Laporan Praktikum Pemrograman Berbasis Objek

Ke – 1

Modul 1 : Pendahuluan Pemrograman Java



Oleh :

Risyad Pangestu

140810170003

TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

2018

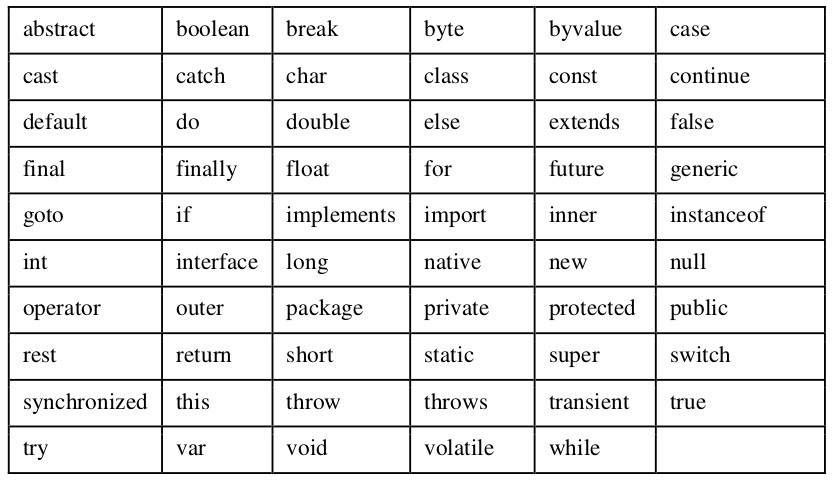
A. Tujuan Pembelajaran

1. Mengenal dan mempersiapkan lingkungan kerja Java
2. Mengenal penamaan identifier dan tipe dasar
3. Membuat, mengkompilasi dan menjalankan program Java
4. Memahami penggunaan berbagai macam jenis operator yang ada di Java
5. Menganalisa beberapa problem yang terjadi saat pemrograman dan
6. memberikan solusi

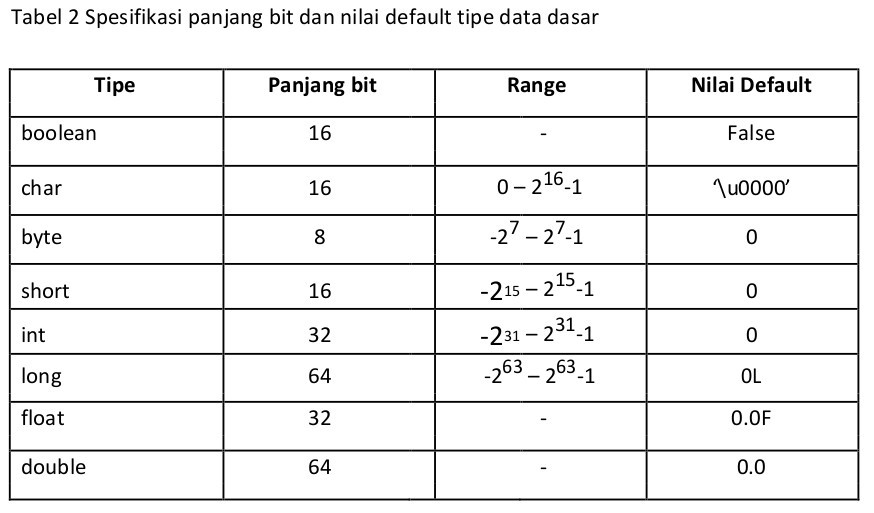
B. Dasar Teori

Bahasa pemograman java merupakan salah satu bahasa yang bersifat open source yang dikeluarkan oleh Sun Microsystem (sekarang bagian dari Oracle). Informasi ditail dapat dilihat pada laman <http://www.oracle.com/technetwork/java/index.html>. Java Development Kit (JDK) memuat Java Runtime Envirotment (JRE) dan perangkat pengembangan(seperti compiler dan debuger). JRE digunakan untuk eksekusi program java. Oleh karena itu, JDK dan JRE memiliki peran penting dalam mengkompilasi dan eksekusi program java.

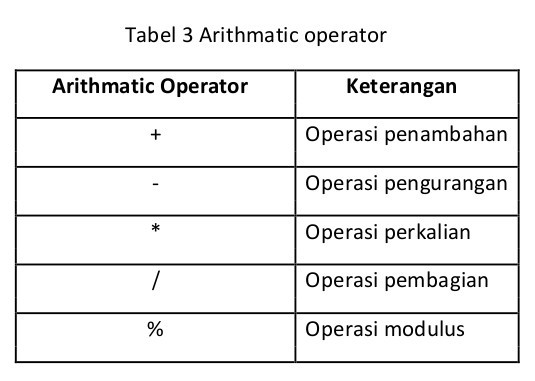
Identifier digunakan untuk memberikan nama variabel, method atau class. Penamaan identifier harus diawali dengan karakter, tanda $ (dollar) atau tanda \_ (underscore). Penamaan identifier ini bersifat case-sensitive dan tidak boleh mengandung spasi atau diawali angka. Java mempunyai 59 kata kunci yang tidak bisa dipakai sebagai identifier termasuk di dalamnya 3 kata literal, yaitu true, false dan null, yang juga tidak bisa dipakai untuk penamaan identifier dan juga terdapat 2 reserved words yang tidak bisa digunakan sebagai nama identifier yaitu *const* dan *goto*.



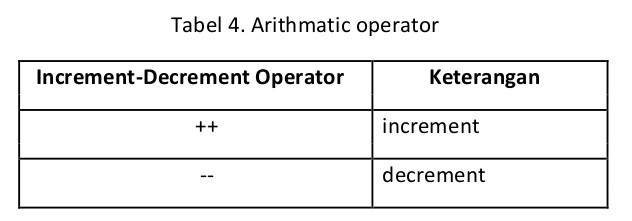
Java mempunyai 8 tipe dasar, yaitu boolean, char, byte, short, int, long, float, dan double. Dari 8 tipe data dasar tersebut bisa dikelompokkan menjadi 4 kelompok data yaitu, integer, floating point, character, dan logical. Yang termasuk dalam kelompok data integer atau bilangan bulat adalah byte, short, int dan long. Nilai default untuk kelompok data integral adalah int. Yang termasuk dalam kelompok data floating point adalah float dan double dengan nilai default double. Sedang yang termasuk dalam tipe data character adalah char yang direpresentasikan dengan Unicode leter. Dan kelompok data yang terakhir adalah logical yaitu boolean dengan 2hanya dua buah nilai saja yang diijinkan yaitu “true” dan “false”. Spesifikasi panjang bit dan range serta nilai default untuk masing-masing tipe data bisa dilihat pada Tabel 2. Casting diperlukan untuk mengkonversi dari suatu tipe ke tipe data yang lebih kecil panjang bitnya. Sedangkan promotion terjadi pada saat mengkonversi dari suatu tipe data ke tipe data yang lebih panjang bitnya.



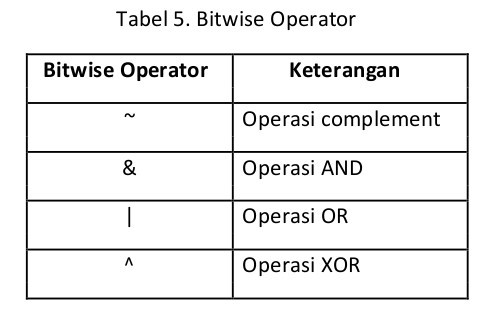
Operator dapat diklasifikasikan menjadi 2 bentuk, yaitu unary operator dan binary operator. Unary operator adalah operator yang hanya melibatkan 1 operan. Sedangkan binary operator adalah operator yang melibatkan 2 operan. Java mempunyai berbagai macam jenis operator yang dapat digolongkan menjadi operator aritmatika, increment-decrement, bitwise, boolean, logik, shift (geser), penugasan, kombinasi dan kondisi. Arithmatic operator (operator aritmatika) adalah operator yang berfungsi untuk operasi aritmatika. Yang termasuk dalam arithmatic operator bisa dilihat pada Tabel 3.



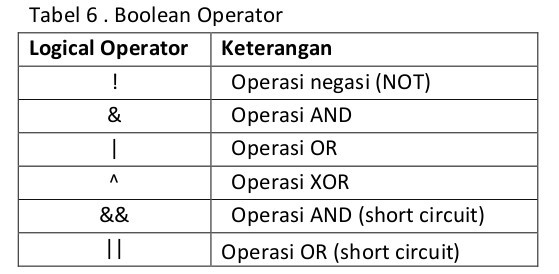
Increment – decrement operator adalah operator yang berguna untuk menaikkan 1 nilai (increment) dan menurunkan 1 nilai (decrement). Yang termasuk increment-decrement operator bisa dilihat pada Tabel 4.



Bitwise operator adalah operator yang dipakai untuk operasi bit pada nilai operan. Yang termasuk bitwise operator bisa dilihat pada Tabel 5.

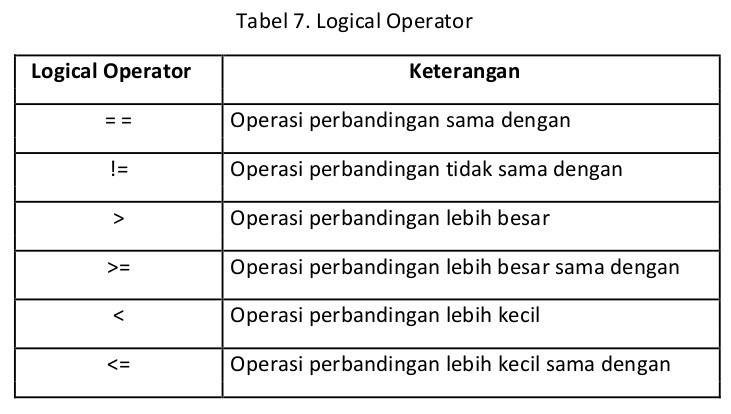


Boolean operator (operator boolean) adalah operator yang mengharuskan operannya bertipe boolean (true atau false). Yang termasuk boolean operator bisa dilihat pada Tabel 6.



Operator !, &, | dan ^ mempunyai implementasi yang sama sebagaimana ketika ia menjadi bitwise operator. Hanya saja di logical operator, operan yang dilibatkan disini harus bertipe boolean, yang hanya mempunyai nilai true atau false.

Logical operator (operator logika) adalah operator yang sering dipakai untuk operasi perbandingan dan selalu menghasilkan suatu nilai bertipe boolean (true atau false). Yang termasuk logical operator bisa dilihat pada Tabel 7.



Combination operator (operator kombinasi) adalah operator yang terdiri dari gabungan 2 operator. Biasanya combination operator ini dipakai untuk mempersingkat waktu penulisan program. Yang termasuk operator combination bisa dilihat pada Tabel 9.

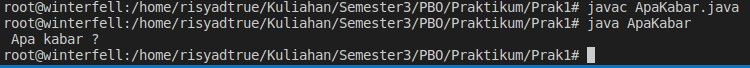
Conditional operator (operator konditional) adalah operator yang dipakai untuk operasi kondisi (persyaratan), sama sebagaimana if-then-else dan hanya berlaku untuk pernyataan tunggal. Operator ini mengembalikan suatu nilai yang benar sesuai dengan kondisi yang diberikan. Conditional operator (operator konditional) ini hanya ada 1 macam, yaitu ? disertai dengan tanda : (titik dua). Jika kondisi persyaratan yang terletak di sebelah kiri tanda ? bernilai benar, maka pernyataan yang berada di sebelah kiri tanda : yang akan diambil. Demikian juga sebaliknya, jika kondisi persyaratan bernilai salah, maka pernyataan yang berada di sebelah kanan tanda : yang akan

diambil.

C. Percobaan

**Percobaan 4 : Menampilkan suatu tulisan ke layar**

Output :

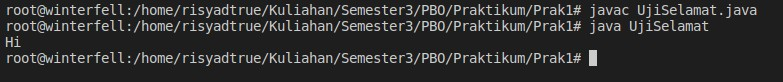


Analisa :

Program tersebut menghasilkan output berupa tulisan.

**Percobaan 5 : Melibatkan class lain dalam program**

Output :



Analisa :

Program dalam dua file yang berbeda dapat di panggil pada salah satu file yang mengandung fungsi main dan memilik objek dalam class yang di panggil.

**Percobaan 6 : Memberikan nilai ke suatu tipe data**

Output :

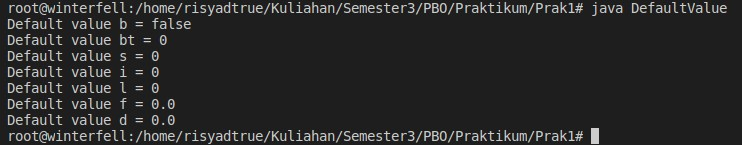


Analisa :

jika ingin memunculkan nilai dalam suatu tipe data di variable dapat dengan menambahkan “ + (variable) “

**Percobaan 7 : Mencetak nilai default dari tipe dasar**

Output :

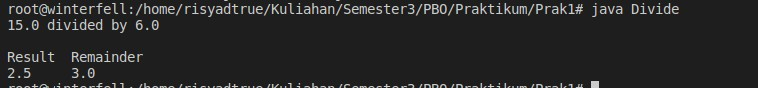


Analisa :

Program tersebut untuk mengetahui nilai default suatu tipe data.

**Percobaan 8 : Mengamati hasil perubahan nilai dari suatu operasi matematis**

Output :

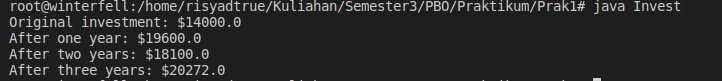


Analisa :

Program untuk menghitung pembagian dan sisa pembagian.

**Percobaan 9 : Mengamati hasil perubahan nilai dari suatu operasi matematis**

Output :

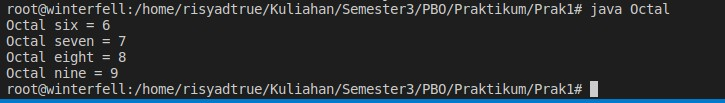


Analisa :

Program tersebut menunjukan nilai dapat disimpan dan di operasi di operasi matematis berikutnya.

**Percobaan 10 : Menampilkan bilangan oktal ke format desimal**

Output :

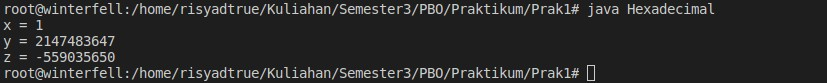


Analisa :

Program tersebut menampilkan nilai decimal dari oktal nilai nilai yang ada.

**Percobaan 11 : Menampilkan bilangan heksadesimal ke format desimal**

Output :



Analisa :

Program tersebut menampilkan bilangan decimal dari bilangan hexadecimal, dengan aturan penulisan yang berbeda.

**Percobaan 12 : Mengamati perubahan nilai pada suatu tipe**

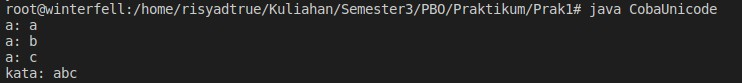
Output :



Analisa :

**Percobaan 13 : Memahami pemakaian Unicode**

Output :

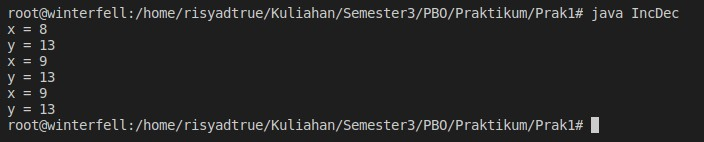


Analisa :

Program tersebut menampilkan nilai unicode dari huruf a ,b ,c.

**Percobaan 14 : Melakukan increment dan decrement nilai**

Output :

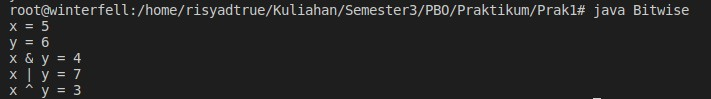


Analisa :

Program tersebut menampilkan hasil dari increment dan decrement

**Percobaan 15 : Melakukan operasi bit**

Output :

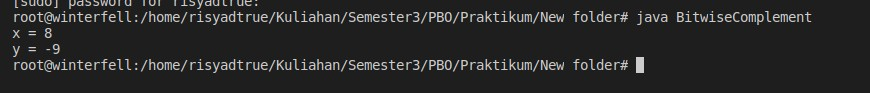


Analisa :

Program tersebut mengoprasikan operasi bit seperti AND, OR dan XOR

**Percobaan 16 : Melakukan operasi komplemen**

Output :

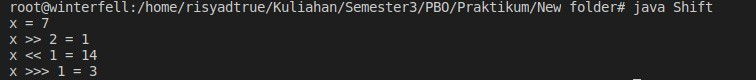


Analisa :

Program tersebut menampilkan hasil operasi bit dengan menggunakan tanda “ ~ ”.

**Percobaan 17 : Melakukan operasi shift**

Output :

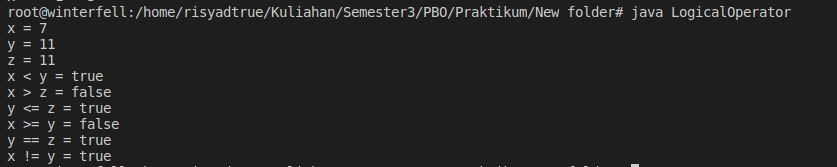


Analisa :

program tersebut melakukan operasi right shift, left shift dan using right shift.

**Percobaan 18 : Menggunakan logical operator**

Output :



Analisa :

Program tersebut menampilkan hail logical operator dari nilai nilai yang ada di dalam variable.

**Percobaan 19 : Menggunakan operator boolean and**

Output :



Analisa :

Hasil dari operasi Boolean And nilai 5 dan 7.

**Percobaan 20 : Menggunakan operator boolean and short-circuit**

Output :



Analisa :

Hasil dari operasi Boolean AND short-circuit

**Percobaan 21 : Menggunakan boolean or**

Output :



Analisa :

Hasil operasi Boolean Or.

**Percobaan 22 : Menggunakan boolean or short-circuit**

Output :



Analisa :

Hasil dari operasi Boolean OR short-circuit

**Percobaan 23 : Menggunakan operator kondisi**

Output :

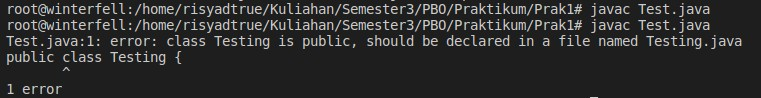


Analisa :

Hasil operasi dari operator conditional.

D. Latihan

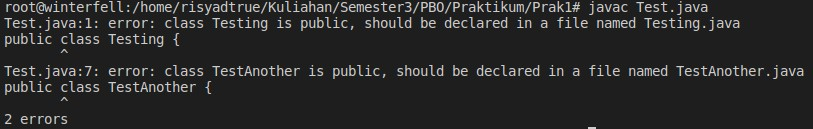
**Latihan 1 : Menganalisa dan membenahi kesalahan pada program**

****

Analisa :

Kesalahannya adalah, nama class berbeda dengan nama file .javanya. Sementara pada konsep Java nama file dan classnya harus sama.

**Latihan 2 : Menganalisa dan membenahi kesalahan pada program**



Analisa :

Kesalahnya adalah, suatu file.java hanya bisa ada satu class di dalamnya, sedangkan pada program ini ada dua class di dalan satu file.

**Latihan 3 : Menganalisa dan membenahi kesalahan pada program**



Analisa :

Kesalahanya adalah, pada pernyataan main method yang seharusnya setelah kata *String* ada tanda “ [] “ maka harus diperbaiki.

**Latihan 4 : Menganalisa dan membenahi kesalahan pada program**

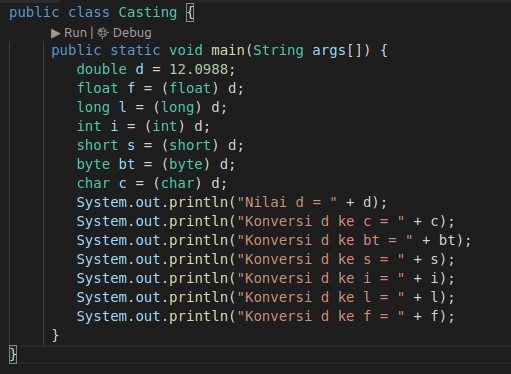
****

Analisa :

Kesalahanya adalah, pada pernyataan main method yang seharusnya setelah kata *String* ada tanda “ [] “ maka harus diperbaiki.

**Latihan 5 : Membuat formulasi proses casting dari tipe-tipe primitif**

Code :



Output :

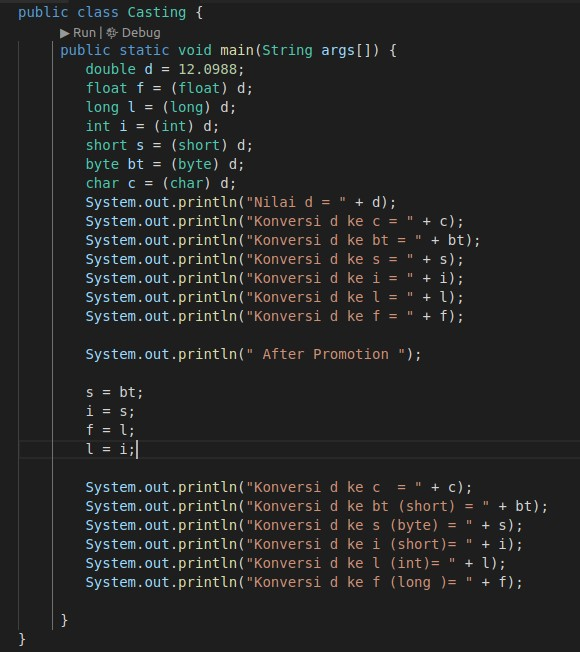


Analisa :

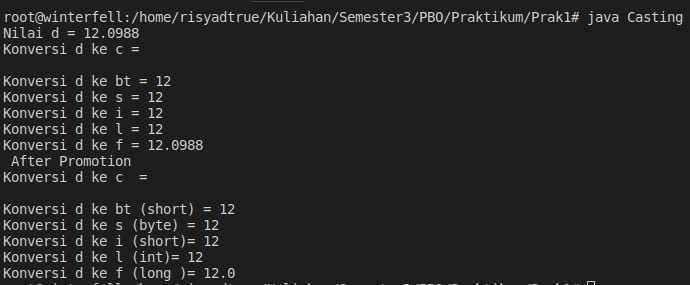
Program tersebut menghasilkan nilai konversi dari nilai double d = 12.0988 ke bentuk tipe tipe primitif.

**Latihan 6 : Membuat formulasi proses promotion dari tipe-tipe primitif**

code :

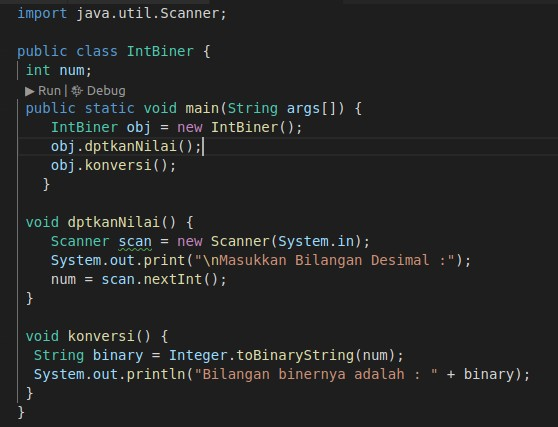
****

Output :



**Latihan 7 : Menampilkan representasi biner dari bilangan desimal bertipe int**

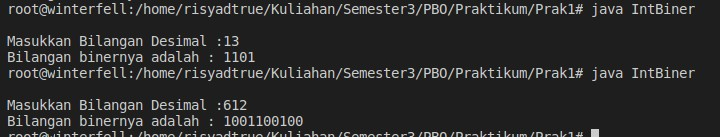
Code :



Output :

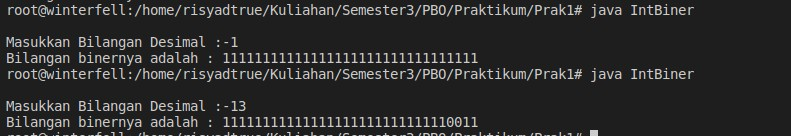
angka : 13

angka : 612



angka : -1

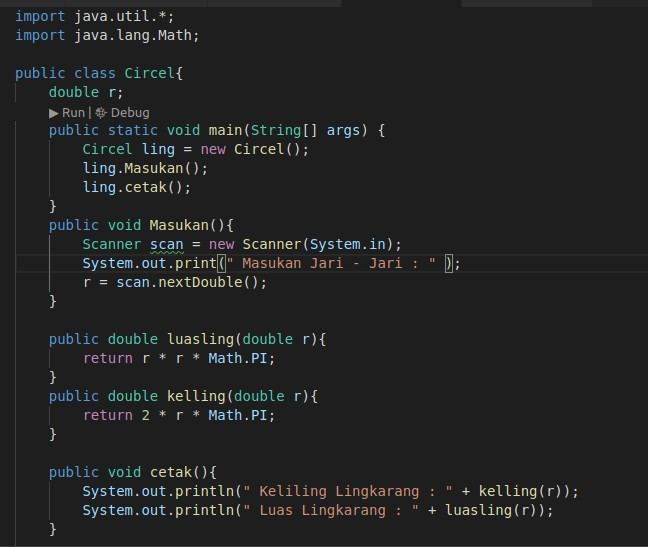
angka : -13



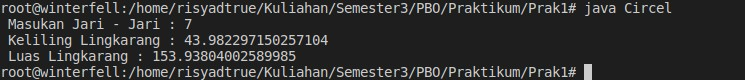
E. Tugas

Tugas 1 :

Code :

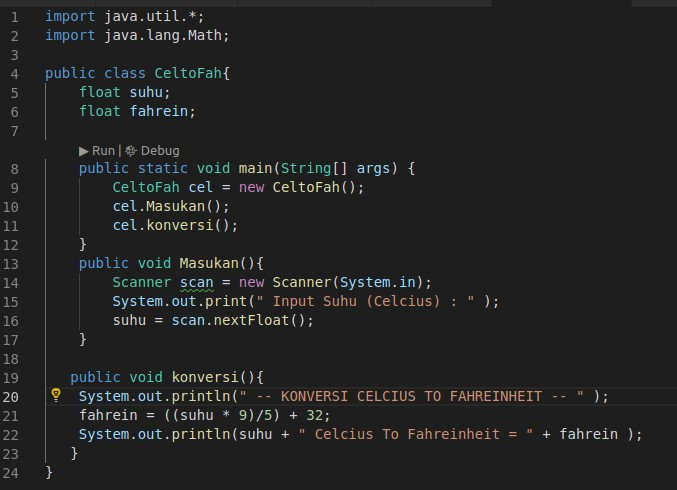


Output :

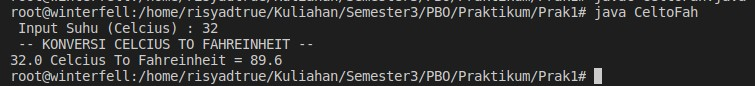


Tugas 2 :

Code :



Output :



Tugas 3 :

Kesalahan dalam penulisan tipe data, di akhir nilai harus di tambah “ L “ unutk mengindikasi bahwa nilainya “ Long “

Tugas 4 :

|  |
| --- |
| import java.util.\*;  public class KonversiWaktu{  int jam\_awal, menit\_awal, jam\_akhir, menit\_akhir, durasi\_menit, menit\_terakhir,jam\_terakhir, durasi\_jam, total;  public static void main(String[] args) {  KonversiWaktu wkt = new KonversiWaktu();  wkt.inputAwal();  wkt.hitung();  wkt.cetak();  }  void inputAwal(){  Scanner scan = new Scanner(System.in);  System.out.print("Masukan Jam awal : ");  jam\_awal = scan.nextInt();  System.out.print("Masukan menit awal : ");  menit\_awal = scan.nextInt();  System.out.print("Masukan Jam akhir : ");  jam\_akhir = scan.nextInt();  System.out.print("Masukan menit akhir : ");  menit\_akhir = scan.nextInt();  }  void hitung(){  if((menit\_akhir-menit\_awal)<0){  menit\_terakhir=60+menit\_akhir;  durasi\_menit=menit\_terakhir-menit\_awal;  jam\_terakhir=jam\_akhir-1;  durasi\_jam=jam\_terakhir-jam\_awal;  total=(durasi\_jam\*60)+durasi\_menit;  }  else if ((menit\_awal-menit\_akhir)<0){  menit\_terakhir = 60+menit\_akhir;  durasi\_menit = menit\_terakhir-menit\_awal;  jam\_terakhir = jam\_akhir-1;  durasi\_jam = jam\_terakhir-jam\_awal;  total=(durasi\_jam\*60)+durasi\_menit;  }  }  void cetak(){  System.out.println(" Selish Waktu anda : " + total + " Menit " );  }  } |

Output :

