

**LAPORAN**  
**PROGRAM PERHITUNGAN LUAS BANGUN DATAR**

**UJIAN AKHIR SEMESTER I**



**Disusun oleh :**

**MULYA SAFITRI**

**202013038**

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

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**POLITEKNIK KAMPAR**  
**2020-2021**

## **KATA PENGANTAR**

Puji dan syukur atas kehadiran Allah SWT, atas segala limpahan taufik serta hidayah-Nya yang telah memberi penulis kesempatan untuk menyelesaikan Laporan Ujian Akhir Semester, Tugas Akhir Semester yang berjudul Program Perhitungan Luas Bangun Datar ini. Adapun tujuan penulisan Laporan ini adalah untuk melengkapi Tugas dengan mata kuliah Algoritma Pemograman.

Dalam proses pembuatan Laporan ini, tentunya penulis mendapat bimbingan, arahan, koreksi dan saran. Untuk itu penulis mengucapkan terima kasih kepada Bapak Slamet Triyanto,S.ST., selaku dosen pengampu mata kuliah Algoritma Pemograman.

Penulis menyadari bahwa baik dari segi penulisan maupun isi, Laporan ini masih memiliki kekurangan, oleh karena itu penulis sangat mengharapkan kritik yang membangun dan saran dari pembaca agar terbentuknya kesempurnaan Laporan ini. Atas partisipasinya penulis mengucapkan terima kasih.

Bangkinang, 16 Maret 2021

( Penulis )

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR GAMBAR.....</b>	<b>iv</b>
<b>DAFTAR TABEL .....</b>	<b>vi</b>
<b>BAB I.....</b>	<b>1</b>
<b>TINJAUAN PUSTAKA.....</b>	<b>1</b>
<b>A. Pengertian.....</b>	<b>1</b>
1. Pengertian Bahasa Java.....	1
<b>B. Kelebihan dan Kekurangan Bahasa Java.....</b>	<b>2</b>
a. Kelebihan .....	2
b. Kekurangan .....	3
<b>C. Sejarah Bahasa Java.....</b>	<b>4</b>
<b>D. Tujuan Praktikum .....</b>	<b>6</b>
<b>E. Alat Dan Bahan .....</b>	<b>6</b>
<b>BAB II.....</b>	<b>7</b>
<b>PEMBAHASAN .....</b>	<b>7</b>
<b>A. DASAR-DASAR ALGORITMA PEMOGRAMAN.....</b>	<b>7</b>
1. Program Hello World .....	7
2. Tipe Data dan Variabel .....	10
3. Percabangan .....	16
4. Perulangan .....	21
5. Sorting (Pengurutan).....	26
<b>BAB III.....</b>	<b>28</b>
<b>LANGKAH KERJA MENJALANKAN PROGRAM.....</b>	<b>28</b>
<b>A. Instalasi Java Jdk.....</b>	<b>28</b>
1. Download java Jdk .....	28
2. Mengekstrak Java Jdk .....	28
3. Instal Java Jdk.....	28
<b>B. Program Perhitungan Luas Bangun Datar .....</b>	<b>31</b>

<b>BAB IV .....</b>	<b>46</b>
<b>PENUTUP.....</b>	<b>46</b>
<b>A. Kesimpulan.....</b>	<b>46</b>
<b>B. Saran .....</b>	<b>46</b>
<b>DAFTAR PUSTAKA.....</b>	<b>47</b>

## DAFTAR GAMBAR

Gambar 1.1 Logo Java .....	1
Gambar 2.1 program Hello World .....	8
Gambar 2.2 contoh program untuk materi konstruktor.....	10
Gambar 2.3 contoh program untuk tipe data.....	13
Gambar 2.4 hasil compile gambar program tipe data .....	14
Gambar 2.5 contoh program untuk percabangan IF.....	17
Gambar 2.6 hasil compile dari gambar percabangan IF .....	17
Gambar 2.7 contoh program untuk fungsi if else.....	18
Gambar 2.8 Hasil compile gambar contoh program untuk fungsi if else .....	18
Gambar 2.9 Gambar contoh program untuk fungsi Else If.....	19
Gambar 2.10 Hasil compile gambar contoh program untuk fungsi Else If .....	19
Gambar 2.11 Gambar contoh program Switch case.....	20
Gambar 2.12 Bentuk umum dari perulangan for .....	21
Gambar 2.13 Contoh program dari perulangan for.....	22
Gambar 2.14 Bentuk Umum dari perulangan While .....	23
Gambar 2.15 Contoh program dari perulangan While.....	23
Gambar 2.16 Hasil Compile program dari perulangan While .....	24
Gambar 2.17 Bentuk Umum dari perulangan Do While.....	25
Gambar 2.18 Contoh program dari perulangan Do While.....	25
Gambar 3.1 tampilan file jdk .....	28
Gambar 3.2 Tampilan Folder Bin .....	29
Gambar 3.3 Tampilan System.....	29
Gambar 3.4 Tampilan Environment Variabel.....	30
Gambar 3.5 Tampilan Versi Java.....	30
Gambar 3.6 Tampilan awal Program .....	31
Gambar 3.7 Hasil Compile Program berhasil login .....	31
Gambar 3.8 Hasil Compile Program tidak berhasil Login.....	32
Gambar 3.9 Tampilan Program untuk fungsi Menu1 .....	32
Gambar 3.10 Hasil Compile fungsi menu1 .....	33

Gambar 3.11 Hasil Compile fungsi menu1 pilihan 1 program .....	33
Gambar 3.12 Hasil Compile fungsi menu1 pilihan 2 keluar.....	33
Gambar 3.13 tampilan program untuk fungsi menu bangun datar.....	34
Gambar 3.14 tampilan program untuk fungsi menu bangun datar percabangan ..	34
Gambar 3.15 Hasil Compile fungsi menu pilihan 1 luas persegi.....	35
Gambar 3.16 Hasil Compile fungsi menu pilihan 1 keluar dari program.....	35
Gambar 3.17 Program untuk luas persegi .....	36
Gambar 3.18 Program untuk luas persegi Panjang .....	36
Gambar 3.19 Program untuk luas jajargejang.....	37
Gambar 3.20 Program untuk luas segitiga .....	37
Gambar 3.21 Program untuk luas belah ketupat .....	37
Gambar 3.22 Program untuk luas layang-layang.....	38
Gambar 3.23 Program untuk luas trapezium .....	38
Gambar 3.24 Program untuk luas lingkaran .....	39
Gambar 3.25 Program untuk luas semua bangun datar .....	39
Gambar 3.26 Hasil Compile Program untuk luas persegi.....	40
Gambar 3.27 Hasil Compile Program untuk luas persegi panjang .....	40
Gambar 3.28 Hasil Compile Program untuk luas jajargenjang.....	41
Gambar 3.29 Hasil Compile Program untuk luas segitiga.....	41
Gambar 3.30 Hasil Compile Program untuk luas belahketupat.....	42
Gambar 3.31 Hasil Compile Program untuk luas layang-layang.....	42
Gambar 3.32 Hasil Compile Program untuk luas trapesium .....	43
Gambar 3.33 Hasil Compile Program untuk luas Lingkaran.....	43
Gambar 3.34 Hasil Compile Program untuk luas semua bangun datar .....	44
Gambar 3.35 Tampilan Program untuk perulangan dan sorting .....	44
Gambar 3.36 Hasil Compile program fungsi perulangan dan sorting .....	45

## **DAFTAR TABEL**

Tabel 2.1 tipe data bilangan bulat .....	11
Tabel 2.2 tipe data bilangan pecahan .....	11
Tabel 2.3 karakter khusus .....	12
Tabel 2.4 daftar keyword java.....	15

# **BAB I**

## **TINJAUAN PUSTAKA**

### **A. Pengertian**

#### **1. Pengertian Bahasa Java**



Gambar 1.1 Logo 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. Kelebihan dan Kekurangan Bahasa Java**

### **a. Kelebihan**

1. Multiplatform. Kelebihan utama dari Java ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / bytecode) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan di atas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan bytecode tersebut.
2. OOP (Object Oriented Programming-Pemrogram Berorientasi Objek). Java merupakan salah satu bahasa pemrograman dengan konsep OOP. Dimana program yang dibangun berorientasikan kepada Object. Aplikasi yang dibangun dengan konsep OOP terdiri atas object-object yang saling berhubungan.

3. Perpustakaan Kelas Yang Lengkap Java terkenal dengan kelengkapan library/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.
4. Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.
5. Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

#### **b. Kekurangan**

1. Tulis sekali, jalankan di mana saja - Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk J2SE, misalnya SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.

2. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/direverse-engineer.
3. Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berlutut dengan mesin komputer berumur lebih dari 4 tahun.

### C. Sejarah Bahasa Java

Konsep awal bahasa Java dikembangkan pertama kali pada tahun 1991 oleh perusahaan **Sun Microsystems**. Beberapa programmer di perusahaan tersebut merasa bahasa C dan C++ (yang saat itu sedang populer), tidak lagi mencukupi untuk project yang sedang mereka kerjakan.

Sun Microsystems kemudian membuat tim kecil yang bernama **Green Project**. Tujuannya adalah untuk mengembangkan bahasa pemrograman baru yang lebih baik dari C++. Beberapa anggota penting dari tim tersebut diantaranya **James Gosling**, **Mike Sheridan** dan **Patrick Naughton**. Ketiga programmer inilah yang nantinya dinobatkan sebagai co-founder dari bahasa Java.

Bahasa baru ini pada awalnya bernama bahasa **Oak**, yang terinspirasi dari pohon Oak yang tumbuh di depan kantor James Gosling. Bahasa Oak ditujukan untuk perangkat televisi interaktif, namun dianggap terlalu canggih untuk teknologi TV kabel pada saat itu. Di tahun 1993, Green Project mengalihkan fokus utama pengembangan bahasa Oak ke teknologi yang disebut dikenal sebagai “internet”.

Akhirnya di tahun 1995, Sun Microsystems secara resmi merilis bahasa pemrograman ini sebagai bahasa “**Java**“, karena ternyata “Oak” sudah terdaftar sebagai merk dagang di perusahaan lain, yakni “**Oak Technologies**“.

Pemilihan bahasa “**Java**” terinspirasi dari “**Java coffee**“, atau kopi jawa. Karena itulah logo bahasa Java berupa gambar secangkir kopi. Konsep dasar dari bahasa Java sendiri banyak terinspirasi dari bahasa C dan C++, sehingga jika anda sudah familiar dengan salah satu bahasa ini, kode program di Java terasa sangat mirip.

Dalam perjalanannya, perusahaan Sun Microsystem di akuisisi oleh **Oracle** pada tahun 2010. Sehingga “pemilik” dari bahasa Java saat ini adalah perusahaan Oracle. Meskipun begitu, mayoritas teknologi yang ada di bahasa Java berlisensi open source (**GNU General Public License**), sehingga bisa dipakai dengan gratis.

Bahasa Java terbagi ke dalam beberapa bidang teknologi sesuai dengan target sistem yang dirancang. Diantaranya:

a. **Java Micro Edition (Java ME)**: ditujukan untuk membuat aplikasi di perangkat kecil (embedded system) dengan komputasi yang terbatas, seperti kalkukator, tv box, jam tangan, dll.

b. **Java Standard Edition (Java SE)**: ditujukan untuk perangkat komputer desktop atau server.

c. **Java Enterprise Edition (Java EE)**: ditujukan untuk perangkat besar serta aplikasi internet.

#### **D. Tujuan Praktikum**

Praktikum ini memiliki beberapa tujuan yaitu:

1. Agar mahasiswa lebih memahami tentang bahasa Java
2. Agar memahami dasar-dasar bahasa Java untuk membuat program menghitung mesin kasir Polkam Mart.
3. Untuk memahami sourcode yang digunakan pada pembuatan program kasir Polkam Mart.

#### **E. Alat Dan Bahan**

1. Alat
  - a) Laptop
2. Bahan
  - a) Sublime Text
  - b) CMD
  - c) Program Perhitungan Luas Bangun Datar.

## BAB II

### PEMBAHASAN

#### A. DASAR-DASAR ALGORITMA PEMOGRAMAN

##### 1. Program Hello World

Ok, untuk memulai perjalanan kita bersama Java, mau tak mau Anda memang harus menguasai sebuah hal yang tidak praktis saat pertama kali bertemu dengan Java. Tidak seperti bahasa pemrograman lain yang sederhana untuk memulai *hello world* yang hanya cukup dengan menggunakan *function* untuk mencetak sebuah *string* “Hello World!”, di Java Anda harus membuat terlebih dahulu sebuah *class* dan membuat sebuah *method* yang merupakan *method* utama dari kode program. *Method* utama yang akan dieksekusi tersebut dinamakan dengan *method* `main()`. *Method* tersebut wajib ada di salah satu *class* dari kode program yang kita bangun menggunakan Java.

Karena Java merupakan salah satu bahasa pemrograman yang berparadigma berorientasi objek, Anda memang harus mulai membiasakan diri dengan istilah - istilah seperti *inheritance*, *attribute*, *instantiation*, dan lainnya. Sekarang kita akan memulainya dengan membuat sebuah *file* yang bernama `HelloWorld.java`. Kemudian buat kode berikut di dalam *file* tersebut:

```

public class HelloWorld {

    public static void main(String[] args) {
        System.out.println("Hello World...");
    }
}

```

Gambar 2.1 program Hello World

Ada yang perlu Anda perhatikan dari kode diatas:

- a. `public` adalah sebuah *keyword* di Java yang menandakan bahwa objek, *method*, atau atribut dapat diakses dari *class* lain.
- b. `static` adalah sebuah *keyword* untuk membuat sebuah *method* tidak perlu diinstansiasi terlebih dahulu
- c. `void` adalah sebuah *keyword* untuk membuat sebuah *method* tidak me-*return* nilai apapun alias kosong
- d. `System.out.println()` adalah sebuah *method* yang telah di-*import* otomatis untuk digunakan mencetak *output* di konsol.
- e. Nama *class* dan nama *file* harus sama.
- f. **Class** adalah entitas yang menggambarkan keadaan dan perilaku dari objek(logis). Artinya, Class adalah **Cetakan** yang berisi serangkaian perintah untuk membangun jenis objek tertentu. `class` adalah sebuah *keyword* di Java yang digunakan untuk membuat sebuah *clas*. Class juga bisa diartikan sebagai **sekelompok objek** yang memiliki sifat umum. Contoh dari Class pada kehidupan nyata adalah kendaraan, manusia, hewan, dan lain sebagainya. Misalnya pada class kendaraan bisa memiliki objek seperti mobil, sepeda motor, pesawat, dll. Sebuah Class bisa memiliki salah satu dari jenis variabel seperti Variabel lokal, variabel instance, dan

variabel kelas. Class akan mendeklarasikan tipe data baru dan dijalankan sebagai cetakan, dimana model dari objek yang dibuat berdasarkan tipe data tersebut.

- g. **Objek** adalah turunan atau hasil dari suatu class, Java sendiri adalah Bahasa Pemrograman yang murni berbasis Objek, sehingga konsep pemrograman dengan prosedural tidak dikenal oleh java. Contoh objek bisa Anda lihat di kehidupan sehari-hari seperti pesawat, mobil, kucing, dll. Semua Objek memiliki **state** dan **behavior**, state adalah keadaan sedangkan behavior adalah perilaku. Misalkan objek dalam kehidupan nyata adalah Fizi(nama manusia) yang memiliki keadaan(nama, alamat, umur) dan perilaku(tidur, makan, melompat). Objek dalam perangkat lunak juga memiliki keadaan dan perilaku. Keadaan pada perangkat lunak disimpan dalam field(variabel) dan perilakunya ditunjukkan melalui method-method.
- h. **Konstruktor** adalah method yang memiliki aturan nama yang harus sama dengan nama class. Setiap Class yang dibuat memiliki Konstruktor, jika Anda tidak membuat konstruktor pada class, secara otomatis Compiler Java akan membuat konstruktor default. Dalam konstruktor, Anda tidak boleh memakai keyword void. Selain itu, Anda juga bisa membuat konstruktor lebih dari satu dan menjalankannya setiap kali suatu objek baru dibuat. Suatu Konstruktor java bisa memiliki parameter ataupun tidak. Berikut adalah sintaks untuk Membuat Konstruktor Pada Java dengan menggunakan parameter ataupun tidak.



```
public class Kendaraan{  
    // Konstruktor Tanpa Parameter  
    public Kendaraan(){  
    }  
    // Konstruktor Dengan Parameter  
    public Kendaraan(String jenis){  
    }  
}
```

Gambar 2.2 contoh program untuk materi konstruktor

## 2. Tipe Data dan Variabel

Bahasa pemrograman pada umumnya, mengenal adanya variabel yang digunakan untuk menyimpan nilai atau data. Sedangkan Java sendiri dikenal sebagai bahasa pemrograman dengan sifat strongly typed yang artinya diharuskan mendeklarasikan tipe data dari semua variabel, dan apabila lupa atau salah mengikuti aturan pendeklarasian variabel, maka akan mendapatkan error pada saat proses kompilasi.

### a. Tipe Data

Java memiliki tipe data yang dapat dikategorikan menjadi dua kelompok, yaitu tipe data primitif dan referensi.

1. **Tipe Data Primitif** Delapan macam tipe data primitif dalam pemrograman Java, yaitu :

a. Integer ( Bilangan Bulat )

Integer merupakan tipe data numerik yang digunakan apabila tidak berurusan dengan pecahan atau bilangan desimal. Tipe data numerik yang termasuk integer adalah sebagai berikut :

Tabel 2.1 tipe data bilangan bulat

<b>Tipe</b>	<b>Deskripsi</b>
<b>Byte</b>	Memiliki nilai integer dari -128 sampai +127 dan menempati 1 byte (8 bits) di memori
<b>Short</b>	Memiliki nilai integer dari -32768 sampai 32767 dan menempati 2 byte (16 bits) di memori
<b>Int</b>	Memiliki nilai integer dari -2147483648 sampai 2147483647 dan menempati 4 byte (32 bits) di memori
<b>Long</b>	Memiliki nilai integer dari -9223372036854775808 sampai 9223372036854775807 dan menempati 8 byte (64 bits) di memori

Bilangan integer biasanya menggunakan int, dan bukan byte, short maupun long. Bilangan integer juga mengenal nilai positif dan negatif ( signed number ). Tipe data byte dan short hanya digunakan pada aplikasi khusus yang memperhatikan penggunaan memori. Sedangkan long jarang digunakan karena jarang memerlukan bilangan sebesar kapasitas long.

b. Floating Point ( Bilangan Pecahan )

Floating Point digunakan untuk menangani bilangan desimal atau perhitungan yang lebih detail dibanding integer. Ada dua macam floating point, yaitu :

Tabel 2.2 tipe data bilangan pecahan

<b>Tipe</b>	<b>Deskripsi</b>
<b>Float</b>	memiliki nilai $-3.4 \times 10^8$ sampai $+3.4 \times 10^8$ dan menempati 4 byte di memori
<b>Double</b>	memiliki nilai $-1.7 \times 10^{308}$ sampai $+1.7 \times 10^{308}$

Semua bilangan pecahan atau desimal dalam Java tanpa diakhiri huruf f akan dianggap sebagai double. Sedangkan bilangan yang ingin dikategorikan sebagai float harus diakhiri dengan huruf F. Misalnya : 4.22 F atau 2.314f. Sedangkan untuk bilangan double, bisa menambah dengan huruf D, karena secara default bilangan dengan koma atau pecahan atau desimal akan dianggap sebagai double.

#### c. Char

Char adalah karakter tunggal yang didefinisikan dengan diawali dan diakhiri dengan tanda ' ( petik tunggal ). Char berbeda dengan String, karena String bukan merupakan tipe data primitif, tetapi sudah merupakan sebuah objek. Tipe char mengikuti aturan unicode, sehingga dapat menggunakan kode /u kemudian diikuti bilangan dari 0 sampai 65535, tetapi yang biasa digunakan adalah bilangan heksadesimal dari 0000 sampai FFFF. Misalnya : '\u123'.

Selain karakter biasa, juga terdapat karakter khusus yang didefinisikan dengan cara mengawalinya menggunakan tanda \ seperti pada tabel berikut :

Tabel 2.3 karakter khusus

Kode	Nama	Nilai Unicode
\b	Backspace	\u0008
\t	Tab	\u0009
\n	Linefeed	\u000a
\r	Carriage return	\u000d
\*	Double quote	\u0022
\'	Single quote	\u0027
\\	Backslash	\u005c

#### d. Boolean

Dalam Java dikenal tipe data boolean yang terdiri dari dua nilai saja, yaitu true dan false. Boolean sangat penting dalam mengevaluasi suatu kondisi, dan sering digunakan untuk menentukan alur program.

#### 2. Tipe Data Referensi

Kelebihan pemrograman berorientasi objek adalah dapat mendefinisikan tipe data baru yang merupakan objek dari class tertentu. Tipe data ini digunakan untuk mereferensikan objek atau class tertentu, seperti String.

#### **Latihan 2. TipeData.**

```
java class TipeData {  
    public static void main(String[] args) { // Tipe data primitif  
  
        long data1 = 546767226531;  
        int data2 = 2235641;  
        short data3 = 714;  
        byte data4 = 34;  
        float data6 = (float) 1.733; // tipe data pecahan  
        double data5 = 4.967; // tipe data pecahan  
        char data7 = 'C';  
        boolean data8 = true;  
  
        System.out.println("Nilai Long : "+ data1);  
        System.out.println("Nilai Int : "+ data2);  
        System.out.println("Nilai Short : "+ data3);  
        System.out.println("Nilai Byte : "+ data4);  
        System.out.println("Nilai Double : "+ data5);  
        System.out.println("Nilai Float : "+ data6);  
        System.out.println("Nilai Char : "+ data7);  
        System.out.println("Nilai Boolean : "+ data8);  
    } }
```

Gambar 2.3 contoh program untuk tipe data

Outputnya:



```
ca Command Prompt
G:\JAVUA>javac TipeData.java
G:\JAVUA>java TipeData
Nilai Long : 226531
Nilai Int : 2235641
Nilai Short : 714
Nilai Byte : 34
Nilai Double : 4.967
Nilai Float : 1.733
Nilai Char : C
Nilai Boolean : true
```

Gambar 2.4 hasil compile gambar program tipe data

## b. Variabel

Variabel merupakan container yang digunakan untuk menyimpan suatu nilai pada sebuah program dengan tipe tertentu. Untuk mendefinisikan variabel, kita dapat menggunakan identifier untuk menamai variabel tersebut.

### 1. Identifier

Identifier adalah kumpulan karakter yang dapat digunakan untuk menamai variabel, method, class, interface, dan package. Sebagaimana bahasa pemrograman pada umumnya, Java memiliki peraturan untuk identifier yang valid atau sah. Identifier bisa disebut valid atau sah apabila diawali dengan :

- Huruf / abjad
- Karakter mata uang
- Underscore ( \_ )

Identifier dapat terdiri dari :

- Huruf / abjad
- Angka
- Underscore ( \_ )

Identifier tidak boleh mengandung @, spasi atau diawali dengan angka. Selain itu, identifier tidak boleh menggunakan keyword atau kata-kata yang memiliki arti atau digunakan dalam pemrograman Java. Daftar Keyword Java :

Tabel 2.4 daftar keyword java

Abstract	Double	Int	Strictfp
Boolean	False	Static	Super
Break	Extends	long	Switch
Byte	Final	Native	Synchronized
Case	Finally	New	This
Catch	Float	Package	Throw
Char	For	Private	Throws
Class	Goto	Protected	Transient
Const	If	Public	Try
Continue	Implements	Return	Void
Default	Import	Short	Volatile
do	Instanceof	interface	While

Selain menggunakan karakter biasa, kita juga dapat menggunakan unicode sebagai identifier.

#### a. Mendeklarasikan Variabel

##### 1. Sintaks dasar :

[tipe data] [nama variabel]

Menuliskan tipe data dari variabel, contoh :

int bilangan;

char karakter;

float bilangan desimal;

boolean status;

##### 2. Setelah mendeklarasikan variabel dengan tipe data, selanjutnya memberikan nilai variabel tersebut dengan tanda = .

bilangan = 20;

karakter = 'k';

bilDesimal = 22.2f;

status = true;

3. Dapat juga mendeklarasikan dan memberikan nilai dalam satu baris.

```
int bilangan = 20; char karakter = 'k'; float bilesimal = 22.2f; boolean status = true;
```

4. Kita dapat membuat variabel menjadi konstanta yang tidak dapat diubah nilainya dengan menambahkan keyword sebelum tipe data dari variabel. Contoh :

```
final int konstanta integer = 10;
```

```
final float pajak = 15.5;
```

5. Agar konstanta ini dapat diakses oleh class lain tanpa harus membuat objek terlebih dahulu, maka kita dapat menambahkan modifier public dan keyword static seperti berikut :

```
public static final konstantainteger = 10;.
```

### 3. Percabangan

Percabangan adalah suatu pilihan atau opsi dimana terdapat kondisi tertentu yang harus dipenuhi oleh program untuk menjalankan suatu perintah, jika kondisi atau syarat itu terpenuhi maka program akan menjalankan perintahnya tetapi jika tidak maka program tidak akan menjalankan perintahnya atau melewatinya dan melihat kondisi lainnya untuk dijalankan atau berhenti.berikut kita akan membahas jenis percabangan yaitu if, else if, if else. Langsung saja kita ke pembahasannya.

#### a. Fungsi IF

Percabangan if ini akan digunakan hanya pada satu kondisi atau satu perintah yang akan dijalankan program ketika kondisi atau syarat terpenuhi,namun jika tidak maka akan diabaikan . If ini merupakan percabangan dengan statement atau kondisi yang sifatnya satu. Artinya ketika dia sampai pada kondisi yang diminta, dia akan melihat apakah kondisi tersebut terpenuhiatau tidak. Jika

tidak maka dia akan mengabaikannya dan program akan langsung keluar atau berhenti, namun jika kondisi terpenuhi atau ya, maka akan dijalankan secara otomatis.

contoh :

```
1  public class Percabangan_if{
2
3  public static void main(String [] args) {
4
5      int a = 1;
6
7      int b = 2;
8
9      if ( a < b ) { //jika a lebih kecil dari b tampilkan kata "Program Dijalankan"
10
11          System.out.println("Program Dijalankan");
12
13      }
14  }
15
16 }
```

Gambar 2.5 contoh program untuk percabangan IF

Hasil Outputnya adalah :

```
E:\>javac Percabangan_if.java

E:\>java Percabangan_if
Program Dijalankan

E:\>_
```

Gambar 2.6 hasil compile dari gambar percabangan IF

## b. Fungsi If else

Percabangan IF Else Ini digunakan untuk percabangan dalam dua kondisi . Contohnya yaitu dari satu kondisi yang tidak terpenuhi, maka dia akan secara otomatis menjalankan yang kedua atau perintah kedua tanpa harus mengoreksi benar atau salah. artinya dia akan mengecek apakah kondisi pertama terpenuhi atau tidak. Jika ya maka dia akan menjalankan kondisi kedua tanpa mengecek apakah kondisi tersebut terpenuhi atau tidak.

Contohnya :



```

1 public class Percabangan_ifelse{
2
3 public static void main(String [] args) {
4
5     int a = 1;
6     int b = 2;
7
8     if ( a > b ) {
9         /*jika a lebih besar dari b tampilkan kata "Program Dijalankan"
10         maka selain itu tampilkan kata "Program yang kedua dijalankan"
11         */
12         System.out.println("Program Dijalankan");}
13
14     else {
15         System.out.println("Program yang kedua dijalankan");
16     }
17 }
18 }
19
20 }

```

Gambar 2.7 contoh program untuk fungsi if else

Hasil Outputnya :

```

E:\>javac Percabangan_ifelse.java

E:\>java Percabangan_ifelse
Program yang kedua dijalankan

E:\>

```

Gambar 2.8 Hasil compile gambar contoh program untuk fungsi if else

Maka hasil yang keluar adalah kata "Program yang kedua dijalankan" karena variabel a lebih kecil dari b sedangkan kondisi yang di minta di fungsi if nya adalah variabel a lebih besar dari b.

### c. Funsif else If

Else If merupakan percabangan yang bisa digunakan untuk banyak kondisi atau persyaratan dengan perintah yang juga tentunya banyak. Contohnya untuk kasus pemberian predikat nilai dari siswa siswi. Katakanlah ada score dari 0 – 100 dan predikat dari E sampai terbaik adalah A, maka dapat menggunakan percabangan ini.

Contohnya :

```
1  import java.util.*;
2  public class Percabangan_elseif{
3  public static void main(String [] args) {
4  int nilai;
5  Scanner nilaiujian=new Scanner(System.in);
6  System.out.println("Masukkan Nilai= ");
7  nilai=nilaiujian.nextInt();
8  if(nilai<= 20){
9  System.out.println("Predikat Anda E");
10 /*jika variabel lebih kecil dari 20
11 tampilkan "Predikat Anda E"
12 */
13 else if(nilai <= 40){
14 System.out.println("Predikat Anda D");
15 /*jika variabel lebih kecil dari 40
16 tampilkan "Predikat Anda D"
17 */
18 else if(nilai <= 60){
19 System.out.println("Predikat Anda C");
20 /*jika variabel lebih kecil dari 60
21 tampilkan "Predikat Anda C"
22 */
23 else if(nilai <= 80){
24 System.out.println("Predikat Anda B");
25 /*jika variabel lebih kecil dari 80
26 tampilkan "Predikat Anda B"
27 */
28 else if(nilai <= 100){
29 System.out.println("Predikat Anda A");
30 /*jika variabel lebih kecil dari 100
31 tampilkan "Predikat Anda A"
32 */
33 else{
34     System.out.println("Maaf sudah tidak ada predikat selain A");
35 }
36 /*jika kondisi diatas semua tidak terpenuhi maka tampilkan "Maaf sudah tidak ada predikat selain A"
37 */
38 }}
```

Gambar 2.9 Gambar contoh program untuk fungsi Else If

Hasil Outputnya :

```
E:\>javac Percabangan_elseif.java
E:\>java Percabangan_elseif
Masukkan Nilai=
10
Predikat Anda E
E:\>javac Percabangan_elseif.java
E:\>java Percabangan_elseif
Masukkan Nilai=
30
Predikat Anda D
E:\>javac Percabangan_elseif.java
E:\>java Percabangan_elseif
Masukkan Nilai=
55
Predikat Anda C
E:\>javac Percabangan_elseif.java
E:\>java Percabangan_elseif
Masukkan Nilai=
75
Predikat Anda B
E:\>javac Percabangan_elseif.java
E:\>java Percabangan_elseif
Masukkan Nilai=
90
Predikat Anda A
E:\>javac Percabangan_elseif.java
E:\>java Percabangan_elseif
Masukkan Nilai=
105
Maaf sudah tidak ada predikat selain A
```

Gambar 2.10 Hasil compile gambar contoh program untuk fungsi Else If

#### d. Switch

Kondisi SWITCH CASE adalah percabangan kode program dimana kita membandingkan isi sebuah variabel dengan beberapa nilai. Jika proses perbandingan tersebut menghasilkan true, maka block kode program akan di proses.

Kondisi SWITCH CASE terdiri dari 2 bagian, yakni perintah SWITCH dimana terdapat nama variabel yang akan diperiksa, serta 1 atau lebih perintah CASE untuk setiap nilai yang akan diperiksa.

Berikut ini bentuk dasar dari switch :

```
public class percabangan2{  
    public static void main(String []args){  
        int x = 10;  
        switch(x){  
            case 1 : System.out.println("satu"); break;  
            case 2 : System.out.println("dua"); break;  
            default : System.out.println("bukan satu atau dua"); } } }
```

Gambar 2.11 Gambar contoh program Switch case

## 4. Perulangan

### a. 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;  
}
```

Gambar 2.12 Bentuk umum dari perulangan for

#### Keterangan:

- **Nilai awal** : Bagian dimana kita menginisialisasi nilai awal dimana menjadi titik awal perulangan dimulai.
- **Kondisi** : Bagian untuk menetapkan suatu kondisi tertentu, ketika kondisi bernilai benar perulangan akan terus berjalan.
- **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+", ");  
        }  
    }  
}
```

Gambar 2.13 Contoh program dari perulangan for

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**.

**b. Perulangan While Pada Java**

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

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

Gambar 2.14 Bentuk Umum dari perulangan While

Keterangan:

- Kondisi – Sebagai sebuah syarat untuk menghentikan proses perulangan
- 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();  
        }  
    }  
}
```

Gambar 2.15 Contoh program dari perulangan While

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

Terdapat variabel bilangan dengan tipe data integer dan nilai awal adalah 1. Kemudian ada perulangan while dengan kondisi **bilangan!=0** syarat ini berarti jika nilai yang terdapat di dalam variabel bilangan isi-nya bukan 0 maka perulangan akan terus dilakukan sampai variabel bilangan harus berisi nilai 0. Perulangan dilakukan dengan mencetak kalimat “Masukan Bilangan 0 : ” dimana user akan memasukan bilangan, selama bilangan yang dimasukan oleh pengguna bukanlah 0 program ini akan terus diulang-ulang. Program akan berhenti ketika pengguna memasukan bilangan 0. Lebih jelasnya lihat hasil output dibawah ini:

```
run:
Masukkan Bilangan 0 : 1
Masukkan Bilangan 0 : 4
Masukkan Bilangan 0 : 5
Masukkan Bilangan 0 : 6
Masukkan Bilangan 0 : 7
Masukkan Bilangan 0 : 0
BUILD SUCCESSFUL (total time: 2 minutes 45 seconds)
```

Gambar 2.16 Hasil Compile program dari perulangan While

Coba perhatikan pada hasil output diatas pada iterasi ke-1 saya memasukan bilangan 1, program akan mengecek apakah 1 tidak sama dengan 0 ? Jawabannya iya benar maka perulangan di lakukan kembali, pada iterasi ke-2 saya memasukan bilangan 4, dicek lagi apakah 4 tidak sama dengan 0 ? iya benar maka program akan diulang terus hingga pada iterasi ke-6 saya memasukan bilangan 0, sehingga ketika dicek apakah 0 tidak sama dengan 0 ? jawabannya salah sehingga kondisi bernilai **false** maka perulangan while saat itu juga dihentikan.

### c. 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);
```

Gambar 2.17 Bentuk Umum dari perulangan Do While

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);  
    }  
}
```

Gambar 2.18 Contoh program dari perulangan Do While



### Hasil Output:

1

Coba perhatikan contoh program diatas, saya membuat variabel bilangan dengan nilai 1. Kemudian selanjutnya terdapat **perulangan do while**, dimana program akan mencetak terlebih dahulu nilai pada variabel bilangan lihat pada : **System.out.println(bilangan);** nilai pada variabel tersebut adalah 1 dan nilai tersebut dicetak, lalu baru setelah itu di cek kondisi apakah nilai pada variabel bilangan dalam hal ini apakah 1 lebih besar dari atau sama dengan 5 ? Jawabannya salah (*false*) sehingga perulangan dihentikan. Namun karena perulangan do while dikerjakan terlebih dahulu makanya nilai 1 telah dicetak sebelum pengecekan kondisi dilakukan.

## 5. Sorting (Pengurutan)

Algoritma Sorting merupakan kumpulan langkah sistematis untuk melakukan pengurutan sejumlah data berdasarkan nilai tertentu. Pengurutan dapat dilakukan dari nilai terkecil ke nilai terbesar (ascending) ataupun sebaliknya.

Comparison Sort atau pengurutan dengan perbandingan merupakan algoritma sorting yang dalam proses pengurutannya melakukan perbandingan antar data. Berikut ini saya berikan contoh implementasi algoritma Bubble Sort, Selection Sort dan Insertion Sort.

### a. Bubble Sort

Kita bisa membuat program java untuk mengurutkan element array menggunakan bubble sort. **algoritma bubble sort** dikenal sebagai algoritma sorting yang paling sederhana. Dalam **algoritma bubble sort**, setiap array dilalui dari elemen pertama sampai elemen terakhir. Di sini, element saat ini dibandingkan dengan element berikutnya. Jika element saat ini lebih besar dari element berikutnya, itu bertukar.

### **b. Selection Sort**

Kita bisa membuat program java untuk mengurutkan element array menggunakan **selection sort**. Dalam **algoritma selection sort**, kita mencari element terendah dan mengaturnya ke lokasi yang tepat. Kemudian menukar element saat ini dengan jumlah terendah berikutnya.

### **c. Insertion Sort**

Kita bisa membuat program java untuk mengurutkan element array menggunakan **insertion sort**. Penyisipan baik untuk elemen kecil hanya karena membutuhkan lebih banyak waktu untuk menyortir sejumlah besar elemen.

## BAB III

### LANGKAH KERJA MENJALANKAN PROGRAM

#### A. Instalasi Java Jdk

Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus di disain untuk memanfaatkan dependensi implementasi seminimal mungkin.

Sehingg komputer mengerti apa yang dimaksud dengan bahasa java. Java jdk adalah salah satu compiler bahasa Java

1. Download java Jdk
  - a. Nyalakan komputer atau laptop, setelah itu beri koneksi internet.
  - b. Buka browser dan carilah compiler java Jdk
  - c. Setelah itu download

2. Mengekstrak Java Jdk

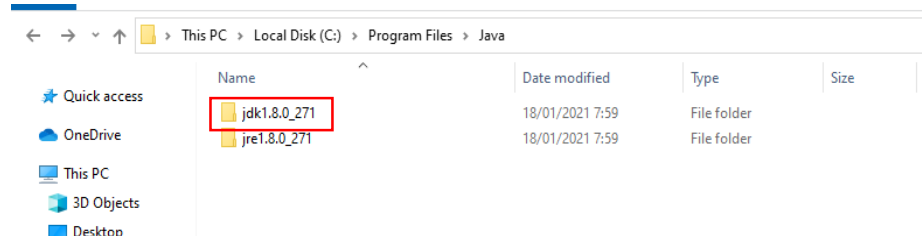
Berikut adalah langkah untuk mengekstrak Java Jdk

- a. Setelah Java jdk berhasil terdownload lalu buka
- b. Kemudian klik ekstrak to dan pilihlah lokasi penyimpanan kemudian kelik “ok”. Setelah itu akan muncul file baru jdk yang berisi folder bin, include, lib, libexec.

3. Instal Java Jdk

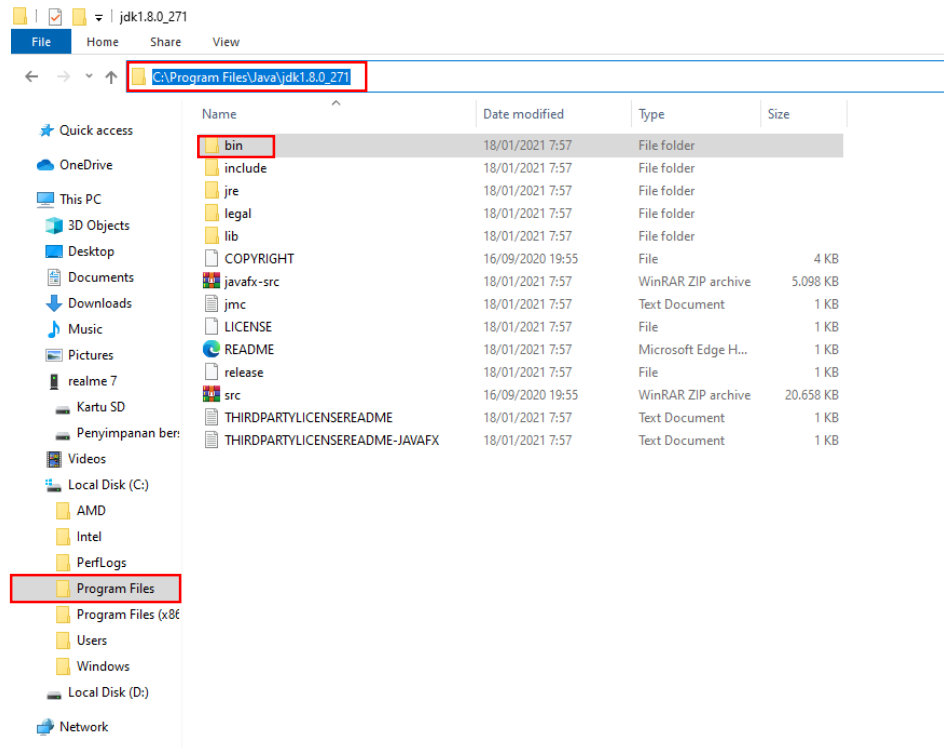
Berikut langkah penginstalan Java Jdk

- a. Langkah awal penginstalan Java Jdk adalah masuk ke file yang sudah di ekstrak contohnya di local disk C.



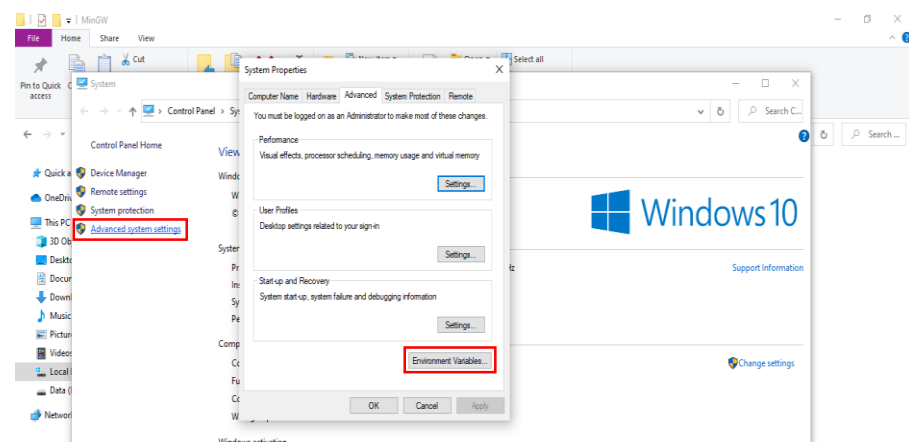
Gambar 3.1 tampilan file jdk

- b. Setelah masuk pilih folder Java Jdk kemudian cari folder “bin” kemudian klik sekali dan copy link nya, seperti gambar dibawah ini.



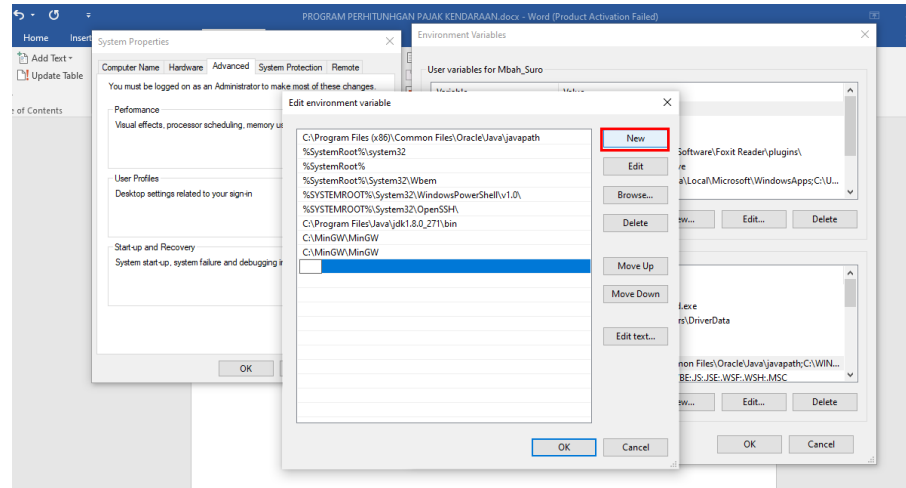
Gambar 3.1 Tampilan Folder Bin

- c. Kemudian masuk ke this Pc kemudian klik kanan dan pilih propertis  
d. Lalu pilih Advanced System Setting, kemudian pada sistem propertis pilih “environment variabels”



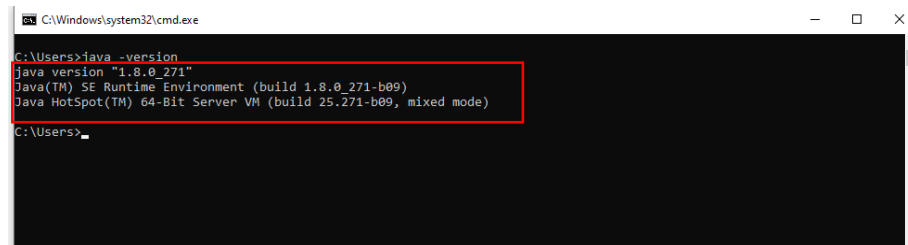
Gambar 3.2 Tampilan System

- e. Kemudian lihat di sistem variabel cari “path” kemudian klik new dan pastekan link tadi yang sudah di copy kemudian ok



Gambar 3.3 Tampilan Environment Variabel

- f. Untuk menguji compiler nya sudah terinstal cek di cmd dengan tekan win+R kemudian masuk setelah itu ketik java spasi -version untuk melihat versi compiler yang sudah d instal, dari gambar dibawah versi Java adalah 1.8.0\_271.

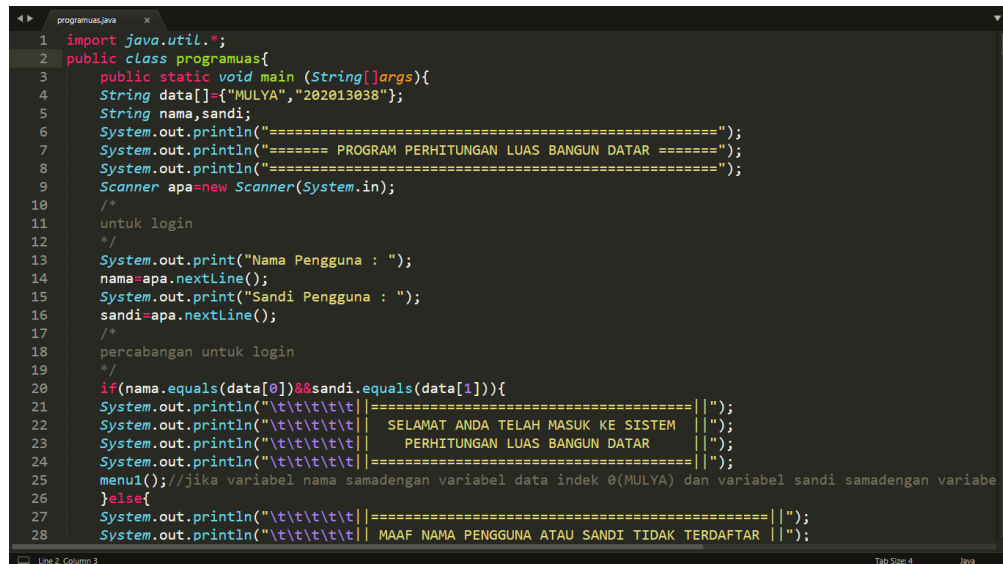


Gambar 3. 4 Tampilan Versi Java

## B. Program Perhitungan Luas Bangun Datar

Berikut adalah contoh program Aplikasi java menggunakan bahasa java.

### 1. Tampilan Awal Program



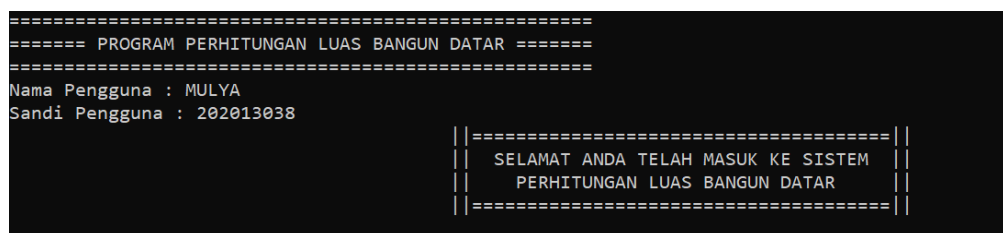
```
1 import java.util.*;
2 public class programuas{
3     public static void main (String[] args){
4         String data[]={"MULYA","202013038"};
5         String nama,sandi;
6         System.out.println("=====");
7         System.out.println("===== PROGRAM PERHITUNGAN LUAS BANGUN DATAR =====");
8         System.out.println("=====");
9         Scanner apa=new Scanner(System.in);
10        /*
11        untuk login
12        */
13        System.out.print("Nama Pengguna : ");
14        nama=apa.nextLine();
15        System.out.print("Sandi Pengguna : ");
16        sandi=apa.nextLine();
17        /*
18        percabangan untuk login
19        */
20        if(nama.equals(data[0])&&sandi.equals(data[1])){
21            System.out.println("\t\t\t\t\t|=====||");
22            System.out.println("\t\t\t\t\t| SELAMAT ANDA TELAH MASUK KE SISTEM ||");
23            System.out.println("\t\t\t\t\t| PERHITUNGAN LUAS BANGUN DATAR ||");
24            System.out.println("\t\t\t\t\t|=====||");
25            menu1();//jika variabel nama samadengan variabel data indek 0(MULYA) dan variabel sandi samadengan variabe
26        }else{
27            System.out.println("\t\t\t\t\t|=====||");
28            System.out.println("\t\t\t\t\t| MAAF NAMA PENGGUNA ATAU SANDI TIDAK TERDAFTAR ||");
```

Gambar 3.6 Tampilan awal Program

Pembahasan :

Gambar di atas merupakan tampilan awal dari program perhitungan luas bangun datar. Di line ke 1 merupakan console import java.util.\* untuk fungsi scanner. Lalu di line ke 2 terdapat penamaan kelas untuk program tersebut dengan nama (programuas). Lalu di line ke 3 ada method utama dari program dan di line ke 4-5 dilanjutkan dengan pendeklarasian dari variable data[], nama, dan sandi, dengan tipe data String. Lalu di line ke 9 pembuatan variable baru untuk menyimpan nilai scanner dari variable lain. Nama pengguna yaitu variable nama dan sandi pengguna yaitu variable sandi disimpan di variabel scanner apa. Lalu dilanjutkan di line 20-30 merupakan fungsi percabangan untuk membuat sistem login nya.

Outputnya :



```
=====
===== PROGRAM PERHITUNGAN LUAS BANGUN DATAR =====
=====
Nama Pengguna : MULYA
Sandi Pengguna : 202013038
|=====|
| SELAMAT ANDA TELAH MASUK KE SISTEM |
| PERHITUNGAN LUAS BANGUN DATAR |
|=====|
```

Gambar 3.7 Hasil Compile Program berhasil login

```

=====
===== PROGRAM PERHITUNGAN LUAS BANGUN DATAR =====
=====
Nama Pengguna : Mulya
Sandi Pengguna : 202013038

||=====||
|| MAAF NAMA PENGGUNA ATAU SANDI TIDAK TERDAFTAR ||
||=====||

```

Gambar 3.8 Hasil Compile Program tidak berhasil Login

2. Tampilan untuk Pilihan utama yaitu program atau keluar.

```

32 static void menu1(){
33     System.out.println();
34     String menu1[]={"Program","Keluar"};
35     System.out.println("=====");
36     System.out.println("|| BERIKUT PILIHAN MENU UTAMA ||");
37     System.out.println("=====");
38     System.out.println("\t1. "+menu1[0]);
39     System.out.println("\t2. "+menu1[1]);
40     Scanner pilihan1=new Scanner(System.in);
41     int kamu;
42     System.out.println();
43     System.out.print("Silahkan Pilih Menu : ");
44     kamu=pilihan1.nextInt();
45     if(kamu==1){
46         menu1();
47     }else{
48         System.exit(0);
49     }
50 }

```

Gambar 3.9 Tampilan Program untuk fungsi Menu1

Pembahasan :

Gambar diatas merupakan kodingan untuk menampilkan Pilihan menu utama yaitu pilihan 1 yaitu ke Program dan 2 yaitu Keluar dari program. Di line 32 merupakan method dari dengan nama menu1. Di line 33 merupakan Sout. Di line 34 yaitu pendeklarasian variable menu1 menggunakan array dengan tipe data String. Di line 38-39 untuk memanggil dan menampilkan variable array menu1. Di line 40 membuat variable scanner pilihan1. Line 41 deklarasi variable kamu dengan tipe data kamu. Di line 43-49 merupakan fungsi percabangan untuk memilih menu utama, jika memilih 1 yaitu program user akan di arahkan ke fungsi menu, namun jika user memilih 2 yaitu keluar maka sistem akan otomatis keluar dari program dan harus kembali login jika ingin menjalankan program perhitungan. Line 50 untuk menutup fungsi menu1.

Hasil Outputnya :

```
=====
||  BERIKUT PILIHAN MENU UTAMA  ||
=====

    1. Program
    2. Keluar

Silahkan Pilih Menu :
```

Gambar 3.10 Hasil Compile fungsi menu1

```
=====
||  BERIKUT PILIHAN MENU UTAMA  ||
=====
    1. Program
    2. Keluar

Silahkan Pilih Menu : 1

                                     =====
                                     || Berikut Pilihan Bangun Datar ||
                                     =====
                                     || 1 . luas PERSEGI           ||
                                     || 2 . luas PERSEGI PANJANG    ||
                                     || 3 . luas JAJARGENJANG       ||
                                     || 4 . luas SEGITIGA           ||
                                     || 5 . luas BELAH KETUPAT      ||
                                     || 6 . luas LAYANG-LAYANG      ||
                                     || 7 . luas TRAPESIUM          ||
                                     || 8 . luas LINGKARAN           ||
                                     || 9 . luas semua bangun datar ||
                                     || 10. keluar dari program    ||
                                     =====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||
```

Gambar 3.11 Hasil Compile fungsi menu1 pilihan 1 program

```
=====
||  BERIKUT PILIHAN MENU UTAMA  ||
=====

    1. Program
    2. Keluar

Silahkan Pilih Menu : 2

K:\>
```

Gambar 3.12 Hasil Compile fungsi menu1 pilihan 2 keluar



3. Tampilan untuk menu pemilihan Bangun datar.

```
51 static void menu(){ //fungsi untuk menu  
52     System.out.println();  
53     System.out.println("\t\t\t\t\t=====");  
54     System.out.println("\t\t\t\t\t|| Berikut Pilihan Bangun Datar ||");  
55     System.out.println("\t\t\t\t\t=====");  
56     System.out.println("\t\t\t\t\t| 1 . luas PERSEGI |"); // menu 1 merupakan pilihan mminuman  
57     System.out.println("\t\t\t\t\t| 2 . luas PERSEGI PANJANG |"); //menu 2 merupakan pilihan makanan  
58     System.out.println("\t\t\t\t\t| 3 . luas JAJARGENJANG |");  
59     System.out.println("\t\t\t\t\t| 4 . luas SEGITIGA |");  
60     System.out.println("\t\t\t\t\t| 5 . luas BELAH KETUPAT |");  
61     System.out.println("\t\t\t\t\t| 6 . luas LAYANG-LAYANG |");  
62     System.out.println("\t\t\t\t\t| 7 . luas TRAPESIUM |");  
63     System.out.println("\t\t\t\t\t| 8 . luas LINGKARAN |");  
64     System.out.println("\t\t\t\t\t| 9 . luas semua bangun datar |");  
65     System.out.println("\t\t\t\t\t| 10. keluar dari program |");  
66     System.out.println("\t\t\t\t\t=====");  
67     System.out.println();  
68     Scanner list=new Scanner(System.in); //buat variabel apa untuk fungsi Scannernya  
69     int menuutama;//variabel menu dengan tipe integer  
70     System.out.print("|| Silahkan pilih menu nomor berapa yang ingin di jalankan : "); //kita tampilkan kata"S  
71     System.out.print("||");  
72     menuutama=list.nextInt();
```

Gambar 3.13 tampilan program untuk fungsi menu bangun datar

```

77     if(menuutama==1){
78         persegi();}
79     else if(menuutama==2){
80         persegi panjang();}
81     else if(menuutama==3){
82         jajargenjang();}
83     else if(menuutama==4){
84         segitiga();}
85     else if (menuutama==5){
86         belahketupat();}
87     else if (menuutama==6){
88         layang layang();}
89     else if (menuutama==7){
90         trapesium();}
91     else if (menuutama==8){
92         lingkaran();}
93     else if (menuutama==9){
94         semuabangundatar();}
95     else {
96         System.exit(0);}
97 }

```

Gambar 3.14 tampilan program untuk fungsi menu bangun datar percabangan

Pembahasan :

Gambar di atas merupakan tampilan untuk memilih pilihan bangun datar apa yang ingin kita jalankan perhitungannya. Di line 51 merupakan method untuk menu. Di line 52-67 merupakan tampilan list dari bangun datar yang tersedia di program. Line 68 membuat variable scanner dengan nama list. Line 69 deklarasi variable menuutama dengan tipe data integer. Line 70-72 untuk memasukkan variable menuutama ke variable scanner list. Di line 77-97 merupakan fungsi percabangan yang apabila user memilih 1 maka akan langsung di arahkan ke menu persegi dan seterusnya, dan jika user memilih 10 maka user akan langsung keluar dari program.

Hasil Outputnya:

```
=====
|| Berikut Pilihan Bangun Datar ||
=====
|| 1 . luas PERSEGI           ||
|| 2 . luas PERSEGI PANJANG  ||
|| 3 . luas JAJARGENJANG     ||
|| 4 . luas SEGITIGA         ||
|| 5 . luas BELAH KETUPAT    ||
|| 6 . luas LAYANG-LAYANG    ||
|| 7 . luas TRAPESIUM        ||
|| 8 . luas LINGKARAN        ||
|| 9 . luas semua bangun datar ||
|| 10. keluar dari program   ||
=====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||1
=====
||                               ||
||      PERHITUNGAN LUAS PERSEGI      ||
||                               ||
=====

Masukkan Sisi = 4
Luas Persegi = 16.0
Apakah anda ingin mengitung kembali ? (YA/TIDAK)
```

Gambar 3.15 Hasil Compile fungsi menu pilihan 1 luas persegi

```
=====
|| Berikut Pilihan Bangun Datar ||
=====
|| 1 . luas PERSEGI           ||
|| 2 . luas PERSEGI PANJANG  ||
|| 3 . luas JAJARGENJANG     ||
|| 4 . luas SEGITIGA         ||
|| 5 . luas BELAH KETUPAT    ||
|| 6 . luas LAYANG-LAYANG    ||
|| 7 . luas TRAPESIUM        ||
|| 8 . luas LINGKARAN        ||
|| 9 . luas semua bangun datar ||
|| 10. keluar dari program   ||
=====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||10
K:\>
```

Gambar 3.16 Hasil Compile fungsi menu pilihan 1 keluar dari program

4. Tampilan untuk setiap menu bangun datar.

```

99 static void persegi(){
100     System.out.println("=====");
101     System.out.println("|| ||");
102     System.out.println("|| PERHITUNGAN LUAS PERSEGI ||");
103     System.out.println("|| ||");
104     System.out.println("=====");
105     System.out.println();
106     Scanner persegi=new Scanner(System.in);
107     Scanner plhn=new Scanner(System.in);
108     float sisi,luas;
109     String pilihan;
110     System.out.print("\tMasukkan Sisi = ");
111     sisi=persegi.nextFloat();
112     luas=sisi*sisi;
113     System.out.println("\tLuas Persegi = "+luas);
114     System.out.println("Apakah anda ingin mengitung kembali ? (YA/TIDAK) ");
115     pilihan=plhn.nextLine();
116     if (pilihan.equals("YA")){
117         menu();
118     }
119     if(pilihan.equals("Ya")){
120         menu();
121     }
122     if (pilihan.equals("yA")){
123         menu();
124     }
125     else{
126         System.out.println();
127         System.out.println();
128         System.out.println("=====");
129         System.out.println("|| TERIMA KASIH SUDAH MENGGUNAKAN PROGRAM MENGHITUNG LUAS BANGUN DATAR ||");
130         System.out.println("|| SEMOGA HARIMU MENYENANGKAN ||");
131         System.out.println("=====");
132     }
133 }

```

Gambar 3.17 Program untuk luas persegi

```

134 static void persegipanjang(){
135     System.out.println("=====");
136     System.out.println("|| ||");
137     System.out.println("|| PERHITUNGAN LUAS PERSEGI PANJANG ||");
138     System.out.println("|| ||");
139     System.out.println("=====");
140     System.out.println();
141     Scanner pp=new Scanner(System.in);
142     Scanner plhn=new Scanner(System.in);
143     float panjang,lebar,luaspp;
144     String pilihan;
145     System.out.print("\tMasukkan Nilai Panjang = ");
146     panjang=pp.nextFloat();
147     System.out.print("\tMasukkan nilai lebar = ");
148     lebar=pp.nextFloat();
149     luaspp=panjang*lebar;
150     System.out.println("\tLuas Persegi Panjang = "+luaspp);
151     System.out.println("Apakah anda ingin mengitung kembali ? (YA/TIDAK) ");
152     pilihan=plhn.nextLine();
153     if (pilihan.equals("YA")){
154         menu();
155     }
156     if(pilihan.equals("Ya")){
157         menu();
158     }
159     if (pilihan.equals("yA")){
160         menu();
161     }
162 }

```

Gambar 3.18 Program untuk luas persegi Panjang

```

171 static void jajargenjang(){
172     System.out.println("=====");
173     System.out.println("||");
174     System.out.println("||          PERHITUNGAN LUAS JAJARGENJANG          ||");
175     System.out.println("||");
176     System.out.println("=====");
177     System.out.println();
178     Scanner pp=new Scanner(System.in);
179     Scanner jjg=new Scanner(System.in);
180     float luasjjg,a,t;
181     String pilihan;
182     System.out.print("\tMasukkan nilai alas Jajargenjang = ");
183     a=pp.nextFloat();
184     System.out.print("\tMasukkan nilai tinggi Jajargenjang = ");
185     t=pp.nextFloat();
186     luasjjg=a*t;
187     System.out.println("\tLuas Jajargenjang = "+luasjjg);
188     System.out.println("Apakah anda ingin mengitung kembali ? (YA/TIDAK) ");
189     pilihan=jjg.nextLine();
190     if (pilihan.equals("YA")){
191         menu();
192     }
193     if(pilihan.equals("Ya")){
194         menu();
195     }
196     if (pilihan.equals("yA")){
197         menu();
198     }
199     if (pilihan.equals("ya")){
200         menu();
201     }
202 }

```

Gambar 3.19 Program untuk luas jajargejang

```

208 static void segitiga(){
209     System.out.println("=====");
210     System.out.println("||");
211     System.out.println("||          PERHITUNGAN LUAS SEGITIGA          ||");
212     System.out.println("||");
213     System.out.println("=====");
214     System.out.println();
215     Scanner pp=new Scanner(System.in);
216     Scanner s=new Scanner(System.in);
217     float a,t;
218     Double luass;
219     String pilihan;
220     System.out.print("\tMasukkan nilai alas Segitiga = ");
221     a=pp.nextFloat();
222     System.out.print("\tMasukkan nilai tinggi Segitiga= ");
223     t=pp.nextFloat();
224     luass=0.5*a*t;
225     System.out.println("\tLuas Segitiga = "+luass);
226     System.out.println("Apakah anda ingin mengitung kembali ? (YA/TIDAK) ");
227     pilihan=s.nextLine();
228     if (pilihan.equals("YA")){
229         menu();
230     }
231     if(pilihan.equals("Ya")){
232         menu();
233     }
234     if (pilihan.equals("yA")){
235         menu();
236     }
237     if (pilihan.equals("ya")){
238         menu();
239     }
240 }

```

Gambar 3.20 Program untuk luas segitiga

```

246 static void belahketupat(){
247     System.out.println("=====");
248     System.out.println("||");
249     System.out.println("||          PERHITUNGAN LUAS BELAH KETUPAT          ||");
250     System.out.println("||");
251     System.out.println("=====");
252     System.out.println();
253     Scanner pp=new Scanner(System.in);
254     Scanner bk=new Scanner(System.in);
255     float d1,d2;
256     Double luasbk;
257     String pilihan;
258     System.out.print("\tMasukkan nilai diagonal 1 belahketupat = ");
259     d1=pp.nextFloat();
260     System.out.print("\tMasukkan nilai diagonal 2 belahketupat = ");
261     d2=pp.nextFloat();
262     luasbk=0.5*d1*d2;
263     System.out.println("\tLuas Belah Ketupat = "+luasbk);
264     System.out.println("Apakah anda ingin mengitung kembali ? (YA/TIDAK) ");
265     pilihan=bk.nextLine();
266     if (pilihan.equals("YA")){
267         menu();
268     }
269     if(pilihan.equals("Ya")){
270         menu();
271     }
272     if (pilihan.equals("yA")){
273         menu();
274     }
275     if (pilihan.equals("ya")){
276         menu();
277     }
278 }

```

Gambar 3.21 Program untuk luas belah ketupat

```

284 static void layanglayang(){
285     System.out.println("=====");
286     System.out.println("||");
287     System.out.println("||          PERHITUNGAN LUAS LAYANG-LAYANG          ||");
288     System.out.println("||");
289     System.out.println("=====");
290     System.out.println();
291     Scanner pp=new Scanner(System.in);
292     Scanner ll=new Scanner(System.in);
293     float d1,d2;
294     Double luasll;
295     String pilihan;
296     System.out.print("\tMasukkan nilai diagonal 1 Layang-Layang = ");
297     d1=pp.nextFloat();
298     System.out.print("\tMasukkan nilai diagonal 2 Layang-Layang = ");
299     d2=pp.nextFloat();
300     luasll=0.5*d1*d2;
301     System.out.println("\tLuas Layang-Layang = "+luasll);
302     System.out.println("Apakah anda ingin mengitung kembali ? (YA/TIDAK) ");
303     pilihan=ll.nextLine();
304     if (pilihan.equals("YA")){
305         menu();
306     }
307     if(pilihan.equals("Ya")){
308         menu();
309     }
310     if (pilihan.equals("yA")){
311         menu();
312     }
313 }

```

Gambar 3.22 Program untuk luas layang-layang

```

322 static void trapesium(){
323     System.out.println("=====");
324     System.out.println("||");
325     System.out.println("||          PERHITUNGAN LUAS TRAPESIUM          ||");
326     System.out.println("||");
327     System.out.println("=====");
328     System.out.println();
329     Scanner pp=new Scanner(System.in);
330     Scanner tr=new Scanner(System.in);
331     float luast,a,b,t;
332     String pilihan;
333     System.out.print("\tMasukkan nilai Sisi sejajar atas (a) trapesium = ");
334     a=pp.nextFloat();
335     System.out.print("\tMasukkan nilai Sisi sejajar bawah (b) trapesium = ");
336     b=pp.nextFloat();
337     System.out.print("\tMasukkan nilai Tinggi trapesium = ");
338     t=pp.nextFloat();
339     luast=(a+b/2)*t;
340     System.out.println("\tLuas Belah Ketupat = "+luast);
341     System.out.println("Apakah anda ingin mengitung kembali ? (YA/TIDAK) ");
342     pilihan=tr.nextLine();
343     if (pilihan.equals("YA")){
344         menu();
345     }
346     if(pilihan.equals("Ya")){
347         menu();
348     }
349     if (pilihan.equals("yA")){
350         menu();
351     }
352 }

```

Gambar 3.23 Program untuk luas trapesium

```

361 static void lingkaran(){
362     System.out.println("=====");
363     System.out.println("|          PERHITUNGAN LUAS LINGKARAN          |");
364     System.out.println("|          |");
365     System.out.println("|          |");
366     System.out.println("=====");
367     System.out.println();
368     Scanner pp=new Scanner(System.in);
369     Scanner li=new Scanner(System.in);
370     Double luasl,r;
371     String pilihan;
372     System.out.print("\tMasukkan nilai Jari-jari Lingkaran = ");
373     r=pp.nextDouble();
374     luasl=3.14*r*r;
375     System.out.println("\tLuas Lingkaran = "+luasl);
376     System.out.println("Apakah anda ingin menghitung kembali ? (YA/TIDAK) ");
377     pilihan=li.nextLine();
378     if (pilihan.equals("YA")){
379         menu();
380     }
381     if(pilihan.equals("Ya")){
382         menu();
383     }
384     if (pilihan.equals("yA")){
385         menu();
386     }
387     if (pilihan.equals("ya")){
388         menu();
389     }
390     else{
391         System.out.println();
392         System.out.println("=====");
393         System.out.println("|| TERIMAKASIH SUDAH MENGGUNAKAN PROGRAM MENGHITUNG LUAS BANGUN DATAR ||");
394         System.out.println("||          SEMOGA HARIMU MENYENANGKAN          ||");
395         System.out.println("=====");
396     }
397 }

```

Gambar 3.24 Program untuk luas lingkaran

```

395 static void semuaBangundatar() {
396     System.out.println("=====");
397     System.out.println("|          PERHITUNGAN LUAS SEMUA BANGUN DATAR          |");
398     System.out.println("|          |");
399     System.out.println("|          |");
400     System.out.println("=====");
401     System.out.println();
402
403     //persegi
404     System.out.println("=====");
405     System.out.println("|          PERHITUNGAN LUAS PERSEGI          |");
406     System.out.println("|          |");
407     System.out.println("|          |");
408     System.out.println("=====");
409     System.out.println();
410     Scanner persegi=new Scanner(System.in);
411     Float sisi,luas;
412     System.out.print("\tMasukkan Sisi = ");
413     sisi=persegi.nextFloat();
414     luas=sisi*sisi;
415     System.out.println("\tLuas Persegi = "+luas);
416     System.out.println();
417
418     //persegi panjang
419     System.out.println("=====");
420     System.out.println("|          PERHITUNGAN LUAS PERSEGI PANJANG          |");
421     System.out.println("|          |");
422     System.out.println("|          |");
423     System.out.println("=====");
424     System.out.println();
425     Scanner pp=new Scanner(System.in);
426     Float panjang,lebar,luaspp;

```

Gambar 3.25 Program untuk luas semua bangun datar

#### Pembahasan:

Gambar-gambar di atas merupakan tampilan setiap menu bangun datar. Yang didalamnya terdapat variable scanner, deklarasi variabel masing-masing fungsi, lalu ada juga fungsi percabangan untuk kembali menghitung atau keluar dari program.

Hasil Outputnya :

```
Silahkan Pilih Menu : 1

|| Berikut Pilihan Bangun Datar ||
=====
|| 1 . luas PERSEGI           ||
|| 2 . luas PERSEGI PANJANG  ||
|| 3 . luas JAJARGENJANG     ||
|| 4 . luas SEGITIGA         ||
|| 5 . luas BELAH KETUPAT    ||
|| 6 . luas LAYANG-LAYANG    ||
|| 7 . luas TRAPESIUM        ||
|| 8 . luas LINGKARAN        ||
|| 9 . luas semua bangun datar ||
|| 10. keluar dari program   ||
=====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||1
=====
||                                     ||
||      PERHITUNGAN LUAS PERSEGI      ||
||                                     ||
=====

Masukkan Sisi = 4
Luas Persegi = 16.0
Apakah anda ingin mengitung kembali ? (YA/TIDAK)
```

Gambar 3.26 Hasil Compile Program untuk luas persegi

```
|| Berikut Pilihan Bangun Datar ||
=====
|| 1 . luas PERSEGI           ||
|| 2 . luas PERSEGI PANJANG  ||
|| 3 . luas JAJARGENJANG     ||
|| 4 . luas SEGITIGA         ||
|| 5 . luas BELAH KETUPAT    ||
|| 6 . luas LAYANG-LAYANG    ||
|| 7 . luas TRAPESIUM        ||
|| 8 . luas LINGKARAN        ||
|| 9 . luas semua bangun datar ||
|| 10. keluar dari program   ||
=====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||2
=====
||                                     ||
||      PERHITUNGAN LUAS PERSEGI PANJANG      ||
||                                     ||
=====

Masukkan Nilai Panjang = 5
Masukkan nilai lebar = 2
Luas Persegi Panjang = 10.0
Apakah anda ingin mengitung kembali ? (YA/TIDAK)
```

Gambar 3.27 Hasil Compile Program untuk luas persegi panjang

```

=====
|| Berikut Pilihan Bangun Datar ||
=====
|| 1 . luas PERSEGI           ||
|| 2 . luas PERSEGI PANJANG  ||
|| 3 . luas JAJARGENJANG     ||
|| 4 . luas SEGITIGA         ||
|| 5 . luas BELAH KETUPAT    ||
|| 6 . luas LAYANG-LAYANG    ||
|| 7 . luas TRAPESIUM        ||
|| 8 . luas LINGKARAN        ||
|| 9 . luas semua bangun datar ||
|| 10. keluar dari program   ||
=====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||3
=====
||                               ||
||      PERHITUNGAN LUAS JAJARGENJANG      ||
||                               ||
=====

Masukkan nilai alas Jajargenjang = 21
Masukkan nilai tinggi Jajargenjang = 2
Luas Jajargenjang = 42.0
Apakah anda ingin mengitung kembali ? (YA/TIDAK)

```

Gambar 3.28 Hasil Compile Program untuk luas jajargenjang

```

=====
|| Berikut Pilihan Bangun Datar ||
=====
|| 1 . luas PERSEGI           ||
|| 2 . luas PERSEGI PANJANG  ||
|| 3 . luas JAJARGENJANG     ||
|| 4 . luas SEGITIGA         ||
|| 5 . luas BELAH KETUPAT    ||
|| 6 . luas LAYANG-LAYANG    ||
|| 7 . luas TRAPESIUM        ||
|| 8 . luas LINGKARAN        ||
|| 9 . luas semua bangun datar ||
|| 10. keluar dari program   ||
=====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||4
=====
||                               ||
||      PERHITUNGAN LUAS SEGITIGA      ||
||                               ||
=====

Masukkan nilai alas Segitiga = 32
Masukkan nilai tinggi Segitiga= 12
Luas Segitiga = 192.0
Apakah anda ingin mengitung kembali ? (YA/TIDAK)

```

Gambar 3.29 Hasil Compile Program untuk luas segitiga



```

=====
|| Berikut Pilihan Bangun Datar ||
=====
|| 1 . luas PERSEGI ||
|| 2 . luas PERSEGI PANJANG ||
|| 3 . luas JAJARGENJANG ||
|| 4 . luas SEGITIGA ||
|| 5 . luas BELAH KETUPAT ||
|| 6 . luas LAYANG-LAYANG ||
|| 7 . luas TRAPESIUM ||
|| 8 . luas LINGKARAN ||
|| 9 . luas semua bangun datar ||
|| 10. keluar dari program ||
=====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||5
=====
||                                     ||
||      PERHITUNGAN LUAS BELAH KETUPAT      ||
||                                     ||
=====

Masukkan nilai diagonal 1 belahketupat = 12
Masukkan nilai diagonal 2 belahketupat = 23
Luas Belah Ketupat = 138.0
Apakah anda ingin mengitung kembali ? (YA/TIDAK)

```

Gambar 3.30 Hasil Compile Program untuk luas belahketupat

```

=====
|| Berikut Pilihan Bangun Datar ||
=====
|| 1 . luas PERSEGI ||
|| 2 . luas PERSEGI PANJANG ||
|| 3 . luas JAJARGENJANG ||
|| 4 . luas SEGITIGA ||
|| 5 . luas BELAH KETUPAT ||
|| 6 . luas LAYANG-LAYANG ||
|| 7 . luas TRAPESIUM ||
|| 8 . luas LINGKARAN ||
|| 9 . luas semua bangun datar ||
|| 10. keluar dari program ||
=====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||6
=====
||                                     ||
||      PERHITUNGAN LUAS LAYANG-LAYANG      ||
||                                     ||
=====

Masukkan nilai diagonal 1 Layang-Layang = 12
Masukkan nilai diagonal 2 Layang-Layang = 23
Luas Layang-Layang = 138.0
Apakah anda ingin mengitung kembali ? (YA/TIDAK)

```

Gambar 3.31 Hasil Compile Program untuk luas layang-layang

```

=====
|| Berikut Pilihan Bangun Datar ||
=====
|| 1 . luas PERSEGI ||
|| 2 . luas PERSEGI PANJANG ||
|| 3 . luas JAJARGENJANG ||
|| 4 . luas SEGITIGA ||
|| 5 . luas BELAH KETUPAT ||
|| 6 . luas LAYANG-LAYANG ||
|| 7 . luas TRAPESIUM ||
|| 8 . luas LINGKARAN ||
|| 9 . luas semua bangun datar ||
|| 10. keluar dari program ||
=====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||7
=====
||                                     ||
||          PERHITUNGAN LUAS TRAPESIUM          ||
||                                     ||
=====

Masukkan nilai Sisi sejajar atas (a) trapesium = 12
Masukkan nilai Sisi sejajar bawah (b) trapesium = 21
Masukkan nilai Tinggi trapesium = 12
Luas Belah Ketupat = 270.0
Apakah anda ingin mengitung kembali ? (YA/TIDAK)

```

Gambar 3.32 Hasil Compile Program untuk luas trapesium

```

=====
|| Berikut Pilihan Bangun Datar ||
=====
|| 1 . luas PERSEGI ||
|| 2 . luas PERSEGI PANJANG ||
|| 3 . luas JAJARGENJANG ||
|| 4 . luas SEGITIGA ||
|| 5 . luas BELAH KETUPAT ||
|| 6 . luas LAYANG-LAYANG ||
|| 7 . luas TRAPESIUM ||
|| 8 . luas LINGKARAN ||
|| 9 . luas semua bangun datar ||
|| 10. keluar dari program ||
=====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||8
=====
||                                     ||
||          PERHITUNGAN LUAS LINGKARAN          ||
||                                     ||
=====

Masukkan nilai Jari-jari Lingkaran = 12
Luas Lingkaran = 452.15999999999997
Apakah anda ingin mengitung kembali ? (YA/TIDAK)

```

Gambar 3.33 Hasil Compile Program untuk luas Lingkaran

```

=====
|| Berikut Pilihan Bangun Datar ||
=====
|| 1 . luas PERSEGI ||
|| 2 . luas PERSEGI PANJANG ||
|| 3 . luas JAJARGENJANG ||
|| 4 . luas SEGITIGA ||
|| 5 . luas BELAH KETUPAT ||
|| 6 . luas LAYANG-LAYANG ||
|| 7 . luas TRAPESIUM ||
|| 8 . luas LINGKARAN ||
|| 9 . luas semua bangun datar ||
|| 10. keluar dari program ||
=====

|| Silahkan pilih menu nomor berapa yang ingin di jalankan : ||9

=====
||
|| PERHITUNGAN LUAS SEMUA BANGUN DATAR ||
||
=====

=====
||
|| PERHITUNGAN LUAS PERSEGI ||
||
=====

Masukkan Sisi = 5
Luas Persegi = 25.0

=====
||
|| PERHITUNGAN LUAS PERSEGI PANJANG ||
||
=====

Masukkan Nilai Panjang = 12
Masukkan nilai lebar = 12
Luas Persegi Panjang = 144.0

```

Gambar 3.34 Hasil Compile Program untuk luas semua bangun datar

##### 5. Tampilan untuk fungsi perulangan for dan bubble sorting.

```

549     double luas[]={luasp,luaspp,luasjgg,luass,luasbk,luasll,luast,luasl}; //array
550     double tpenampungan=0;
551
552
553     for(int i=0; i<(luas.length); i++){
554         for(int j=0; j<(luas.length-1); j++){
555             if(luas[j]>luas[j+1]){
556                 tpenampungan=luas[j];
557                 luas[j]=luas[j+1];
558                 luas[j+1]=tpenampungan;
559             }
560             System.out.print("Proses pengurutan hasil dari bangun datar yang ");
561             System.out.print(j+1+" = ");
562             for(int k=0; k<luas.length;k++){
563                 System.out.print(luas[k]+" ");
564             }
565             System.out.println();
566         }
567     }
568
569     System.out.println();
570     System.out.print("Hasil Nilai setelah di urutkan = {");
571     for(int j=0; j<luas.length; j++){
572         System.out.print(luas[j]+" , ");
573     }
574     System.out.print("}");
575     System.out.println();

```

Gambar 3.35 Tampilan Program untuk perulangan dan sorting

Pembahasan :

Gambar di atas merupakan tampilan program untuk fungsi perulangan yang akan di gabungkan dengan fungsi bubble sorting. Di line 549-550 merupakan deklarasi variable dengan tipe data Double. di line 553 merupakan for yang pertama akan di eksekusi. Dan di line 554 perulangan for yang akan di eksekusi setelah kondisi pada for yang pertama masih true. Di line 555 menggunakan fungsi if untuk memindahkan nilai yang lebih besar ke urutan yang lebih kecil. Contoh 2,1 nah 1 akan di pindahkan ke posisi yang di tempati oleh nilai 2.

Hasil Outputnya :

```
Proses pengurutan hasil dari bangun datar yang 1 = 25.0 144.0 42.0 18.0 24.0 11.0 113.0 28.259999999999998
Proses pengurutan hasil dari bangun datar yang 2 = 25.0 42.0 144.0 18.0 24.0 11.0 113.0 28.259999999999998
Proses pengurutan hasil dari bangun datar yang 3 = 25.0 42.0 18.0 144.0 24.0 11.0 113.0 28.259999999999998
Proses pengurutan hasil dari bangun datar yang 4 = 25.0 42.0 18.0 24.0 144.0 11.0 113.0 28.259999999999998
Proses pengurutan hasil dari bangun datar yang 5 = 25.0 42.0 18.0 24.0 11.0 144.0 113.0 28.259999999999998
Proses pengurutan hasil dari bangun datar yang 6 = 25.0 42.0 18.0 24.0 11.0 113.0 144.0 28.259999999999998
Proses pengurutan hasil dari bangun datar yang 7 = 25.0 42.0 18.0 24.0 11.0 113.0 28.259999999999998 144.0

Hasil Nilai setelah di urutkan = {25.0 , 42.0 , 18.0 , 24.0 , 11.0 , 113.0 , 28.259999999999998 , 144.0 , }
Apakah anda ingin mengitung kembali ? (YA/TIDAK)
```

Gambar 3.36 Hasil Compile program fungsi perulangan dan sorting

## **BAB IV**

### **PENUTUP**

#### **A. Kesimpulan**

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*".

Bahasa Java memiliki kelebihan, yaitu Multiplatform, bersifat OOP (Object Oriented Programming-Pemrogram Berorientasi Objek), Perpustakaan Kelas Yang Lengkap, Bergaya C++,Pengumpulan sampah otomatis.

Bahasa java juga memiliki kelemahan,yaitu Tulis sekali, jalankan di mana saja - Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain, Mudah didekompilasi, Penggunaan memori yang banyak.

#### **B. Saran**

Sebaiknya jika ingin menggunakan bahasa java atau membuat program dengan bahasa java. Kita harus sangat teliti karena java merupakan bahasa yang bersifat case sensitive yaitu berpengaruh huruf besar dan kecil dan tanda titik koma.

## DAFTAR PUSTAKA

- Andre. (2010, mai 24). *Tutorial Belajar Java Part 1 : Pegertian Bahasa Pemrograman Java*. Retrieved from Duniaikom: <https://www.duniaikom.com/tutorial-belajar-java-pengertian-bahasa-pemrograman-java/#:~:text=Terjemahan%20bebasnya%2C%20Java%20adalah%20%E2%80%9Cbahasa,tidak%20bergantung%20pada%20sistem%20tertentu%E2%80%9C>.
- Dimas, S. (2020, maret 21). *Contoh Perulangan For, While, Do While pada Java*. Retrieved from Kelas\_programmer: <https://kelasprogrammer.com/perulangan-for-while-do-while-pada-java/>
- Fakmaludin, I. S. (2018). *PEMOGRAMAN C UNTUK PENYELESAIAN KASUS*. Yogyakarta: Penerbit Periuk.
- unknown. (2017, januari). *Implementasi Algoritma Sorting pada Pemograman Java*. Retrieved from JAVA COMMUNITY: <https://www.community-java.com/2017/01/implementasi-algoritma-sorting.html>
- unknown. (2021, maret 11). *Java*. Retrieved from Wikipedia: <https://id.wikipedia.org/wiki/Java>