**Link github :**

**https://github.com/tafhdytllah/test**

SOAL PRAKTEK JUNIOR PROGRAMMER

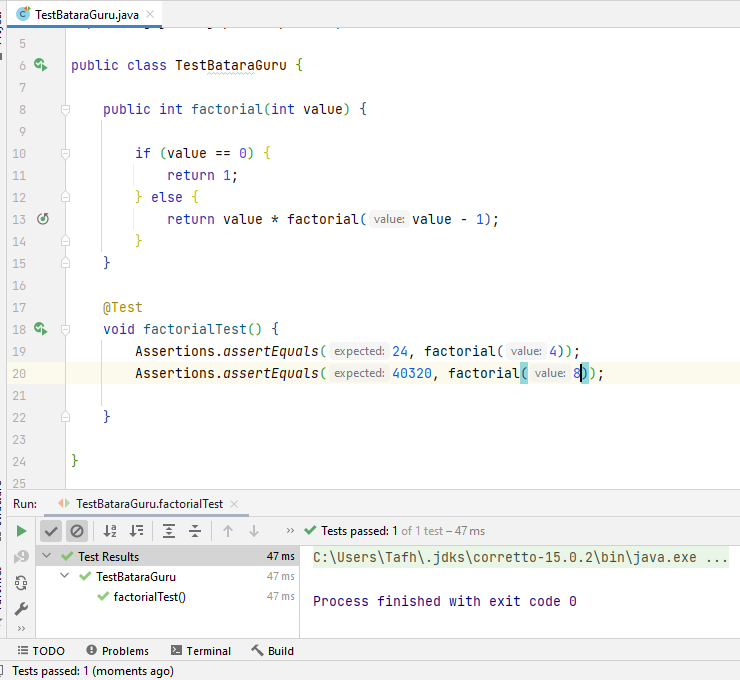
1. Oop (objek oriented programming) adalah konsep pemrograman berorientasi obek yg terdiri dari objek, class, inheriten / turunan, abstraction, encapsulation, polymorpysm

Bahasa yg berorientasi objek : Java, Kotlin, dart, Php, javascript

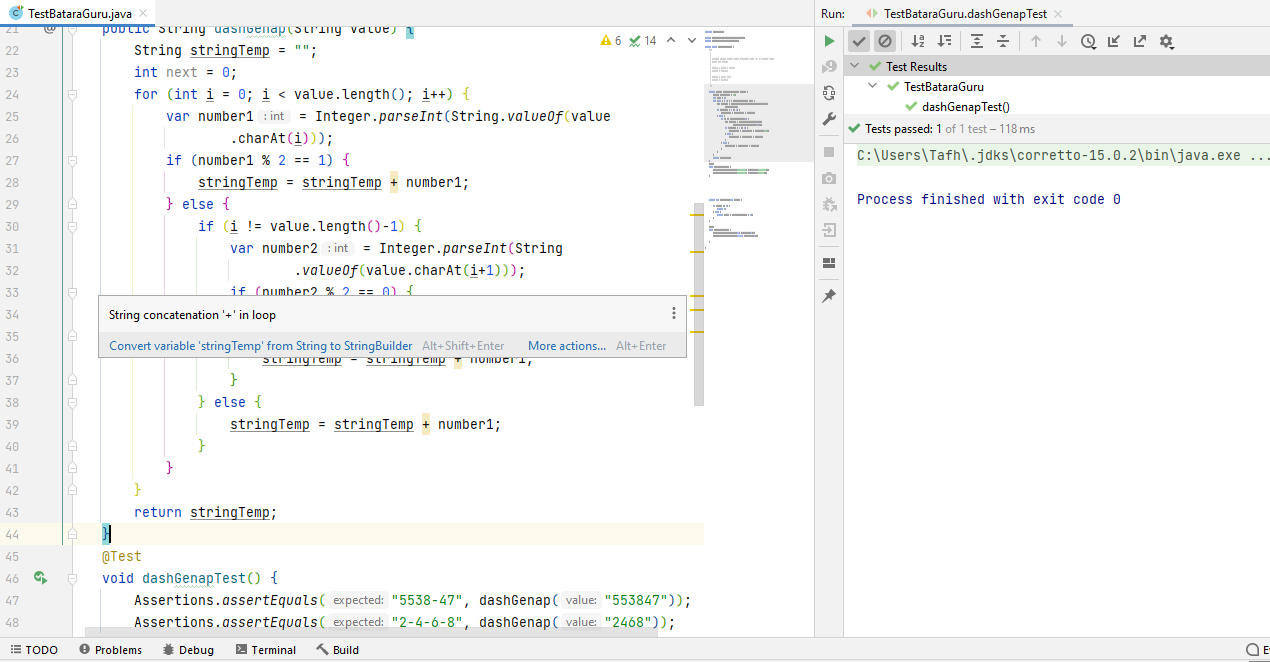
1. Kegunaan framework untuk memudahkan programmer dalam mendevelop aplikasi atau sebuah program yang akan dibuat, sebenarnya bisa juga tanpa framework, akan tetapi waktu pengerjaan akan semakin lama jika membuat project besar. Karena di dalam framework ada banyak tools2 yang sudah disiapkan untuk mempermudah dan mempercepat programmer hanya tinggal menggunakannya, contoh framework dalam pembuatan aplikasi mobile ada Flutter dan React Native.

Setiap Bahasa memiliki framework karena programmer Bahasa itu sendiri yang membuat dengan tujuan untuk mempermudah mereka dalam mendevelop sebuah aplikasi menggunakan Bahasa yng mereka gunakan.

1. package bataraguru;  
     
   import org.junit.jupiter.api.Assertions;  
   import org.junit.jupiter.api.Test;  
     
   public class TestBataraGuru *{* public int factorial*(*int value*) {* if *(*value == 0*) {* return 1;  
    *}* else *{* return value \* factorial*(*value - 1*)*;  
    *}  
    }* @Test  
    void factorialTest*() {* Assertions.*assertEquals(*24, factorial*(*4*))*;  
    Assertions.*assertEquals(*40320, factorial*(*8*))*;  
     
    *}  
     
   }*



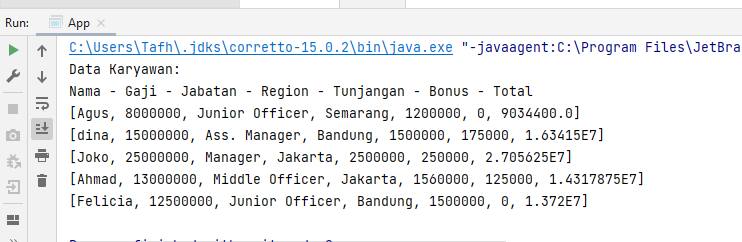
1. public String dashGenap*(*String value*) {* String stringTemp = "";  
    int next = 0;  
    for *(*int i = 0; i < value.length*()*; i++*) {* var number1 = Integer.*parseInt(*String.*valueOf(*value  
    .charAt*(*i*)))*;  
    if *(*number1 % 2 == 1*) {* stringTemp = stringTemp + number1;  
    *}* else *{* if *(*i != value.length*()*-1*) {* var number2 = Integer.*parseInt(*String  
    .*valueOf(*value.charAt*(*i+1*)))*;  
    if *(*number2 % 2 == 0*) {* stringTemp = stringTemp + *(*number1+"-"*)*;  
    *}* else *{* stringTemp = stringTemp + number1;  
    *}  
    }* else *{* stringTemp = stringTemp + number1;  
    *}  
    }  
    }* return stringTemp;  
   *}*@Test  
   void dashGenapTest*() {* Assertions.*assertEquals(*"5538-47", dashGenap*(*"553847"*))*;  
    Assertions.*assertEquals(*"2-4-6-8", dashGenap*(*"2468"*))*;  
   *}*



1. –

**Link github :**

**https://github.com/tafhdytllah/test**



Logic

A.

if gaji >= 15.000.000

tunjangan = gaji \* 10 / 100

else if gaji >= 10.000.000

tunjangan = gaji \* 12 / 100

else

tunjangan = gaji \* 15 / 100

B.

int total = gaji + tunjangan

if region == Jakarta

total = total - total \* 2.5 / 100

if region == bandung

total = total - total \* 2 / 100

else

total = total - total \* 1.8 / 100

int bonus

manager = 250\_000

ass. manager = 175\_000

senior officer = 150\_000

middle officer = 125\_000

junior officer = 100\_000

Source Code

package com.example.app;  
  
import java.sql.DriverManager;  
import java.sql.Connection;  
import java.sql.Statement;  
import java.sql.ResultSet;  
import java.util.ArrayList;  
  
public class App *{* static final String *JDBC\_DRIVER* = "com.mysql.jdbc.Driver";  
 static final String *DB\_URL* = "jdbc:mysql://localhost/test";  
 static final String *USER* = "root";  
 static final String *PASS* = "";  
  
 static Connection *conn*;  
 static Statement *stmt*;  
 static ResultSet *rs*;  
  
 public static void main*(*String*[]* args*) {* try *{* Class.*forName(JDBC\_DRIVER)*;  
  
 *conn* = DriverManager.*getConnection(DB\_URL*, *USER*, *PASS)*;  
  
 *stmt* = *conn*.createStatement*()*;  
  
 String sql = "SELECT \* FROM karyawan";  
  
 *rs* = *stmt*.executeQuery*(*sql*)*;  
  
  
 ArrayList*<*ArrayList*<*String*>>* karyawan = new ArrayList*<*ArrayList*<*String*>>()*;  
  
 while *(rs*.next*()) {* var nama = *rs*.getString*(*"nama"*)*;  
 var gaji = *rs*.getInt*(*"gaji"*)*;  
 var jabatan = *rs*.getString*(*"jabatan"*)*;  
 var region = *rs*.getString*(*"region"*)*;  
  
 */\*\*  
 \* Soal 6 A  
 \*  
 \* if gaji >= 15.000.000  
 \* tunjangan = gaji \* 10 / 100  
 \* else if gaji >= 10.000.000  
 \* tunjangan = gaji \* 12 / 100  
 \* else  
 \* tunjangan = gaji \* 15 / 100  
 \*/* int tunjangan;  
 if *(*gaji >= 15\_000\_000*) {* tunjangan = gaji \* 10 / 100;  
 *}* else if *(*gaji >= 10\_000\_000*) {* tunjangan = gaji \* 12 / 100;  
 *}* else *{* tunjangan = gaji \* 15 / 100;  
 *}  
  
 /\*\*  
 \* Soal 6 B  
 \*  
 \* int total = gaji + tunjangan  
 \*  
 \* if region == Jakarta  
 \* total = total - total \* 2.5 / 100  
 \* if region == bandung  
 \* total = total - total \* 2 / 100  
 \* else  
 \* total = total - total \* 1.8 / 100  
 \*  
 \* int bonus  
 \*  
 \* manager = 250\_000  
 \* ass. manager = 175\_000  
 \* senior officer = 150\_000  
 \* middle officer = 125\_000  
 \* junior officer = 100\_000  
 \*/* int bonus;  
 switch *(*jabatan*) {* case "Manager":  
 bonus = 250\_000;  
 break;  
 case "Ass. Manager":  
 bonus = 175\_000;  
 break;  
 case "Senior Officer":  
 bonus = 150\_000;  
 break;  
 case "Middle Officer":  
 bonus = 125\_000;  
 break;  
 case "Junior Office":  
 bonus = 100\_000;  
 break;  
 default:  
 bonus = 0;  
 *}* double total = gaji + tunjangan + bonus;  
  
 switch *(*region*) {* case "Jakarta":  
 total = total - *(*total \* 2.5 / 100*)*;  
 break;  
 case "Bandung":  
 total = total - *(*total \* 2 / 100*)*;  
 break;  
 default:  
 total = total - *(*total \* 1.8 / 100*)*;  
  
 *}* ArrayList*<*String*>* inner = new ArrayList*<*String*>()*;  
 inner.add*(*nama*)*;  
 inner.add*(*String.*valueOf(*gaji*))*;  
 inner.add*(*jabatan*)*;  
 inner.add*(*region*)*;  
 inner.add*(*String.*valueOf(*tunjangan*))*;  
 inner.add*(*String.*valueOf(*bonus*))*;  
 inner.add*(*String.*valueOf(*total*))*;  
  
 karyawan.add*(*inner*)*;  
  
 *}* System.*out*.println*(*"Data Karyawan:"*)*;  
 System.*out*.println*(*"Nama - Gaji - Jabatan - Region - Tunjangan - Bonus - Total"*)*;  
  
 for *(*int i = 0; i < karyawan.size*()*; i++*) {* System.*out*.println*(*karyawan.get*(*i*))*;  
 *}  
  
  
 stmt*.close*()*;  
 *conn*.close*()*;  
  
 *}* catch *(*Exception e*) {* e.printStackTrace*()*;  
 *}  
 }  
}*