

**PEMBUATAN PROGRAM PERHITUNGAN  
LUAS BANGUN DATAR**

**LAPORAN**



Disusun Oleh:

**FERDI RAHMAT  
NIM: 202013005**

**DOSEN PENGAMPU:  
SLAMET TRIYANTO,S.ST**

**PROGRAM STUDI TEKNIK INFORMATIKA  
POLITEKNIK KAMPAR**

**2020/2021**

## **KATA PENGANTAR**

Dengan memanjatkan puji syukur kehadiran Allah SWT. Atas limpahan rahmad dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan tugas ujian akhir semester yang berjudul “Pembuatan Program Perhitungan Luas Bangun Datar”

Didalam tulisan ini disajikan pokok-pokok bahasan yang meliputi Pembuatan program perhitungan luas bangun datar. Penulis mengharapkan saran yang membangun agar tulisan ini bermanfaat bagi perkembangan ilmu pengetahuan kedepan.

Bangkinang, 13 Maret 2021

penulis

## DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI.....	3
BAB I.....	5
PENDAHULUAN.....	5
A. Latar Belakang.....	5
B. Tujuan Laporan .....	5
BAB II .....	5
PEMBAHASAN .....	5
A. Pengertian Bahasa Java.....	5
B. Sejarah Java .....	6
C. Variabel Java.....	7
1. Jenis-jenis variable java .....	7
D. Tipe Data.....	8
1. Macam-Macam Tipe Data pada Java.....	8
E. Operator Java.....	12
1. Operator Aritmatika .....	12
2. Operator Pembandingan.....	12
3. Operator Logika .....	13
F. Percabangan .....	13
1. STATEMENT IF .....	13
2. STATEMENT IF ELSE.....	14
3. STATEMENT ELSE IF .....	15
4. STATEMENT SWITCH-CASE .....	16
G. Perulangan.....	17
1. Perulangan For Pada Java .....	17
2. Perulangan While Pada Java .....	19
3. Perulangan Do While Pada Java.....	20
H. Larik/Array .....	21
BAB III.....	22

<b>PROGRAM PERHITUNGAN LUAS BANGUN DATAR .....</b>	<b>22</b>
<b>A. Source code .....</b>	<b>22</b>
1. Persiapan .....	22
2. Menampilkan Daftar .....	23
3. Pemilihan Rumus .....	23
4. Akhiran.....	24
<b>B. Tampilan Program.....</b>	<b>25</b>
1. Tampilan Menu Awal .....	25
2. Tampilan Hasil Menu .....	26
<b>BAB IV .....</b>	<b>27</b>
<b>PENUTUP.....</b>	<b>27</b>
<b>A. Kesimpulan .....</b>	<b>27</b>
<b>B. Saran .....</b>	<b>27</b>
<b>DAFTAR PUSTAKA .....</b>	<b>27</b>

## BAB I

### PENDAHULUAN

#### A. Latar Belakang

Saat ini, dengan perkembangan teknologi yang semakin berkembang dengan lingkungan yang semakin luas dan banyak diminati, dan mempunyai nilai ekonomi uang sangat tinggi yaitu dunia pemrograman komputer. Seperti halnya dunia pemrograman komputer, seperti C,C++,Pascal Basic,Java, dan lain-lain. Di antara bahasa pemrograman , java adalah yang paling diminati, karena perannya yang sudah tidak di ragukan lagi dan ke eksisannyadalam perkembangan TI.

#### B. Tujuan Laporan

Tujuan laporan ini adalah untuk menyelesaikan tugas akhir ujian semester I mata kuliah algoritma dan pemrograman TA.2020/2021.

## BAB II

### PEMBAHASAN

#### A. Pengertian Bahasa Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk **telepon genggam**. Bahasa ini awalnya dibuat oleh **James Gosling** saat masih bergabung di **Sun Microsystems**, yang saat ini merupakan bagian dari **Oracle** dan dirilis tahun **1995**. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada **C** dan **C++** namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam **p-code** (*bytecode*) dan dapat dijalankan pada berbagai **Mesin Virtual Java (JVM)**. Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform **sistem operasi** yang berbeda, java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Saat ini java merupakan bahasa pemrograman yang paling

populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

## B. Sejarah Java

Bahasa pemrograman Java terlahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, dan James Gosling, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot *Duke* yang dibuat oleh Joe Palrang. Pertemuan proyek berlangsung di sebuah gedung perkantoran *Sand Hill Road* di Menlo Park. Sekitar musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program *Java Oak* pertama, yang ditujukan sebagai pengendali sebuah peralatan dengan teknologi layar sentuh (*touch screen*), seperti pada PDA sekarang ini. Teknologi baru ini dinamai "*\*7*" (*Star Seven*). Setelah era *Star Seven* selesai, sebuah anak perusahaan TV kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto. Perusahaan baru ini bertambah maju: jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer. Mereka

menjadikan peramban (*browser*) Mosaic sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner, terinspirasi dari film 1980-an, *Blade Runner*. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java. Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0a2 dibuka. Kesuksesan mereka diikuti dengan pemberitaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995. Sayangnya terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel Sheraton Palace. Tiga dari pimpinan utama proyek, Eric Schmidt dan George Paolini dari Sun Microsystems bersama Marc Andreessen, membentuk Netscape. Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling. Konon kopi ini berasal dari Pulau Jawa. Jadi nama bahasa

pemrograman Java tidak lain berasal dari kata Jawa (bahasa Inggris untuk Jawa adalah Java).

### C. Variabel Java

Variabel adalah lokasi atau tempat di memori yang digunakan untuk menyimpan informasi atau data dari sebuah *program* untuk selanjutnya dapat diproses. Ketika kita membuat sebuah variabel, berarti kita memerintahkan sebuah memori untuk menyediakan alokasi *space* atau ruang bagi variabel. Ruang atau *space* yang dialokasikan tergantung dari tipe data dari variabel tersebut.

#### 1. Jenis-jenis variable java

##### a. Local Variable

Variabel yang dideklarasikan dalam sebuah lingkup kecil (lokal) seperti *method/constructor/blocks*. Berarti, setiap variabel yang dideklarasikan dalam *method* disebut lokal variabel. Variabel lokal mempunyai ruang lingkup yang kecil juga. Karena dideklarasikan di sebuah *method*, maka variabel tersebut juga hanya bisa diakses dalam sebuah *method* juga.

##### b. instance variable

Variabel yang dideklarasikan dalam sebuah *class* namun diluar *method*. Java adalah bahasa pemrograman berbasis *object*, setiap kita menulis kode program java, secara tidak langsung kita sudah membuat *class*. Pendeklarasikan variabel langsung di dalam *class* tanpa dibungkus *method* sering disebut dengan instance variabel. Instance variable juga dapat kita beri *access modifier*-nya masing-masing.

##### c. Static Variable

Variabel yang direklarasikan dengan *keyword static*, biasanya digunakan untuk mendeklarasikan “sesuatu” yang bersifat *live*, yang artinya variabel ini dapat diakses langsung tanpa membuat *object* dari *class* terlebih dahulu. Juga, *static variable* akan

membuat sebuah “dunia” sendiri dan akan dipakai bersama-sama dengan *object* lain yang terbentuk. Penjelasan lebih lengkap akan dijelaskan ketika kita sudah memasuki materi *class* dan *object*.

## D. Tipe Data

### 1. Macam-Macam Tipe Data pada Java

Java mempunyai 11 macam tipe data yang terdiri atas tipe data sederhana dan referensi / komposit. Tipe sederhana meliputi byte, short, int, long, char, float, double dan boolean yang terbagi menjadi 3 tipe. Sedangkan tipe data referensi meliputi class, array dan interface

#### a. Integer(Bilangan Bulat)

Tipe data yang masuk menjadi bagian ini adalah byte, short, int dan long. Semua tipe data ini bersifat *Signed*, yaitu bisa mempresentasikan nilai positif dan negatif. Tidak seperti tipe data lainnya, Java tidak mendukung tipe data *unsigned* yang hanya bisa mempresentasikan nilai positif. Untuk jelasnya akan dijelaskan oleh tabel dan penjelasan di bawah ini :

Tipe Data	Ukuran (bit)	Range
<i>Byte</i>	8	-128 s.d. 127
<i>Short</i>	16	-32768 s.d. 32767
<i>Int</i>	32	-2147483648 s.d. 2147483647
<i>Long</i>	64	-9223372036854775808 s.d. 9223372036854775807



## **Byte**

Type *byte* umumnya digunakan pada saat kita bekerja dengan sebuah data *stream* dari suatu file maupun jaringan, yaitu untuk keperluan proses membaca/menulis. Selain itu, tipe ini juga digunakan saat bekerja dengan data biner yang tidak kompatibel dengan tipe-tipe lain yang didefinisikan di dalam Java.

## **Short**

Pada umumnya diaplikasikan pada komputer-komputer 16-bit, yang saat ini semakin jarang keberadaanya.

## **Int**

Tipe ini merupakan tipe yang paling banyak dipakai dalam merepresentasikan angka dalam Java, dikarenakan dianggap paling efisien dibandingkan dengan tipe-tipe integer lainnya. Tipe *Int* banyak digunakan untuk indeks dalam struktur pengulangan maupun dalam konstruksi sebuah *array*. Selain itu, secara teori setiap ekspresi yang melibatkan tipe integer (*byte*, *short*, *int*, *long*) semuanya akan dipromosikan ke *int* terlebih dahulu sebelum dilakukan proses perhitungan.

## **Long**

Tipe ini digunakan untuk kasus-kasus tertentu yang nilainya berada di luar rentang tipe *int*, karna tipe ini punya range paling tinggi dibanding *Integer* lainnya. Dengan kata lain, tipe *long* terpaksa digunakan jika data memiliki range diluar range *int*.

### **b. Floating-Point (Bilangan Pecahan)**

Tipe *floating-point* digunakan untuk merepresentasikan nilai-nilai yang mengandung pecahan atau angka decimal di belakang koma, seperti 3.1416, 5.25, dan sebagainya. Bilangan semacam ini disebut sebagai bilangan riil. Dalam Java tipe ini dibedakan menjadi dua jenis, yaitu *float*, dan *double*. Untuk jelasnya akan dijelaskan oleh tabel dan penjelasan di bawah ini :

Tipe	Ukuran		Range	Presisi (jumlah digit)
	bytes	bit		
float	4	32	+/- 3.4 x 10 <sup>38</sup>	6-7
double	8	64	+/- 1.8 x 10 <sup>308</sup>	15

### Float

Tipe ini digunakan untuk menandakan nilai-nilai yang mengandung presisi atau ketelitian tunggal (single-precision) yang menggunakan ruang penyimpanan 32-bit. Presisi tunggal biasanya lebih cepat untuk processor-processor tertentu dan memakan ruang penyimpanan setengah kali lebih sedikit dibandingkan presisi ganda (double precision). Permasalahan yang timbul dari pemakaian tipe float untuk nilai-nilai yang terlalu kecil atau justru terlalu besar, karena nilai yang dihasilkan akan menjadi tidak akurat.

### Double

Tipe ini mengandung tingkat ketelitian ganda atau presisi ganda (double precision) dan menggunakan ruang penyimpanan 64-bit untuk menyimpan nilai. Tipe double tentu lebih cepat untuk melakukan perhitungan-perhitungan matematis daripada tipe float. Untuk perhitungan yang bersifat bilangan riil dan menghasilkan hasil yang lebih akurat, maka lebih baik menggunakan tipe double.

### c. Char

Tipe data *char* merupakan tipe untuk menyatakan sebuah karakter. Java menggunakan karakter *Unicode* untuk merepresentasikan semua karakter yang ada. *Unicode* ialah sekumpulan karakter yang terdapat pada semua bahasa, seperti bahasa Latin, Arab, Yunani dan lain-lainnya. Karena bahasa Java dirancang untuk dapat diterapkan di berbagai macam *platform*, maka Java menggunakan karakter *Unicode* yang membutuhkan ukuran 16-bit. Untuk karakter-karakter yang tidak dapat diketikkan secara langsung melalui keyboard, java menyediakan beberapa *escape sequence* (pasangan karakter yang dianggap sebagai karakter tunggal). *Escape sequence* tidak dianggap sebagai *String*, melainkan tetap sebagai tipe karakter khusus. Di bawah ini akan dijelaskan beberapa contoh tentang *escape sequence*.

<i>Escape Sequence</i>	Keterangan
<code>\ddd</code>	Karakter octal (ddd)
<code>\uxxxx</code>	Karakter Unicode heksadecimal (xxxx)
<code>\'</code>	Petik tunggal
<code>\''</code>	Petik ganda
<code>\\</code>	<i>Backslash</i>
<code>\r</code>	<i>Carriage return</i>
<code>\n</code>	Baris baru ( <i>line feed</i> )
<code>\f</code>	<i>Form feed</i>
<code>\t</code>	<i>Tab</i>
<code>\b</code>	<i>Backspace</i>

### ean

Tipe *boolean* adalah tipe data yang digunakan untuk menampung nilai logika, yaitu nilai yang hanya memiliki dua buah kemungkinan (benar atau salah). Tipe ini ditandai dengan kata kunci *Boolean*. Dalam bahasa Java, nilai benar dipresentasikan dengan kata kunci *true* dan nilai salah dengan kata kunci *false*.

## E. Operator Java

Sebelum masuk ke jenis-jenis operator di dalam bahasa Java, terdapat istilah *operand* dan *operator*. Operand adalah nilai asal yang dipakai dalam sebuah proses operasi. Sedangkan Operator adalah instruksi yang diberikan untuk mendapatkan hasil dari proses tersebut.

Biasanya operator berbentuk karakter matematis atau perintah singkat sederhana. Sebagai contoh, pada operasi:  $10 + 2$ . Angka 10 dan 2 disebut sebagai operand, sedangkan tanda tambah (karakter  $+$ ) adalah operator.

Berikut adalah beberapa kategori operator aritmatika dan operator logika yang sering digunakan di dalam bahasa pemrograman dan algoritma:

### 1. Operator Aritmatika

- a. (jumlah) adalah operator untuk menjumlahkan dua buah angka atau lebih
- b. -(kurang) adalah jenis operator untuk memproses operasi pengurangan antara dua angka atau lebih
  - (kali) adalah operator yang digunakan untuk proses perkalian
- c. / (bagi) adalah operator yang digunakan untuk mendapatkan hasil bagi dari 2 buah angka

MOD atau % (modular) adalah operator yang sering digunakan di dunia algoritma yang digunakan untuk mendapatkan sisa hasil bagi.

### 2. Operator Pembandingan

Operator pembandingan adalah operator yang merupakan bagian dari operator logika yang digunakan untuk membandingkan 2 buah nilai atau operan, sehingga dapat menghasilkan nilai true dan false, beberapa operator pembandingan yang sering digunakan adalah:

- a.  $>$  lebih besar
- b.  $<$  lebih kecil
- c.  $>=$  lebih besar sama dengan
- d.  $<=$  lebih kecil sama dengan
- e.  $<>$  tidak sama dengan
- f.  $!=$  tidak sama dengan

g. == apakah sama dengan

h. := sama dengan

Di dalam bahasa pemrograman Operator pembanding biasanya digunakan dalam flow control IF then ELSE untuk mendapatkan hasil sesuai dengan kondisi yang diinginkan.

### 3. Operator Logika

Operator Logika adalah operator yang digunakan untuk melakukan operasi logika yaitu operator yang menghasilkan nilai TRUE (benar) atau FALSE (salah). Beberapa macam operator logika antara lain:

- a. and : menghasilkan nilai TRUE jika kedua operand bernilai TRUE
- b. or : menghasilkan nilai TRUE jika salah satu operand bernilai TRUE
- c. xor : menghasilkan nilai TRUE jika salah satu operand bernilai TRUE tetapi bukan keduanya bernilai TRUE
- d. ! (simbol tidak): menghasilkan nilai tidak TRUE
- e. && (simbol AND): menghasilkan nilai TRUE jika kedua operand bernilai TRUE
- f. || (simbol OR): menghasilkan nilai TRUE jika salah satu operand bernilai TRUE

Operator logika sering digunakan di dalam flow control berkolaborasi dengan operator pembanding untuk mendapatkan hasil yang paling sesuai dengan kondisi tertentu.

## F. Percabangan

Struktur kontrol pemilihan atau percabangan adalah pernyataan dari Java yang mengijinkan user untuk memilih dan mengeksekusi blok kode spesifik dan mengabaikan blok kode yang lain.

Dalam java terdapat 4 macam jenis percabangan, if, if-else, else-if, dan juga switch. Keempat jenis ini memiliki penggunaan masing-masing.

Berikut penjelasan mengenai penggunaan tiap-tiap percabangan :

### 1. STATEMENT IF

Percabangan if ini digunakan jika kita hanya memiliki satu pernyataan yang akan dijalankan dengan syarat tertentu. Sintaks if seperti berikut :

```
if(kondisi){  
    pernyataan  
}
```

Jika kondisi benar, maka pernyataan akan dijalankan.

```
1public class coba{  
2public static void main (String [] args){  
3int a=0;  
4if (a==0)  
5System.out.println("Nilai a = 0");  
6if (a==1)  
7System.out.println("Nilai a = 1");  
8}  
9}
```

Output : Program akan menampilkan nilai a = 0 saja, karena pada if yang kedua, kondisi tidak memenuhi atau salah.

## 2. STATEMENT IF ELSE

Percabangan if else digunakan saat kita memiliki dua pernyataan dengan syarat tertentu. Sintaks if-else seperti berikut :

```
if(kondisi){  
    pernyataan1  
}else{  
    pernyataan2  
}
```

Jika hasil dari if benar, maka pernyataan1 yang dijalankan, sedangkan jika

```
1public class coba{  
2public static void main (String [] args){  
3int a=0;  
4if (a==0)  
5System.out.println("Nilai a = 0");  
6else  
7System.out.println("Nilai a = 1");  
8}  
9}
```

salah pernyataan dua yang akan dijalankan.

Output : program akan menampilkan nilai a = 0, karena kondisi if bernilai benar, jika pada inisialisasi nilai a tidak bernilai 0, maka program akan menampilkan nilai a = 1.

### 3. STATEMENT ELSE IF

percabangan yang digunakan saat kita memiliki banyak kondisi (lebih dari 2) dan banyak pernyataan (lebih dari 2). Sintaks dari else-if seperti berikut :

```
if(kondisi){
    pernyataan1
}elseif(kondisi2){
    pernyataan2
}else(kondisi3){
    pernyataan3
}
else {
    pernyataan4
}
```

jika kondisi1 benar, maka pernyataan1 akan dijalankan, jika kondisi2 benar, maka pernyataan2 akan dijalankan, jika semua kondisi salah, maka pernyataan4 yang akan dijalankan saja.

```
1    public class coba{
2
3        public static void main (String [] args){
4
5            int a=2;
6
7            if (a==0)
8
9                System.out.println("Nilai a = 0");
10
11            else if(a==1)
12
13                System.out.println("Nilai a = 1");
14
15            else if(a==2)
16
17                System.out.println("Nilai a = 2");
18
19            }
20        }
```

```
11    }
```

Output : Program akan menampilkan nilai a = 2 saja, karena pada else-if yang ketiga atau dengan pernyataan a==2 bernilai benar, sedangkan pernyataan yang lain tidak dijalankan karena kondisi tidak memenuhi.

#### 4. STATEMENT SWITCH-CASE

percabangan yang digunakan saat kita memiliki banyak kondisi (lebih dari 2) dan banyak pernyataan (lebih dari 2). Sebenarnya switch-case ini hampir sama dengan else if, hanya sintaksnya yang berbeda. Sintaks dari switch-case seperti berikut :

```
switch (variabel) {  
    case nilai1: pernyataan1;  
    break;  
    case nilai2: pernyataan2;  
    break;  
    default: pernyataan3;  
}
```

Jika nilai variabel yang ditunjuk bernilai sesuai nilai1, maka pernyataan1 akan dijalankan, jika nilai variabel yang ditunjuk bernilai sesuai nilai2, pernyataan2 dijalankan, jika tidak, maka pernyataan3 yang akan dijalankan.

```
1 public class coba{  
2     public static void main (String [] args){  
3         int a=3;  
4         switch (a) {  
5             case 1:  
6                 System.out.println("Nilai a=1");break;  
7             case 2:  
8                 System.out.println("Nilai a=2"); break;
```



```
9 case 3:
10 System.out.println("Nilai a=3");break;
11 default:
12 System.out.println("Nilai a=4"); break;
13 }
14 }
15 }
```

Output : Program akan menampilkan nilai a = 2 saja, karena kondisi bernilai benar, perhatikan break dibelakang pernyataan, jika break ini dihapus, maka semua pernyataan akan dijalankan. Break digunakan untuk keluar dari switch-case saat 1 pernyataan sudah dijalankan.

## G. Perulangan

### 1. Perulangan For Pada Java

Perulangan for atau looping for termasuk dalam jenis perulangan yang cukup banyak digunakan bukan hanya di java namun di bahasa pemrograman lain seperti PHP, C++ dan Python. Perulangan ini digunakan ketika ingin mengeksekusi perintah program yang sama dengan jumlah proses perulangan yang sudah diketahui dengan mengacu pada kondisi yang ditetapkan. Bentuk umum perulangan for adalah seperti berikut:

```
for (nilai_awal;kondisi;modifier){
    pernyataan_yang_diulang;
}
```

#### Keterangan:

- Nilai awal** : Bagian dimana kita menginisialisasi nilai awal dimana menjadi titik awal perulangan dimulai.

- b. **Kondisi** : Bagian untuk menetapkan suatu kondisi tertentu, ketika kondisi bernilai benar perulangan akan terus berjalan.
- c. **Modifier** : Bagian dimana kita dapat melakukan penambahan nilai (*increment*) atau bisa juga penurunan nilai (*decrement*) pada nilai awal yang telah di inisialisasi.

Setiap instruksi yang terdapat didalam blok perulangan For akan diulang secara terus-menerus hingga kondisi bernilai salah (false).

Contoh:

```
package perulangan;

public class perulangan_for {
    public static void main(String[] args) {
        for (int i=1;i<=10;i++){
            System.out.print(i+", ");
        }
    }
}
```

Pada program diatas pada perulangan for untuk parameter pertama kita lakukan inisialisasi nilai awal variabel i=1. kemudian kita membuat kondisi i<=10. pada bagian modifier kita melakukan penambahan (*increment*) nilai i++. Artinya bahwa ketika kondisi bernilai benar nilai i akan ditambahkan 1 begitu seterusnya hingga kondisi bernilai false.

Hasil Output:

```
1, 2, 3, 4, 5, 6, 7, 8, 9, 10
```

Coba perhatikan dari hasil *output* diatas instruksi didalam blok for **System.out.print(i+", ");** untuk mencetak nilai **i** terus dilakukan hingga nilai **i** memenuhi kondisi **i<=10**.

## 2. Perulangan While Pada Java

Perulangan `while` adalah jenis perulangan yang digunakan ketika kita belum mengetahui jumlah proses perulangan, Banyaknya perulangan ditentukan dari proses inialisasi dan kondisi yang didefinisikan. Berbeda dengan perulangan `for` yang sudah dapat kita ketahui jumlah proses perulangan dari kondisi yang ditetapkan.

```
while (kondisi){  
    pernyataan;  
}
```

Keterangan:

- a. Kondisi – Sebagai sebuah syarat untuk menghentikan proses perulangan
- b. Pernyataan – Bagian instruksi-instruksi program yang akan diulang saat kondisi bernilai benar (*true*)

Untuk menghentikan perulangan harus ada suatu perintah yang digunakan untuk mengubah nilai kondisi menjadi *false* agar proses perulangan dapat berhenti.

Contoh:

```
package perulangan;  
import java.util.Scanner;  
public class perulangan_while {  
    public static void main(String[] args) {  
  
        Scanner input=new Scanner(System.in);  
        int bilangan=1;  
  
        while (bilangan!=0){  
            System.out.print("Masukkan Bilangan 0 : ");  
            bilangan=input.nextInt();  
        }  
    }  
}
```

```
    }  
  }  
}
```

Pada contoh program diatas kita menggunakan fungsi scanner untuk pengguna dapat menginput nilai saat program sedang berjalan.

### 3. Perulangan Do While Pada Java

Perulangan Do While merupakan masih saudara dengan perulangan while perbedaan dari keduanya adalah jika pada perulangan while kondisi sebagai syarat perulangan yang akan dilakukan di cek pada awal sebelum perulangan dilakukan, jika bernilai benar (*true*) baru kemudian perulangan dijalankan. Sedangkan pada perulangan Do While perulangan dilakukan terlebih dahulu minimal sekali baru kemudian setelah itu dicek kondisinya, jadi dengan kata lain do While pengecekan kondisinya setelah perulangan dilakukan sementara while sebelum dilakukan.

Bentuk Umum:

```
do {  
    pernyataan;  
} while (kondisi);
```

Pernyataan di dalam blok do { } akan dieksekusi (minimal 1 kali) lalu kemudian akan di cek kondisi-nya. Coba perhatikan contoh berikut:

```
package perulangan;  
  
public class perulangan_do_while {  
    public static void main(String[] args) {
```

```
int bilangan=1;
do{
    System.out.println(bilangan);
}while (bilangan<=5);
}
}
```

Hasil Output:

1

## H. Larik/Array

Dalam bahasa Java tipe data larik direpresentasikan sebagai sebuah [objek](#) khusus. Karena itu pada bahasa Java larik yang dibuat selalu bersifat dinamik. Namun walaupun bersifat dinamik, larik pada bahasa Java tidak perlu dihancurkan karena proses penghancuran dilakukan secara [otomatis](#) melalui suatu prosedur yang disebut dengan [Pengumpulan sampah](#) ([Inggris](#): Garbage Collecting).

Sama seperti bahasa C, indeks larik selalu dimulai dari 0.

Contoh:

```
public class larik {
    public static void main(String args[]) {
        int[] arr = new int[10];
        arr[0] = 5;
        System.out.println(arr[0]);
    }
}
```

## **BAB III**

### **PROGRAM PERHITUNGAN LUAS BANGUN DATAR**

#### **A. Source code**

Berikut bagian-bagian sourcecodenya:

##### **1. Persiapan**

Pada bagian ini berisi penyiapan variable-variabel yang di butuhkan dalam program.

```

1  import java.util.Scanner;
2  public class project
3  {
4      public static void main(String[] args) throws Exception{
5          int jumlah,i,j,swap,pilih,lagi,alas,tinggi,luas,lebar,panjang,diagonal1,diagonal2,sisi,sisi_atas,sisi_bawah;
6          double pi=3.14,r;
7          Scanner scan=new Scanner(System.in);
8          String text[]={"P","R","O","G","R","A","M"," " ,"P","E","R","H","I","T","U","N","G","A","N"," " ,"L","U","A","S"," " ,"B","A","
9          "N","G","U","N"," " ,"D","A","T","A","R"};
10         int a=0;
11         System.out.println("\t=====*****");
12         System.out.print("\t\t");
13         while(a<text.length){
14             Thread.sleep(60);
15             System.out.print(text[a]);
16             a++;
17         }
18         System.out.println();
19         System.out.println("\t=====*****");
20         System.out.println();
21
22         int array[]={3,5,1,4,6,8,2,7};
23         String list[]={"luas segitiga","luas persegi","luas persegi panjang","luas lingkaran","luas trapesium","luas jajar genjang",
24         "luas layanglayang","luas Belah Ketupat"};
25
26         for(i=0; i<(array.length-1);i++)
27         {
28             for(j=0;j<array.length-i-1;j++)
29             {
30                 if (array[j]>array[j+1])
31                 {
32                     swap = array[j];
33                     array[j] = array[j+1];
34                     array[j+1]=swap;{}
35                 }
36             }
37         }

```

## 2. Menampilkan Daftar

```

38         System.out.println("=====");
39         System.out.println("DAFTAR RUMUS :");
40         for(i=0;i<array.length;i++){
41             System.out.println(array[i]+" . "+list[i]);
42         }
43         System.out.println("=====");

```

## 3. Pemilihan Rumus

Bagian ini berisi percabangan switch case untuk pemilihan rumus.

```

44 pil:
45 for(;;){
46     System.out.print("\nPilih Rumus yang Anda Butuhkan : ");
47     pilih=scan.nextInt();
48     System.out.print("Pilihan Rumus anda No : "+pilih);
49     System.out.println();
50     switch (pilih){
51         case 1:
52             double luas_segitiga;
53             System.out.println("Perhitungan Luas Segitiga");
54             System.out.print("Masukan Alas = ");
55             alas=scan.nextInt();
56             System.out.print("Masukan Tinggi = ");
57             tinggi=scan.nextInt();
58             luas_segitiga=alas*tinggi/2;
59             System.out.println("luas segitiga adalah = "+luas_segitiga);
60         break;
61         case 2:
62             int luas_persegi;
63             System.out.println("Perhitungan Luas Persegi");
64             System.out.print("Masukan Sisi = ");
65             sisi=scan.nextInt();
66             luas_persegi=sisi*sisi;
67             System.out.println("luas persegi adalah = "+luas_persegi);
68         break;
69         case 3:
70             int luas_persegi_panjang;
71             System.out.println(" Perhitungan Luas Persegi Panjang");
72             System.out.print("Masukan Panjang = ");
73             panjang=scan.nextInt();
74             System.out.print("Masukan Lebar = ");
75             lebar=scan.nextInt();
76             luas_persegi_panjang=panjang*lebar;
77             System.out.println("luas luas persegi panjang adalah = "+luas_persegi_panjang);
78         break;
79         case 4:
80             double luas_lingkaran;
81             System.out.println("Perhitungan Luas Lingkaran");
82             System.out.print("Masukan Jari-Jari = ");
83             r=scan.nextInt();
84             luas_lingkaran=pi*r*r;
85             System.out.println("luas lingkaran adalah = "+luas_lingkaran);
86         break;
87         case 5:
88             int luas_trapesium;
89             System.out.println("Perhitungan Luas Trapesium");
90             System.out.print("Masukan Sisi Atas = ");
91             sisi_atas=scan.nextInt();
92             System.out.print("Masukan Sisi Bawah = ");
93             sisi_bawah=scan.nextInt();
94             System.out.print("Masukan Tinggi = ");
95             tinggi=scan.nextInt();
96             luas_trapesium=(sisi_atas+sisi_bawah)*tinggi/2;
97             System.out.println("luas lingkaran adalah = "+luas_trapesium);
98         break;
99         case 6:
100             double luas_jajar_genjang;
101             System.out.println("Perhitungan Luas Jajar Genjang");
102             System.out.print("Masukan Alas = ");
103             alas=scan.nextInt();
104             System.out.print("Masukan Tinggi = ");
105             tinggi=scan.nextInt();
106             luas_jajar_genjang=alas*tinggi;
107             System.out.println("luas Jajar Genjang adalah = "+luas_jajar_genjang);
108         break;
109         case 7:
110             double luas_layanglayang;
111             System.out.println("Perhitungan Luas Layang-Layang");
112             System.out.print("Masukan Diagonal 1 = ");
113             diagonal1=scan.nextInt();
114             System.out.print("Masukan Diagonal 2 = ");
115             diagonal2=scan.nextInt();
116             luas_layanglayang=diagonal1*diagonal2/2;
117             System.out.println("luas Luas Layang-Layang adalah = "+luas_layanglayang);
118         break;
119         case 8:
120             double luas_Belah_Ketupat;
121             System.out.println("Perhitungan Luas Belah Ketupat");
122             System.out.print("Masukan Diagonal 1 = ");
123             diagonal1=scan.nextInt();
124             System.out.print("Masukan Diagonal 2 = ");
125             diagonal2=scan.nextInt();
126             luas_Belah_Ketupat=diagonal1*diagonal2/2;
127             System.out.println("luas Belah Ketupat adalah = "+luas_Belah_Ketupat);
128         break;
129         break;
130     }
131 }
132

```

#### 4. Akhiran

Bagian ini berisi perulangan dan jika kondisi tidak terpenuhi system akan keluar.



```

131
132         break;
133     default:
134         System.out.println();
135         System.out.println("pilihan anda tidak tersedia");
136         continue pil;
137
138     }
139     System.out.print("Apakah Anda Ingin Coba lagi? (1:0) = ");
140     lagi=scan.nextInt();
141     System.out.println();
142     if(lagi==1){
143         continue pil;
144     }else{
145         System.out.println("\t=====*****=====");
146         System.out.println("\t\tTerima Kasih");
147         System.out.println("\t=====*****=====");
148         System.exit(0);
149     }
150     break;
151 }
152 }
153 }
154

```

Untuk penjelasan lebih jelas nya bisa dilihat dilink youtube berikut:

[\(325\) Tugas ALP - Penjelasan Project Ujian Akhir Semester - YouTube](#)

## B. Tampilan Program

### 1. Tampilan Menu Awal

```

=====*****=====
PROGRAM PERHITUNGAN LUAS BANGUN DATAR
=====*****=====

=====
DAFTAR RUMUS :
1. luas segitiga
2. luas persegi
3. luas persegi panjang
4. luas lingkaran
5. luas trapesium
6. luas jajar genjang
7. luas layanglayang
8. luas Belah Ketupat
=====

Pilih Rumus yang Anda Butuhkan :

```

## 2. Tampilan Hasil Menu

```
Pilih Rumus yang Anda Butuhkan : 3
Pilihan Rumus anda No : 3
  Perhitungan Luas Persegi Panjang
Masukan Panjang = 5
Masukan Lebar = 6
luas luas persegi panjang adalah = 30
Apakah Anda Ingin Coba lagi? (1:0) = 1
```

```
Pilih Rumus yang Anda Butuhkan : 8
Pilihan Rumus anda No : 8
Perhitungan Luas Belah Ketupat
Masukan Diagonal 1 = 4
Masukan Diagonal 2 = 5
luas Belah Ketupat adalah = 10.0
Apakah Anda Ingin Coba lagi? (1:0) = 1
```

```
Pilih Rumus yang Anda Butuhkan : 9
Pilihan Rumus anda No : 9
```

```
pilihan anda tidak tersedia
```

```
Pilih Rumus yang Anda Butuhkan : 2
Pilihan Rumus anda No : 2
Perhitungan Luas Persegi
Masukan Sisi = 5
luas persegi adalah = 25
Apakah Anda Ingin Coba lagi? (1:0) = 0
```

```
=====*****=====
                        Terima Kasih
=====*****=====
```

## **BAB IV**

### **PENUTUP**

#### **A. Kesimpulan**

Dari program yang saya buat diatas saya menggunakan type data integer, double, string, dan juga larik/array. Dan menggunakan perulangan for dan percabangan switch case serta menggunakan sorting untuk mengurutkan daftar rumus.

#### **B. Saran**

Meskipun penulis menginginkan kesempurnaan dalam penyusunan makalah ini tetapi kenyataannya masih banyak kekurangan yang perlu penulis perbaiki. Hal ini dikarenakan masih minimnya pengetahuan yang penulis miliki. Oleh karena itu kritik dan saran yang membangun dari para pembaca sangat penulis harapkan untuk perbaikan ke depannya.

### **DAFTAR PUSTAKA**

[\(PDF\) MAKALAH PENGENALAN JAVA SERTA NETBEANS SETTING PATH PADA JAVA | Rizki Fauzan - Academia.edu](#)

[Java - Wikipedia bahasa Indonesia, ensiklopedia bebas](#)

[Pengertian dan Jenis Variabel dalam Bahasa Pemrograman Java - Dunia IT](#)

[Macam-Macam Tipe Data pada Java \(dedykuncoro.com\)](#)

[Mengenal Pelbagai Jenis Operator Aritmatika dan Logika \(smktarunabangsa.sch.id\)](#)

[PERCABANGAN PADA JAVA – Codedevelop \(wordpress.com\)](#)

[Contoh Perulangan For, While, Do While pada Java | Kelas Programmer](#)

[Larik - Wikipedia bahasa Indonesia, ensiklopedia bebas](#)