasil Output :



1. Penggabungan String/Concetenating



* Baris 5 dan 7 merupakan penggabungan konstanta dari masing-masing variable dengan angka 8 dimana :

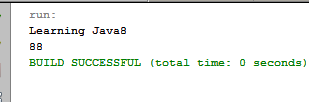
**String newString=”Learning java” +8;** artinya kata “Learning java” akan digabung dengan kata “8”

**String newString1=”8” +8;**artinya kata “8” akan digabung dengan kata “8” .

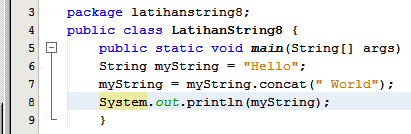
Tanda **+** disini berfungsi sebagai penghubung antara karakter 1 dengan yang lain.

* Baris ke 4-6 digunakan untuk menampilkan variable masing-masing yaitu newStringdan newString1.

Hasil Output :



1. Menggunakan metode concat()



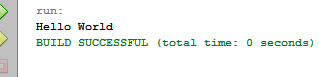
* Baris 6 merupakan sebuah variable dengan nama myString dan tipe datanya String.
* Baris 7 digunakan untuk menggabungkan variable myString dengan kata “World”

dimana :

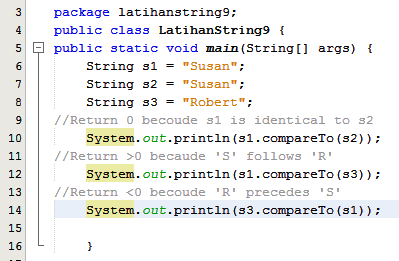
**concat** merupakan metode yang digunakan untuk menggabungkan karakter yang 1 dengan yang lainnya.

* Baris 8 digunakan untuk menampilkan variable myString.

Hasil Output :



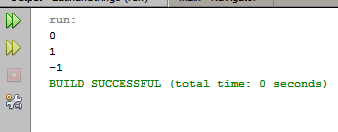
1. Metode compareTo()



* Baris 6-8 merupakan sebuah variable dengan nama s1,s2,s3 dan tipe datanya String.
* Baris 10,12 dan 14 digunakan untuk menampilakan hasil perbandingan antar String dimana:

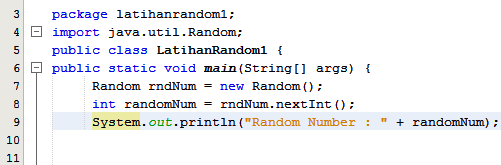
**compareTo** digunakan untuk membandingkan String dengan object lain dengan ketentuan akan bernilai **0** jika yang dibandingkan sama,akan bernilai **>0** jika yang dibandingkan nilainya lebih besar dengan object,dan akan bernilai **<0** jika yang dibandingkan nilainya lebih kecil dengan object.

Hasil Output :



**Praktik2 – Kelas Random**

1. Menggunakan bilangan acak bertipe integer



* Baris 4 merupakan class dari random, dimana :

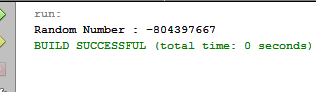
**import java.util.Random** digunakan untuk membuat sebuah nilai acak dari tipe data yang dimasukkan.

* Baris 7 merupakan sebuah keyword untuk memanggil fungsi dari class lain yaitu new Random(); atau membuat object baru dari kelas yang telah dibuat.
* Baris 8 digunakan untuk memasukkan nilai secara random atau acak, dimana:

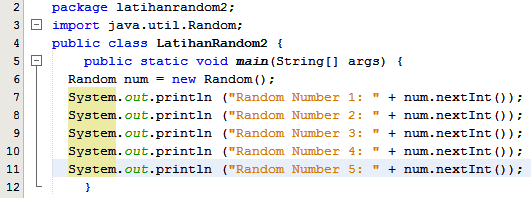
**int randomNum= rndNum.nextInt();** berfungsi untuk memasukkan angka bertipe integer secara acak.

* Baris 9 digunakan untuk menampilkan hasil bilangan acak atau menampilkan variable randomNum.

Hasil Output :

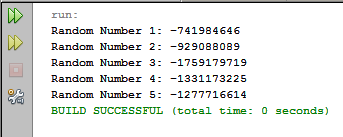


1. Membuat sejumlah bilangan acak

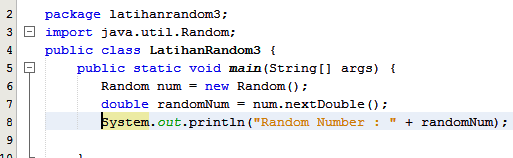


* Baris ke 7-11 digunakan untuk menampilkan nilai acak dengan tipe data integer tetapi angka tiap baris berbeda karena menggunakan class Random.

Hasil Output :



1. Menggunakan bilangan acak bertipe double

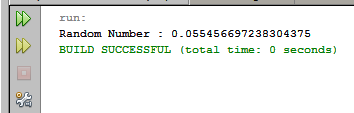


* Baris 6 merupakan sebuah keyword untuk memanggil fungsi dari class lain yaitu new Random(); atau membuat object baru dari kelas yang telah dibuat.
* Baris 7 digunakan untuk memasukkan nilai secara random atau acak dimana:

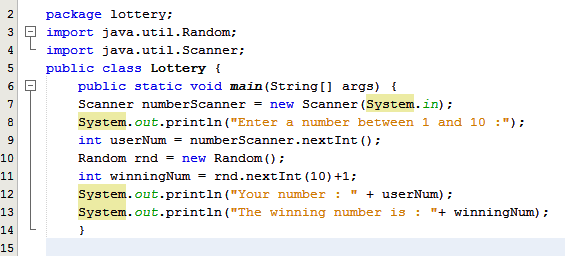
**double randomNum= num.nextDouble();**berfungsi untuk memasukkan angka bertipe double(pecahan/desimal) secara acak.

* Baris 8 digunakan untuk menampilkan hasil dari nilai acak double atau dalam variable randomNum.

Hasil Output :



1. Membuat bilangan acak dengan range tertentu



* Baris 4 merupakan class dari scanner

**import java.util.Scanner;** digunakan untuk melakukan penginputan data ketika program sedang dijalankan.

* Baris 7 merupakan sebuah keyword untuk memanggil fungsi dari class lain yaitu new Scanner(); atau membuat object baru dari kelas yang telah dibuat.
* Baris 8 digunakan untuk menampilkan pernyataan “Enter a number between 1 and 10”
* Baris 9 digunakan untuk memasukkan data bertipe integer dengan range 1-10,dimana:

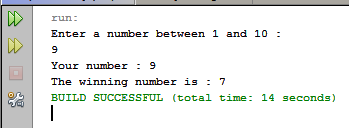
**int userNum = numberScanner.nextInt(10)+1;** artinya kita hanya bisa memasukkan angka dengan ketentuan tidak kurang dari 1 dan tidak lebih dari 10.

* Baris 10 merupakan sebuah keyword untuk memanggil fungsi dari class lain yaitu new Random(); atau membuat object baru dari kelas yang telah dibuat.
* Baris 11 digunakan untuk memasukkan bilangan acak dengan range 1-10

**int winningNum = md.nextInt(10)+1;** artinya program akan memasukkan angka dengan ketentuan tidak kurang dari 1 dan tidak lebih dari 10.

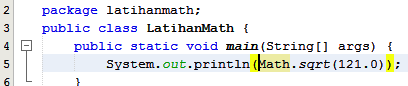
* Baris 12-13akan memanpilkan hasil dari variable userNum dan winningNum.

Hasil Output :



**Praktik 3**

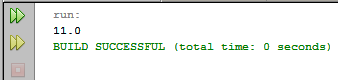
1. Menggunakan metode sqrt

****

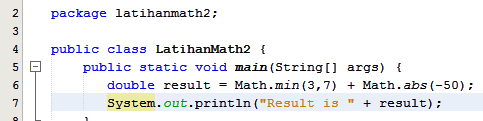
* Baris 6 digunakan untuk menampilkan akar dari 121.0 dimana :

**Math.sqrt** merupakan sebuah Math class yang berfungsi untuk melakukan proses pengakaran aritmatika dari bilangan yang kita masukkan.

Hasil Output :



1. Menggunakan metode min dan abs



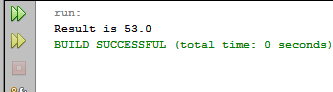
* Baris 6 digunakan untuk menghitung operasi penjumlahan dari (3,7) dan (-50) dimana :

**Math.min** merupakan sebuah Math class yang berfungsi untuk menghitung nilai minimal dari bilangan yang kita masukkan yaitu (3,7).

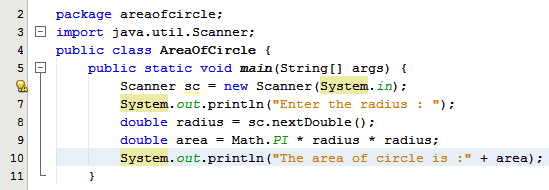
**Math.abs** merupakan sebuah Math class yang berfungsi untuk mengubah nilai menjadi positif dari bilangan yang kita masukkan yaitu (-50)

* Baris 7 digunakan untuk menampilkan hasil dari variable result.

Hasil Output :



1. Menggunakan field PI

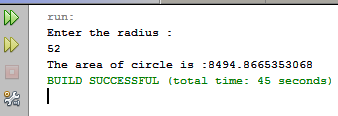


* Baris 7 digunakan untuk menampilkan pernyataan “Enter the radius :”
* Baris 8digunakan untuk memasukkan angka bertipe double(desimal).
* Baris 9 digunakan untuk menghitung nilai dari variable area dimana :

**Math.PI** merupakan sebuah Math class yang memiliki nilai (3.14) atau sebuah operasi yang digunakan untuk menghitung lingkaran.

* Baris 10 untuk menampilkan hasil dari variable area.

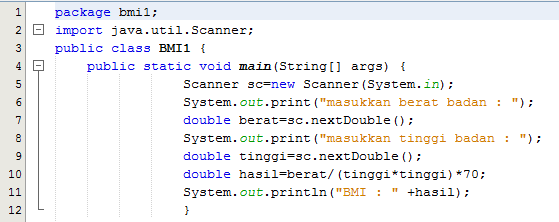
Hasil Output :



1. **LATIHAN**

|  |  |
| --- | --- |
| Modifier and Type | Method and Description |
| char | charAt (int index)  Mengembalikan nilai char pada indeks yang ditentukan. |
| int | codePointAt(int index)  Mengembalikan karakter (kode Unicode point) pada indeks yang ditentukan. |
| int | codePointBefore(int index)  Mengembalikan karakter (kode Unicode point) sebelum indeks yang ditentukan. |
| Int | codePointCount(int beginIndex, int endIndex)  Mengembalikan jumlah kode kode Unicode dalam rentang teks yang ditentukan dari String ini. |
| int | compareTo(String anotherString)  Membandingkan dua string secara leksikografis. |
| int | compareToIgnoreCase(String str)  Membandingkan dua string leksikografis, mengabaikan perbedaan kasus. |
| String | concat (String str)  Menyatukan string yang ditentukan sampai akhir string ini. |
| boolean | contains(CharSequence s)  Mengembalikan true jika dan hanya jika string ini berisi urutan nilai char yang ditentukan. |
| boolean | contentEquals(CharSequence cs)  Membandingkan string ini dengan CharSequence yang ditentukan. |
| boolean | contentEquals(StringBuffer sb)  Membandingkan string ini dengan StringBuffer yang ditentukan. |
| static String | copyValueOf(char[] data)  Setara dengan valueOf (char []). |
| static String | copyValueOf(char[] data, int offset, int count)  Setara dengan valueOf (char [], int, int). |
| boolean | endsWith(String suffix)  Pengujian jika string ini diakhiri dengan akhiran yang ditentukan. |
| boolean | equals(Object anObject)  Membandingkan string ini dengan objek yang ditentukan. |
| boolean | equalsIgnoreCase(String anotherString)  Membandingkan String ini dengan String lain, mengabaikan pertimbangan kasus. |
| static String | format(Locale l, String format, Object... args)  Mengembalikan string berformat menggunakan lokal yang ditentukan, string format, dan argumen. |
| static String | format(String format, Object... args)  Mengembalikan string yang diformat menggunakan string format dan argumen yang ditentukan. |
| byte[] | getBytes()  Encode String ini menjadi urutan byte menggunakan charset default platform, menyimpan hasilnya ke array byte baru. |
| byte[] | getBytes(Charset charset)  Encode String ini menjadi urutan byte menggunakan charset yang diberikan, menyimpan hasilnya ke array byte baru. |
| Void | getBytes(int srcBegin, int srcEnd, byte[] dst, int dstBegin)  Tidak berlaku lagi  Metode ini tidak mengubah karakter menjadi byte. Seperti JDK 1.1, cara yang lebih disukai untuk melakukan ini adalah melalui metode getBytes (), yang menggunakan charset bawaan platform. |
| byte[] | getBytes(String charsetName)  Encode String ini menjadi urutan byte menggunakan charset bernama, menyimpan hasilnya ke array byte baru. |
| void | getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin)  Salin karakter dari string ini ke dalam array karakter tujuan. |
| int | hashCode()  Mengembalikan kode hash untuk string ini. |
| int | indexOf(int ch)  Mengembalikan indeks dalam string ini dari kejadian pertama dari karakter yang ditentukan. |
| int | indexOf(int ch, int fromIndex)  Mengembalikan indeks dalam string ini dari kejadian pertama karakter yang ditentukan, memulai pencarian pada indeks yang ditentukan. |
| int | indexOf(String str)  Mengembalikan indeks dalam string ini dari kejadian pertama dari substring yang ditentukan. |
| int | indexOf(String str, int fromIndex)  Mengembalikan indeks dalam string ini dari kejadian pertama dari substring yang ditentukan, dimulai dari indeks yang ditentukan. |
| String | intern()  Mengembalikan representasi kanonik untuk objek string. |
| boolean | isEmpty()  Mengembalikan true if, dan hanya jika, length () adalah 0. |
| static String | join(CharSequence delimiter, CharSequence... elements)  Mengembalikan String baru yang terdiri dari salinan elemen CharSequence yang digabungkan bersama dengan salinan pembatas yang ditentukan. |
| static String | join(CharSequence delimiter, Iterable<? extends CharSequence> elements)  Mengembalikan String baru yang terdiri dari salinan elemen CharSequence yang digabungkan bersama dengan salinan pembatas yang ditentukan. |
| int | lastIndexOf(int ch)  Mengembalikan indeks dalam string ini dari kejadian terakhir dari karakter yang ditentukan. |
| int | lastIndexOf(int ch, int fromIndex)  Mengembalikan indeks dalam string ini dari kejadian terakhir dari karakter yang ditentukan, mencari mundur mulai dari indeks yang ditentukan. |
| int | lastIndexOf(String str)  Mengembalikan indeks dalam string ini dari kejadian terakhir dari substring yang ditentukan. |
| int | lastIndexOf(String str, int fromIndex)  Mengembalikan indeks dalam string ini dari kejadian terakhir substring yang ditentukan, mencari mundur mulai dari indeks yang ditentukan. |
| int | length()  Mengembalikan panjang string ini. |
| boolean | matches(String regex)  Memberi tahu apakah string ini cocok dengan ekspresi reguler yang diberikan. |
| int | offsetByCodePoints(int index, int codePointOffset)  Mengembalikan indeks dalam String ini yang diimbangi dari indeks yang diberikan oleh kode codePointOffset. |
| boolean | regionMatches(boolean ignoreCase, int toffset, String other, int ooffset, int len)  Pengujian jika dua daerah string sama. |
| boolean | regionMatches(int toffset, String other, int ooffset, int len)  Pengujian jika dua daerah string sama. |
| String | replace(char oldChar, char newChar)  Mengembalikan string yang dihasilkan dari mengganti semua kejadian oldChar dalam string ini dengan newChar. |
| String | replace(CharSequence target, CharSequence replacement)  Mengganti setiap substring dari string ini yang sesuai dengan urutan target literal dengan urutan pengganti literal yang ditentukan. |
| String | replaceAll(String regex, String replacement)  Mengganti setiap substring string ini yang sesuai dengan ekspresi reguler yang diberikan dengan penggantian yang diberikan. |
| String | replaceFirst(String regex, String replacement)  Menggantikan substring pertama dari string ini yang sesuai dengan ekspresi reguler yang diberikan dengan penggantian yang diberikan. |
| String[] | split(String regex)  Perpecahan string ini di sekitar pertandingan dari ekspresi reguler yang diberikan. |
| String[] | split(String regex, int limit)  Perpecahan string ini di sekitar pertandingan dari ekspresi reguler yang diberikan. |
| boolean | startsWith(String prefix)  Pengujian jika string ini dimulai dengan awalan yang ditentukan. |
| boolean | startsWith(String prefix, int toffset)  Pengujian jika substring string ini dimulai pada indeks yang ditentukan dimulai dengan awalan yang ditentukan. |
| CharSequence | subSequence(int beginIndex, int endIndex)  Mengembalikan urutan karakter yang merupakan urutan urutan ini. |
| String | substring(int beginIndex)  Mengembalikan string yang merupakan substring dari string ini. |
| String | substring(int beginIndex, int endIndex)  Mengembalikan string yang merupakan substring dari string ini. |
| char[] | toCharArray()  Mengonversi string ini ke array karakter baru. |
| String | toLowerCase()  Mengubah semua karakter dalam String ini menjadi huruf kecil dengan menggunakan aturan lokal default. |
| String | toLowerCase(Locale locale)  Mengubah semua karakter dalam String ini menjadi huruf kecil dengan menggunakan aturan Locale yang diberikan. |
| String | toString()  Objek ini (yang sudah berupa string!) Itu sendiri dikembalikan. |
| String | toUpperCase()  Mengonversi semua karakter dalam String ini ke huruf besar menggunakan aturan lokal default. |
| String | toUpperCase(Locale locale)  Mengkonversi semua karakter dalam String ke huruf besar menggunakan aturan Locale yang diberikan. |
| String | trim()  Mengembalikan string yang nilainya adalah string ini, dengan spasi terdepan dan trailing dihapus. |
| static String | valueOf(boolean b)  Mengembalikan representasi string dari argumen boolean. |
| static String | valueOf(char c)  Mengembalikan representasi string dari argumen char. |
| static String | valueOf(char[] data)  Mengembalikan representasi string argumen char array. |
| static String | valueOf(char[] data, int offset, int count)  Mengembalikan representasi string dari subarray tertentu dari argumen char array. |
| static String | valueOf(double d)  Mengembalikan representasi string dari argumen ganda. |
| static String | valueOf(float f)  Mengembalikan representasi string dari argumen float. |
| static String | valueOf(int i)  Mengembalikan representasi string argumen int. |
| static String | valueOf(long l)  Mengembalikan representasi string dari argumen panjang. |
| static String | valueOf(Object obj)  Mengembalikan representasi string dari argumen Objek. |

1. **TUGAS**

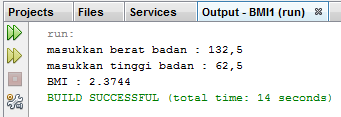
****

* Program untuk menghitug BMI
* Baris 2 merupakan class dari scanner

**import java.util.Scanner;** digunakan untuk melakukan input data ketika program sedangdijalankan.

* Baris 3 merupakan nama class dari program yang kita buat yaitu BMI1.
* Baris 4 merupakan method yang digunakan untuk mengawali pembuatan program.
* Baris 5 merupakan sebuah keyword untuk memanggil fungsi dari class lain yaitu newScanner(); atau membuat object baru dari kelas yang telah kita buat.
* Baris ke 7 dan 9 digunakan untuk memasukkan data yang bertipe double (desimal) ketika program sedang dijalankan.
* Baris 10 digunakan untuk melakukan proses aritmatika dimana **berat/(tinggi\*tinggi)\*70**
* Baris 11 untuk menampilkan hasil dari BMI.

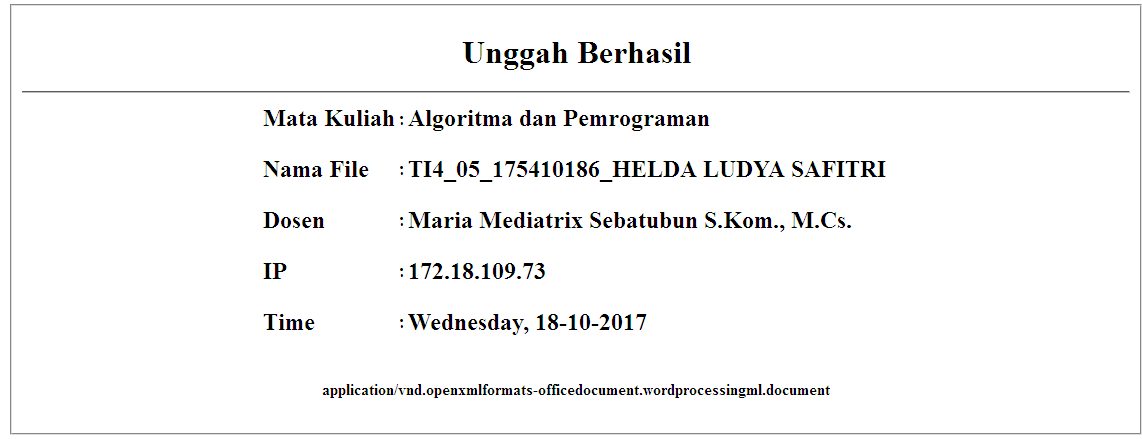
Hasil Output :



1. **KESIMPULAN**

Dari pembahasan praktikum diatas dapat disimpulkan bahwa ada aturan-aturan tertentu dalam penggunaan method pada class String, class Math, maupun dalam perhitungan bilangan acak. Sehingga aturan-aturan tersebut harus diperhatikan dengan baik agar tidak terjadi kesalahan pada program yang dibuat.

1. **LISTING**

****