

# **PEMBELAJARAN PEMROGRAMAN BAHASA JAVA DAN ARTI KEYWORD**

Warno

Program studi Teknik Informatika, Fakultas Teknik, MIPA Universitas Indraprasta PGRI  
Jl. Nangka No.58C Tanjung Barat (TB Simatupang), Jagakarsa, Jakarta Selatan  
warnooke@gmail.com

## **Abstrak**

Pengertian Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan.

Kata kunci : Pemrograman, bahasa Java, *keyword*

## **Pendahuluan**

Pesatnya teknologi, terutama teknologi komputer sudah tak bisa dipungkiri lagi, bagi yang mengikuti perkembangannya, ia tidak akan dipandang sebelah mata. Sebaliknya, bagi yang tidak mengikuti perkembangannya, bersiaplah untuk mundur secara suka rela dari panggung kompetisi. Ibarat wabah, teknologi komputer sudah menyusupi hampir semua bidang kehidupan manusia. Dari pemerintah pusat sampai tingkat pemerintah desa, perusahaan-perusahaan, supermarket, minimarket, perguruan tinggi, bahkan SD hampir semuanya mengenal komputer.

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer stand alone ataupun pada lingkungan jaringan. Java 2 adalah generasi kedua dari java platform. Kata berdiri di atas sebuah mesin interpreter yang diberi nama Java Virtual Machine (JVM). JVM inilah yang akan membaca bytecode dalam file .class dari suatu program sebagai representasi langsung dari program yang berisi bahasa mesin. Oleh karena itu, bahasa java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

## **Pemrograman Java**

Bahasa Java bersifat case sensitive, sehingga Anda harus memperhatikan penggunaan huruf besar dan kecil. Selain itu penulisan source code program tidak harus memperhatikan bentuk tertentu, sehingga Anda bisa saja menuliskan semua baris source code tersebut dalam satu baris asal Anda tidak lupa membubuhkan tanda titik koma (;), atau menuliskan tiap kata dalam satu baris tersendiri. Namun dianjurkan Anda mengikuti layout seperti pada contoh agar program Anda mudah dibaca dan dimengerti. maka beberapa Kelemahan dan Keuntungan dalam menyusun dalam struktur pemrograman yang perlu di ketahui di antaranya adalah :

## **Kelemahan Java**

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

1. 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/*dirreverse-engineer*.

2. 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 berkutut dengan mesin komputer berumur lebih dari 4 tahun.
3. Implementasi J2ME tidak global. Misalnya, J2ME untuk Motorola dengan J2ME untuk Sony Ericson tidak sama. Berbeda lagi J2ME untuk Nokia. Setiap produk selalu mempunyai modul tersendiri yang dinilai aneh penerapannya dan harus di-compile dengan modul yang berbeda-beda

a. **Keunggulan Java**

Bahasa pemrograman lain yang telah ada sebelum Java lahir sudah merupakan bahasa yang baik dan mudah dipelajari oleh programmer profesional. Akan tetapi para programmer ini menginginkan sesuatu yang baru yang memiliki banyak hal yang menyelesaikan masalah mereka. Utamanya adalah keamanan kode mereka. Hal ini melahirkan pikiran yang revolusioner untuk menemukan bahasa pemrograman lain yang disebut Java. Tidak hanya keamanan tapi juga beberapa hal yang sering disebut sebagai Java-Buzzwords. Kata-kata ini menjelaskan berbagai fitur tambahan dan beberapa hal yang membuat Java demikian sukses dan diterima oleh dunia perangkat lunak. Berikut ini adalah keunggulan-keunggulan dari program java :

**1. Platform Independent**

Salah satu keunggulan Java adalah sifatnya yang **'platform independence'**, artinya Java – baik *source program* maupun hasil kompilasinya – sama sekali tidak bergantung kepada sistem operasi dan platform yang digunakan. *Source code* sebuah aplikasi dengan bahasa Java yang ditulis di atas sistem Windows NT misalnya, dengan gampang dapat dipindahkan ke sistem operasi UNIX tanpa harus mengedit satu baris kode-pun. Ini tentunya merupakan satu nilai tambah tersendiri. Bandingkan dengan bahasa C/C++ misalnya, jika kita bekerja pada UNIX FreeBSD dan ingin memindahkannya pada HP UNIX, kita terkadang harus juga mengedit *source code*-nya sehingga sesuai dengan HP UNIX, walaupun keduanya masih berada dalam keluarga UNIX.

**2. Sederhana dan Berorientasi Objek**

Java lahir dari suatu pemikiran mendalam akan bahasa pemrograman yang ada pada saat itu, seperti C dan C++. Hal ini akan memudahkan programmer profesional untuk dapat mengerti lebih jelas tentang Java, fungsionalitas, dan lain sebagainya apabila ia memiliki pengetahuan dasar tentang C++ dan konsep pemrograman berorientasi objek. Tujuannya agar konsep dasar dari teknologi Java dapat dimengerti dengan mudah, dan programmer dapat segera menghasilkan sesuatu sedini mungkin. Tidak hanya ini, penemu Java memastikan bahwa Java juga bermula dari bahasa pemrograman dasar yang sudah ada pada saat itu. Kemudian mereka membuang berbagai fitur yang rumit dan membingungkan.

**3. Automatic garbage collection**

Automatic garbage collection atau 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. Dan fitur ini dapat membersihkan objek yang tidak terpakai dari memori.

**4. Menghilangkan pewarisan berganda yang terdapat pada C++**

Walaupun kelihatannya lebih sebagai suatu kekurangan, namun banyak para ahli yang mengakui bahasa konsep pewarisan berganda lebih banyak mengakibatkan kerugian dari pada keuntungan. *Java* telah didesain sedemikian rupa sehingga Anda tidak akan memerlukan tehnik ini dalam pembuatan *program* apa pun.

## 5. Mengurangi *pointer* aritmetik

Pengaksesan lokasi memori secara langsung dengan menggunakan *pointer* memungkinkan *program* untuk melakukan suatu tindakan yang tidak seharusnya atau tidak boleh dilakukan. Untuk mengurangi dan menghilangkan kemungkinan kesalahan seperti ini, penggunaan *pointer* pada *Java* telah dibatasi dengan menggunakan *reference*.

## 6. Library 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.

## 7. Karena OOP jadi sangat cocok bila digunakan untuk membangun program yang besar.

OOP (*Object Oriented Programming* – Pemrogram Berorientasi Objek) yang artinya semua aspek yang terdapat di *Java* adalah Objek. *Java* merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut *Object*.

Hal ini sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis *Java* secara cepat, tepat, mudah dan terorganisir. Kelebihan ini menjadikan *Java* sebagai salah satu bahasa pemrograman termudah, bahkan untuk fungsi fungsi yang advance seperti komunikasi antara komputer sekalipun.

## 8. 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.

## Hasil dan Pembahasan

### Variabel Java

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.

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.

### Identifier

Identifier adalah kumpulan karakter yang dapat digunakan untuk menamai variabel, method, lass, 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 :

- a. Huruf / abjad
- b. Karakter mata uang
- c. Underscore ( \_ )
- d. Identifier dapat terdiri dari :
- e. Huruf / abjad
- f. Angka
- g. 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 :

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*.

### **Deklarasi Variabel**

Sintaks dasar :

**[tipe data] [nama variabel]**

Menuliskan tipe data dari variabel, contoh :

**int bilangan;**

**char karakter;**

**float bildesimal;**

**boolean status;**

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

**bilangan = 20;**

**karakter = 'k';**

**bildesimal = 22.2f;**

**status = true;**

Dapat juga mendeklarasikan dan memberikan nilai dalam satu baris.

**int bilangan = 20;**

**char karakter = 'k';**

**float bildesimal = 22.2f;**

**boolean status = true;**

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

**final int konstantainteger = 10;**

**final float pajak = 15.5;**

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;**

### **Tipe Data Java**

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

## 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 :  
Daftar deskripsi Bilangan Bulat

**Tabel 1**  
**Diskripsi Bilangan bulat**

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

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 decimal atau perhitungan yang lebih detail dibanding integer. Ada dua macam floating point, yaitu :

**Tabel 2**  
**Diskripsi 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 dimemori
<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 3**  
**Char Khusus dalam bahasa Java**

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.

#### 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.

#### Operator Java

Operator Java merupakan karakter khusus yang berupa simbol atau tanda yang memerintahkan compiler untuk melakukan berbagai operasi terhadap sejumlah operand. Perintah operasi dispesifikasikan oleh operator, dimana operand-nya adalah variabel, pernyataan, atau besaran literal.

Operator yang dibahas pada bagian ini adalah operator aritmatika, increment dan decrement, assignment (penugasan), relasi, logical, dan bitwise.

##### a. Operator Aritmatika

Sama halnya dengan semua bahasa pemrograman, Java menyediakan operator-operator aritmatika untuk manipulasi variabel data numerik. Operator-operator tersebut antara lain :  
Daftar operator dan penggunaan serta diskripsi pada bahasa java

**Tabel 4**  
**Daftar operator dan fungsinya**

Operator	Penggunaan	Deskripsi
+	Op1 + Op2	Menambahkan Op1 dengan Op2
-	Op1 - Op2	Mengurangkan Op1 dengan Op2
*	Op1 * Op2	Mengalikan Op1 dengan Op2
/	Op1 / Op2	Membagi Op1 dengan Op2
%	Op1 % Op2	Menghasilkan sisa hasil bagi Op1 dengan Op2

##### b. Operator *Increment* dan *Decrement*

Operator *Increment* dan *Decrement* digunakan untuk menaikkan atau menurunkan suatu nilai integer (bilangan bulat) sebanyak satu satuan, dan hanya dapat digunakan pada variabel. Ada dua versi operator increment maupun decrement, yaitu prefix dan postfix. Prefix berarti operator digunakan sebelum variabel atau ekspresi, dan postfix berarti operator digunakan sesudahnya. Penjelasan selengkapnya yaitu:

**Tabel 5**  
**Operator Increment dan Decrement**

Operator	Penggunaan	Deskripsi
++	Op++	Op dinaikkan nilainya 1 setelah dilakukan operasi pada Op
	++Op	Op dinaikkan nilainya 1 sebelum dilakukan operasi pada Op
--	Op--	Op diturunkan nilainya 1 sebelum dilakukan operasi pada Op
	--Op	Op diturunkan nilainya 1 sebelum dilakukan operasi pada Op

c. **Operator Assignment** ( Penugasan )

Operator assignment dalam Java digunakan untuk memberikan sebuah nilai ke sebuah variabel. Operator assignment hanya berupa '=', namun selain itu dalam Java dikenal beberapa shortcut assignment operator yang penting, yang digambarkan dalam tabel berikut :  
Daftar operator Assignment ( penugasan)

**Tabel 6**  
**Operator Assignment (penugasan)**

Operator	Penggunaan	Ekuivalen Dengan
+=	Op1 += Op2	Op1 = Op1 + Op2
-=	Op1 -= Op2	Op1 = Op1 - Op2
*=	Op1 *= Op2	Op1 = Op1 * Op2
/=	Op1 /= Op2	Op1 = Op1 / Op2
%=	Op1 %= Op2	Op1 = Op1 % Op2
&=	Op1 &= Op2	Op1 = Op1 & Op2
=	Op1  = Op2	Op1 = Op1   Op2
^=	Op1 ^= Op2	Op1 = Op1 ^ Op2
<<=	Op1 <<= Op2	Op1 = Op1 << Op2
>>=	Op1 >>= Op2	Op1 = Op1 >> Op2
>>>=	Op1 >>>= Op2	Op1 = Op1 >>> Op2

d. **Operator Relasi**

Operator relasi dalam Java digunakan untuk menghasilkan nilai boolean yang sering digunakan untuk mengatur alur jalannya sebuah program.  
Daftar Operator Relasi

**Tabel 7**  
**Operator Relasi**

Operator	Penggunaan	Deskripsi
>	Op1 > Op2	Menghasilkan true jika Op1 lebih besar dari Op2
<	Op1 < Op2	Menghasilkan true jika Op1 lebih kecil dari Op2
>=	Op1 >= Op2	Menghasilkan true jika Op1 lebih besar atau sama dengan Op2
<=	Op1 <= Op2	Menghasilkan true jika Op1 lebih kecil atau sama dengan Op2
==	Op1 == Op2	Menghasilkan true jika Op1 sama dengan Op2
!=	Op1 != Op2	Menghasilkan true jika Op1 tidak sama dengan Op2

e. **Operator Logical**

Operator ini digunakan untuk ekspresi logik yang menghasilkan nilai boolean. Operator-operator yang digunakan adalah AND ( && ), OR ( || ) dan NOT ( ! ).

Daftar Operator logika

**Tabel 8**  
**Operator Logical**

A	B	A  B	A&&B
True	True	True	True
True	False	True	False
False	True	True	False
False	False	False	False

**f. Operator Bitwise**

Operator ini dalam Java digunakan untuk melakukan manipulasi bit.

**Tabel 9**  
**Operator Bitwise**

Operator	Penggunaan	Deskripsi
&	Op1 & Op2	Bitwise AND
	Op1   Op2	Bitwise OR
^	Op1 ^ Op2	Bitwise XOR
~	Op1 ~ Op2	Bitwise Complement
<<	Op1 << Op2	Menggeser bit Op1 ke kiri sejauh Op2
>>	Op1 >> Op2	Menggeser bit Op1 ke kanan sejauh Op2
>>>	Op1 >>> Op2	Geser kanan tanpa mempertahankan sign (dengan nilai 0 sebagai pengisi bit paling kiri)

**g. Bitwise AND**

Bitwise AND akan menghasilkan bit “1”, jika kedua operator bernilai bit “1”. Operasi bitwise AND dapat digambarkan sebagai berikut :

**Tabel 10**  
**Logical bitwise AND**

Op1	Op2	Op1 & Op2
0	0	0
0	1	0
1	0	0
1	1	1

**h. Bitwise OR**

Bitwise OR akan menghasilkan bit “1”, jika salah satu operator bernilai bit “1”. Operasi bitwise OR dapat digambarkan sebagai berikut :

**Tabel 11**  
**Logical bitwise OR**

Op1	Op2	Op1   Op2
0	0	0
0	1	1
1	0	1
1	1	1

**i. Bitwise XOR ( Exclusive OR )**

Bitwise XOR akan menghasilkan bit “1”, jika kedua operator memiliki nilai bit yang berbeda. Operasi bitwise XOR dapat digambarkan sebagai berikut :



Tabel 12

Logical Bitwise XOR (exclusive XOR )		
Op1	Op2	Op1   Op2
0	0	0
0	1	1
1	0	1
1	1	0

**j. Bitwise Complement**

Bitwise Complement akan menghasilkan bit yang berlawanan dengan bit yang dioperasikan. Operasinya dapat digambarkan sebagai berikut :

Tabel 13

Logical Bitwise Complement	
Op	Op
0	1
1	0

**k. Operator Precedence**

Operasi-operasi yang menggunakan operator dapat melibatkan lebih dari 1 operator dan 1 operand. Adapun urutan precedence operator dalam Java, dapat digambarkan dalam tabel berikut :

Tabel 14

Operator precedence pada java.	
Postfix operators	[] . (params) expr++ expr--
Unary operators	++expr --expr +expr -expr ~ !
Creation or cast	New (type) expr
Multiplicative	*/ %
Additive	+-
Shift	<<>>>>
Relational	<><= >= instanceof
Equality	== !=
Bitwise AND	&
Bitwise exclusive OR	^
Bitwise inclusive OR	
Logical AND	&&
Logical OR	
Conditional	?:
Assignment	= += -= *= /= %= &= ^=  = <<= >>= >>>=

**Program membuat calculator dalam percintaan bahasa java.**

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
```

```
import java.awt.event.ActionEvent;
public class TrueLoveCalculator extends JFrame {
    private JPanel contentPane;
    private JTextField txtNama;
    private JTextField txtPasangan;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    TrueLoveCalculator frame = new TrueLoveCalculator();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public TrueLoveCalculator() {
        setTitle("True Love Calculator");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 417, 151);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        JLabel lblNamaAnda = new JLabel("Nama Anda:");
        lblNamaAnda.setBounds(12, 14, 92, 15);
        contentPane.add(lblNamaAnda);

        JLabel lblNamaPasanganAnda = new JLabel("Nama Pasangan Anda:");
        lblNamaPasanganAnda.setBounds(12, 51, 173, 15);
        contentPane.add(lblNamaPasanganAnda);

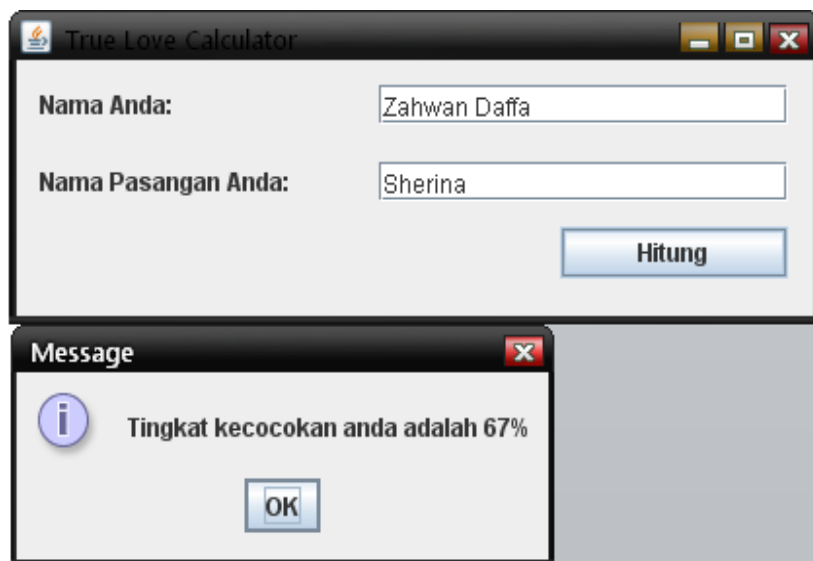
        txtNama = new JTextField();
        txtNama.setBounds(186, 12, 210, 19);
        contentPane.add(txtNama);
        txtNama.setColumns(10);
        txtPasangan = new JTextField();
        txtPasangan.setBounds(186, 49, 210, 19);
        contentPane.add(txtPasangan);
        txtPasangan.setColumns(10);
        JButton btnHitung = new JButton("Hitung");
        btnHitung.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                String nama1, nama2;
```

```
nama1 = txtNama.getText();
nama2 = txtPasangan.getText();
char[] huruf = {'t', 'r', 'u', 'e', 'l', 'o', 'v', 'e'};
int[] frek = new int[huruf.length];
nama1 = nama1.toLowerCase();
nama2 = nama2.toLowerCase();
for (int i = 0; i < nama1.length(); i++){
    for (int j = 0; j < huruf.length; j++){
        if (nama1.charAt(i) == huruf[j]){frek[j]++;
            }
        }
    }
for (int i = 0; i < nama2.length(); i++){
    for (int j = 0; j < huruf.length; j++){
        if (nama2.charAt(i) == huruf[j]){
            frek[j]++;
        }
    }
}

int True =
(frek[0]+(6*frek[1])+(15*frek[2])+(20*frek[3])+(15*frek[4])+(6*frek[5])+(frek[6]))% 10;
int Love =
(frek[1]+(6*frek[2])+(15*frek[3])+(20*frek[4])+(15*frek[5])+(6*frek[6])+(frek[7]))% 10;
    aliyawb
OptionPane.showMessageDialog(null, "Tingkat kecocokan anda adalah " + True + Love + "%");
    }
    }btnHitung.setBounds(279, 80, 117, 25);
    contentPane.add(btnHitung);
    }
}
```

Hasil program java membuat kecocokan dalam menebak prosentasi kecintaan true love calculator diatas adalah :

### Contoh Program Calculator pada manusia



**Gambar 1**  
program calculator love procentase

## **Kesimpulan**

Dari pembahasan di atas, penulis menarik kesimpulan bahwa : Java dikenal sebagai bahasa pemrograman yang bersifat *strongly*, yang berarti diharuskan mendeklarasikan tipe data dari semua variable yang apabila lupa atau salah dalam mengikuti aturan pendeklarasian variable, maka akan terjadi error pada saat proses kompilasi. Tipe data yang terdapat pada bahasa pemrograman Java pada umumnya tidak jauh berbeda dengan bahasa pemrograman lain seperti C, C++, Pascal, Basic, dan lainnya, yakni terdiri dari *integer*, *floating point*, *char*, dan *boolean*. Aturan penulisan variabel pada bahasa pemrograman Java juga tidak jauh berbeda dengan bahasa pemrograman lain seperti C, C++, Pascal, Basic, dan lainnya, yakni harus diawali dengan huruf/abjad, karakter mata uang, dan *underscore* ( `_` ) dan terdiri dari huruf/abjad, angka, dan *underscore*( `_` ), serta tidak boleh menggunakan kata-kata yang dikenal oleh bahasa pemrograman Java (keyword ), seperti *byte*, *case*, *int*, dan lainnya. Sama halnya dengan semua bahasa pemrograman, Java juga menyediakan operator-operator aritmatika untuk memanipulasi data numerik. Selain operator aritmatika, Java juga mempunyai operator-operator lain, seperti operator increment dan decrement, *operator assignment* (penugasan), operator relasi, operator logical, operator bitwise (yang terdiri dari *and*, *or*, *xor*, dan *complement*), dan operator *precedence*.

## **Daftar Pustaka**

Setiyo Cahyono, Februari 2006 penerbit teknik informatika Bandung.

Luwis @2011, penerbit PT elex Media Komputindo, Jakarta.

Deitel, H.M. Java How To Program, sixth Edition. 2004, Prentice Hall.

Exercise 10, Introduction to Computer Science Januari 2003, TU Darmstadt

H.M Deitel, Java How to Program, Sixth Edition. Prentice Hall, August 2004

Rickyanto, Isak, *Dasar Pemrograman Berorientasi Objek*, Yogyakarta : Andy , 2002

Purnama, Rangsang, *Tuntutan Pemrograman Java*, Jakarta : Prestasi Pustaka Publisher.

<http://kodejava.org/http://kodejava.org/>

<http://7seasons.wordpress.com/2011/03/26/tentang-joptionpane-class-pada-java/>

<http://javaprogramming26.blogspot.com/2009/09/swich-case.html>

<http://www.ombar.net/2010/02/pengertian-java.html>