HADIAH 1 PEMOGRAMAN BERORITENTASI OBJEK

1. 50 PESAN ERROR DI JAVA (DENGAN PENJELASAN)
2. 30 FILE HEADER JAVA (DENGAN PENJELASAN->FUNGSI)
3. FORMAT PENULISAN JAVA

Jawaban:

1. Pesan Error di Java:

SYNTAX ERROR:

1. Kesalahan penulisan pada head main method

Seperti sudah anda ketahui, main method adalah method di mana program mulai di eksekusi :

*public static void main (String []args){*

Jika ada sedikit saja kesalahan pada bagian di atas, maka error akan muncul seperti kesalahan kecil berikut ini:



Contoh error:

*Error: Main method not found in class TestApel, please define the main method as:*

*public static void main(String[] args)*

1. Menulis keyword dengan huruf kapital

Menulis keyword Java dengan huruf kapital bisa menyebabkan error.

Misal :

*BOOLEAN a = true;*

Contoh error:

*BOOLEAN cannot be resolved to a type*

1. Mengasumsikan [array 2 dimensi](http://bahasajava.com/mengenal-array-dua-dimensi-java/) langsung diimplemetasikan pada Java

Misalkan :



|  |
| --- |
|  |

Contoh error:

*Syntax error on token “,”, delete this token* *Type mismatch: cannot convert from int[][] to int[]*

1. Terbalik dalam mengisi parameter [konstruktor](http://bahasajava.com/inisialisasi-objek-menggunakan-constructor/)

Terbalik dalam mengisi parameter konstruktor akan dinyatakan sama dengan konstruktor tersebut tidak pernah ada atau tidak didefinisikan.

Misal:



Contoh error:

*The constructor Test(String, int) is undefined*

1. Menggunakan Konstruktor Yang Tidak Ada

Misal:



Contoh error:

*The constructor Test(double) is undefined*

1. Menggunakan Method Yang Tidak Dimiliki Class

Contoh, mencari panjang string:



Contoh error:

*The method size() is undefined for the type String*

1. Terbalik Ketika Mengisi [Parameter Method](http://bahasajava.com/belajar-java-memahami-dasar-method/)

Misal:



Contoh error:

*The method hitungJarak(int, double) in the type Test is not applicable for the arguments (double, int)*

1. Keliru Antara [Operator Prefix Dan Postfix](http://bahasajava.com/belajar-java-dasar-operator-increment-dan-decrement/)

Meskipun tidak menimbulkan *compile error*, namun bisa menyebabkan *logic error*, yaitu program berjalan mulus namun hasilnya salah.  
Contoh 1:

Biasanya menganggap nilai output x = 45 dan y = 46, padahal x dan y adalah 46.

Contoh 2:



Biasanya menganggap output nilai x = 45 dan y = 46, padahal x = 46 dan y = 45

1. Menggunakan Variabel Sebelum Diberi Nilai

Sebelum suatu variabel digunakan maka [variabel tersebut harus diberi nilai terlebih dahulu](http://bahasajava.com/belajar-java-dasar-mendeklarasikan-variabel-untuk-pemula/).

Misal:



Contoh error:

*The local variable x may not have been initialized*

1. Mengakses Variabel Instance Private Melalui Class Lain

Misal:



|  |  |
| --- | --- |
|  |  |

Contoh error:

*The field Apel.berat is not visible*

1. Menggunakan [Operator](http://bahasajava.com/belajar-java-dasar-operator-logika-untuk-pemula/) == Untuk Menguji Apakah Objek Equal Atau Tidak

Misal:

Ketika membandingkan dua objek menggunakan operator ==, maka sebenarnya hanya membandingkan alamat objek pada memori saja. Operator == digunakan untuk membandingkan nilai pada skalar.

Jadi jika objek berada pada alamat yag berbeda pada memori, meskipun memiliki nilai yang sama, maka akan memberikan nilai false jika di uji dengan boolean.

Untuk lebih jelasnya perhatikan perbandingan antara operator == dan method equals pada class String ketika membandingkan 2 objek berikut ini:



Output:



1. Lupa Memberikan Tanda  } Pada Akhir Class Body

Misal:



Contoh error:

*Syntax error, insert “}” to complete ClassBody*

1. Kesalahan Menulis Format Untuk Method Class

Method class harus di tulis dengan format *NamaKelas.NamaMethod(Argumen(s)),*awas jangan keliru dengan method objek.

Misal:



Contoh error:

*The method printWarnaDanBerat() is undefined for the type TestApel*

1. Kesalahan Dalam Penulisan Variabel

Variabel dalam pemrograman Java adalah case sensitive.

Misalnya :

Variabel dengan nama panjang, tidak sama dengan Panjang. Jadi penulisan variabel harus benar-benar diperhatikan.

1. Kesalahan Dalam Penulisan Class Ketika Mendeklarasikan Variabel

Class type dalam Java adalah case sensitive. Ini juga adalah kategori error yang sering terjadi.

Misal :



Contoherror:  
*string can not be resolved to a type*

1. Menggunakan [Method Static](http://bahasajava.com/memahami-variabel-constant-dan-method-static-pada-java/) Seperti Method Objek

Selanjutnya error yang sering terjadi adalah mengirim pesan method static pada objek. Misalnya ketika menggunakan method abs() dari class Math untuk menentukan nilai absolut seperti di bawah ini:



Contoh error:

*Cannot invoke abs() on the primitive type int*

1. Lupa Untuk Mengimport Package

Kebanyakan pemula lupa untuk mengimport package, kecuali untuk package java.lang memang akan diimport secara otomatis.

Contoh:



Seharusnya package diimport dengan : *import java.util.Scanner;*

Contoh error:

*Scanner cannot be resolved to a type*

1. Menulis String Pada Baris Yang Baru

Ini sering terjadi jika string memiliki konten yang panjang dan tidak menggunakan tanda +.

Contoh:



Seharusnya:



Contoh error:

*String literal is not properly closed by a double-quote*

1. Tidak Menggunakan Keyword *Void* Untuk [Method Yang tidak Mengembalikan Nilai](http://bahasajava.com/memanggil-method-yang-tidak-mengembalikan-nilai-method-void/)

Untuk method yang tidak mengembalikan nilai harus menggunakan keyword *void*. Jika tidak, maka akan menghasilkan error, misal:



Contoh error:  
*Return type for the method is missing*

1. Lupa menggunakan keyword return pada value returning method

Untuk [method yang mengembalikan nilai](http://bahasajava.com/memanggil-method-yang-mengembalikan-nilai-value-returning-method/), maka harus menggunakan keyword *return,* bila tidak dapat memunculkan contoh error seperti:



Contoh error:

*This method must return a result of type double*

COMPILE ERROR:

### “… Expected”

This error occurs when something is missing from the code. Often this is created by a missing semicolon or closing parenthesis.



Often this error message does not pinpoint the exact location of the issue. To find it:

* Make sure all opening parenthesis have a corresponding closing parenthesis.
* Look in the line previous to the Java code line indicated. This Java software error doesn’t get noticed by the compiler until further in the code.
* Sometimes a character such as an opening parenthesis shouldn’t be in the Java code in the first place. So the developer didn’t place a closing parenthesis to balance the parentheses.

### “Unclosed String Literal”

The “unclosed string literal” error message is created when the string literal ends without quotation marks, and the message will appear on the same [line as the error](http://www.dreamincode.net/forums/topic/116743-unclosed-string-literal-error/). ([@DreamInCode](https://twitter.com/dreamincode)) A literal is a source code of a value.



Commonly, this happens when:

* The string literal does not end with quote marks. This is easy to correct by closing the string literal with the needed quote mark.
* The string literal extends beyond a line. Long string literals can be broken into multiple literals and concatenated with a plus sign (“+”).
* Quote marks that are part of the string literal are not escaped with a backslash (“\”).

Read a [discussion of the unclosed string literal](https://www.quora.com/What-is-an-unclosed-string-literal) Java software error message. ([@Quora](https://twitter.com/Quora))

### “Illegal Start of an Expression”

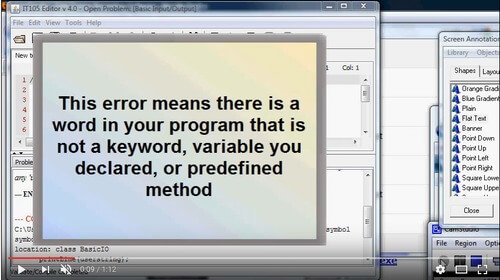
There are numerous reasons why an “illegal start of an expression” error occurs. It ends up being one of the less-helpful error messages. Some developers say it’s caused by bad code.

Usually, expressions are created to produce a new value or assign a value to a variable. The compiler expects to find an expression and cannot find it because the [syntax does not match expectations](https://stackoverflow.com/questions/19581173/java-getting-illegal-start-of-expression-and-expected-errors-for-a-metho). ([@StackOverflow](https://twitter.com/StackOverflow)) It is in these statements that the error can be found.



### “Cannot Find Symbol”

This is a very common issue because all identifiers in Java need to be declared before they are used. When the code is being compiled, the compiler does not understand what the identifier means.



There are many reasons you might receive the “cannot find symbol” message:

* The spelling of the identifier when declared may not be the same as when it is used in the code.
* The variable was never declared.
* The variable is not being used in the same scope it was declared.
* The class was not imported.

### “Public Class XXX Should Be in File”

The “public class XXX should be in file” message occurs when the class XXX and the Java program filename [do not match](https://coderanch.com/t/628555/java/Netbeans-static-variables). The code will only be compiled when the class and Java file are the same. ([@coderanch](https://twitter.com/coderanch)):



To fix this issue:

* Name the class and file the same.
* Make sure the case of both names is consistent.

### “Incompatible Types”

“Incompatible types” is an error in logic that occurs when an assignment statement tries to pair a variable with an expression of types. It often comes when the code tries to place a [text string into an integer](https://stackoverflow.com/questions/18861044/what-is-wrong-with-this-incompatible-type-error) — or vice versa. This is not a Java syntax error.



There really isn’t an easy fix when the compiler gives an “incompatible types” message:

* There are functions that can convert types.
* The developer may need change what the code is expected to do.

### “Invalid Method Declaration; Return Type Required”

This Java software error message means the return type of a method was not explicitly stated in the method signature.



There are a few ways to trigger the “invalid method declaration; return type required” error:

* Forgetting to state the type
* If the method does not return a value then “void” needs to be stated as the type in the method signature.
* Constructor names do not need to state type. But if there is an error in the constructor name, then the compiler will treat the constructor as a method without a stated type.

### “Method <X> in Class <Y> Cannot Be Applied to Given Types”

This Java software error message is one of the more helpful error messages. It explains how the method signature is calling the wrong parameters.



The method called is expecting certain arguments defined in the method’s declaration. Check the method declaration and call carefully to make sure they are compatible.

### “Missing Return Statement”

The “missing return statement” message occurs when a method does not have a return statement. Each method that returns a value (a non-void type) must have a statement that literally returns that value so it can be called outside the method.

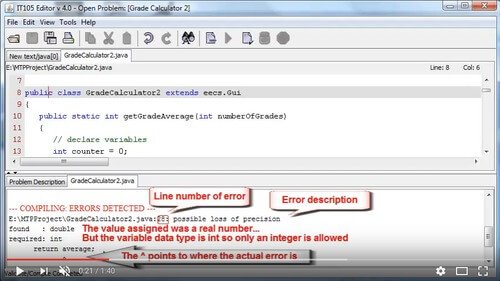


There are a couple reasons why a compiler throws the “missing return statement” message:

* A return statement was simply omitted by mistake.
* The method did not return any value but type void was not declared in the method signature.

### “Possible Loss of Precision”

“Possible loss of precision” occurs when more information is assigned to a variable than it can hold. If this happens, pieces will be thrown out. If this is fine, then the code needs to explicitly declare the variable as a new type.



A “possible loss of precision” error commonly occurs when:

* Trying to assign a real number to a variable with an integer data type.
* Trying to assign a double to a variable with an integer data type.

### “Reached End of File While Parsing”

This error message usually occurs in Java when the program is missing the closing curly brace (“}”). Sometimes it can be quickly fixed by placing it at the end of the code.



The above code results in the following error:



Coding utilities and proper code indenting can make it easier to find these unbalanced braces.

### “Unreachable Statement”

“Unreachable statement” occurs when a statement is written in a place that prevents it from being executed. Usually, this is after a break or return statement.



Often simply moving the return statement will fix the error.

### “Variable <X> Might Not Have Been Initialized”

This occurs when a local variable declared within a method has not been initialized. It can occur when a variable without an initial value is part of an if statement.



### “Operator ... Cannot be Applied to <X>”

This issue occurs when operators are used for types not in their definition.



This often happens when the Java code tries to use a type string in a calculation. To fix it, the string needs to be converted to an integer or float.

### “Inconvertible Types”

The “inconvertible types” error occurs when the Java code tries to perform an illegal conversion.



For example, booleans cannot be converted to an integer.

### “Missing Return Value”

You’ll get the “missing return value” message when the return statement includes an incorrect type. For example, the following code:



Returns the following error:



Usually, there is a return statement that doesn’t return anything.

### “Cannot Return a Value From Method Whose Result Type Is Void”

This Java error occurs when a void method tries to return any value, such as in the following example:



Often this is fixed by changing to method signature to match the type in the return statement. In this case, instances of void can be changed to int:



### “Non-Static Variable ... Cannot Be Referenced From a Static Context”

This error occurs when the compiler tries to [access non-static variables from a static method](https://javarevisited.blogspot.com/2012/02/why-non-static-variable-cannot-be.html) ([@javinpaul](https://twitter.com/javinpaul)):



To fix the “non-static variable ... cannot be referenced from a static context” error, two things can be done:

* The variable can be declared static in the signature.
* The code can create an instance of a non-static object in the static method.

### “Non-Static Method ... Cannot Be Referenced From a Static Context”

This issue occurs when the Java code tries to call a non-static method in a non-static class. For example, the following code:



Would return this error:



To call a non-static method from a static method is to declare an instance of the class calling the non-static method.

### “(array) <X> Not Initialized”

You’ll get the “(array) <X> not initialized” message when an array has been declared but not initialized. Arrays are fixed in length so each array [needs to be initialized](https://stackoverflow.com/questions/5387643/array-initialization-syntax-when-not-in-a-declaration) with the desired length.

The following code is acceptable:



As is:



But not:



## “ArrayIndexOutOfBoundsException”

This is a runtime error message that occurs when the code attempts to access an array index that is not within the values. The [following code](https://stackoverflow.com/questions/5554734/what-causes-a-java-lang-arrayindexoutofboundsexception-and-how-do-i-prevent-it) would trigger this exception:



Here’s [another example](https://www.cs.duke.edu/csed/ap/subset/doc/ap/java/lang/ArrayIndexOutOfBoundsException.html) ([@DukeU](https://twitter.com/DukeU)):



Array indexes start at zero and end at one less than the length of the array. Often it is fixed by using “<” instead of “<=” when defining the limits of the array index.

## “StringIndexOutOfBoundsException”

This is an issue that occurs when the code attempts to access part of the string that is not within the bounds of the string. Usually, this happens when the code tries to create a substring of a string that is not as long as the parameters are set at. Here’s [an example](https://examples.javacodegeeks.com/java-basics/exceptions/java-lang-stringindexoutofboundsexception-how-to-solve-stringindexoutofboundsexception/) ([@javacodegeeks](https://twitter.com/javacodegeeks)):



Like array indexes, string indexes start at zero. When indexing a string, the last character is at one less than the length of the string. The “StringIndexOutOfBoundsException” Java software error message usually means the index is trying to access characters that aren’t there.

## “NullPointerException”

A “NullPointerException” will occur when the program tries to use an object reference that [does not have a value assigned](http://www.geeksforgeeks.org/null-pointer-exception-in-java/) to it ([@geeksforgeeks](https://twitter.com/geeksforgeeks)).



The Java program raises an exception often when:

* A statement references an object with a null value.
* Trying to access a class that is defined but isn’t assigned a reference.

## “NoClassDefFoundError”

The “NoClassDefFoundError” will occur when the interpreter cannot find the file containing a class with the main method. Here’s an example from [DZone](https://dzone.com/articles/java-classnotfoundexception-vs-noclassdeffounderro) ([@DZone](https://twitter.com/DZone)):

If you compile this program:



Two .class files are generated: A.class and B.class. Removing the A.class file and running the B.class file, you’ll get the NoClassDefFoundError:



This can happen if:

* The file is not in the right directory.
* The name of the class must be the same as the name of the file (without the file extension). The names are case sensitive.

## “NoSuchMethodFoundError”

This [error message](http://wiki.cs.und.edu/java_errors#nosuchmethodfounderrormain_could_not_find_or_load_main_class) will occur when the Java software tries to call a method of a class and the method no longer has a definition ([@myUND](https://twitter.com/myUND)):



Often the “NoSuchMethodFoundError” Java software error occurs when there is a typo in the declaration.

## “NoSuchProviderException”

“NoSuchProviderException” occurs when a security provider is requested that is [not available](http://alvinalexander.com/blog/post/java/java-javax-mail-nosuchproviderexception-solved) ([@alvinalexander](https://twitter.com/alvinalexander)):



When trying to find why “NoSuchProviderException” occurs, check:

* The JRE configuration.
* The Java home is set in the configuration.
* Which Java environment is used.
* The security provider entry.

## AccessControlException

AccessControlException indicates that requested access to system resources such as a file system or network is denied, as in this example from [JBossDeveloper](https://developer.jboss.org/thread/274429) ([@jbossdeveloper](https://twitter.com/jbossdeveloper)):



Read this discussion of a [workaround used to get past an “AccessControlException” error.](https://github.com/junit-team/junit4/issues/1213) ([@github](https://twitter.com/github))

## “ArrayStoreException”

An “ArrayStoreException” occurs [when the rules of casting elements in Java arrays](http://mindprod.com/jgloss/array.html#ARRAYSTOREEXCEPTION) are broken. Arrays are very careful about what can go into them. ([@Roedyg](https://twitter.com/Roedyg)) For instance, this example from [JavaScan.com](https://www.javascan.com/306/arraystoreexception-in-java-example) illustrates that this program ([@java\_scan](https://twitter.com/java_scan)):



Results in the following output:



When an array is initialized, the sorts of objects allowed into the array need to be declared. Then each array element needs be of the same type of object.

## “Bad Magic Number”

This Java software error message means something may be wrong with the class definition files on the network. Here’s an example from [The Server Side](http://www.theserverside.com/news/thread.tss?thread_id=33274) ([@TSS\_dotcom](https://twitter.com/TSS_dotcom)):



The “bad magic number” error message could happen when:

* The first four bytes of a class file is not the hexadecimal number CAFEBABE.
* The class file was uploaded as in ASCII mode not binary mode.
* The Java program is run before it is compiled.

## “Broken Pipe”

This [error message](https://www.experts-exchange.com/questions/23307716/Exception-in-thread-main-java-net-SocketException-Broken-pipe.html) refers to the data stream from a file or network socket has stopped working or is closed from the other end ([@ExpertsExchange](https://twitter.com/ExpertsExchange)).



The causes of a broken pipe often include:

* Running out of disk scratch space.
* RAM may be clogged.
* The datastream may be corrupt.
* The process reading the pipe might have been closed.

1. File Header pada Java:
2. Import java.io.\*;

Import java.io.\*;Pengertian IO sendiri merupakan singkatan dari Input Output.  
Input merupakan perintah untuk memasukan data dari peralatan input sebagai contoh keyboar, sedangkan output merupakan perintah atau berfungsi menampilkan data kepada pengguna.

1. Import java.util.Date.\*;

Import java.util.Date.\*;  Merupakan perintah yang digunakan untuk menginputkan data yang berupa Date atau suatu Tanggal dan waktu (timer).  
Fungsi Import java.util.Date.\*; Untuk Memasukan dan Memanggil Inputan yang berupa DATE

1. Import java.awt.\*;

Import java.awt.\*; AWT tersebut singkatan dari Abstract Windowing Toolkit yang merupakan Suatu Perintah Untuk memanggil library yang ada di awt agar dapat digunakan dalam pembuatan program.  
Fungsi Import java.awt digunakan untuk proses inputan dalam sebuah program java yang telah jalankan tersebut.

1. Import javax.swing.event.\*;

Import javax.swing.event.\*; Merupakan perintah dalam Program Java yang berisi kelas-kelas dan interfaces yang memperbolehkan menampilkan ataupun menerima pesan dalam penanganan event untuk komponen grafis yang berada di paket javax.swing.  
Fungsi Import javax.swing.event.\*; digunakan untuk proses deklarasi dari class event yang berisi komponen dari sebuah program yang digunakan untuk mengambil semua file dari package event yang mana package event berada pada package java.awt.

1. Import javax.swing.\*;

Import javax.swing.\*; Merupakan perintah dalam Program Java yang berisi kelas-kelas dan interface (menampilkan ataupun menerima pesan) untuk komponen GUI swing java yang menyediakan dukungan untuk komponen grafis yang portable pada contoh disini biasanya menggunakan sebuah gambar atau membaca dan memanggil grafis pada suatu gambar yang akan di deklarasikan.  
Fungsi Import javax.swing.\*; Mendeklarasikan Sebuah Komponen yang Berupa Grafis dengan perintah yang dilakukan dalam pembuatan Program.

1. Import java.lang.\*;

Import java.lang.\*; Merupakan perintah dalam Program Java yang berisi kelas-kelas dan interfaces yang diperlukan untuk berbagai jenis perintah program java yang dibuat.  
Fungsi Import java.lang.\*; untuk menghubungkan penrintah - perintah dalam program yang yang digunakan untuk kompiler ke semua program java tersebut.

1. Import java.applet.\*;

Import java.applet.\*; Merupakan perintah dalam Program Java yang berisi kelas-kelas Applet.Applet merupakan suatu aplikasi java yang bisa disisipkan kedalam halaman HTML.  
Perintah Method pada Java applet bisa berjalan pada web browser yang telah mendukung java seperti mozilla, IE, dan chrome dan lainnya. Java Applet juga dapat digunakan sebagai browser yang bergunaka untuk memainkan sebuah klip audio.  
Fungsi Import java.applet.\*; Biasanya digunakan untuk pembuatan sebuah Website dengan bahasa pemrograman Java dan biasanya digunakan dalam tampilan Mobile.

1. Import java.text.\*;

Import java.text.\*; Merupakan perintah dalam Program Java berisi kelas-kelas dan interfaces yang memperbolehkan program java untuk memanipulasi angka, tanggal, karakter dan juga string.  
  
Fungsi Import java.text.\*; Memanipulasi Perintah perintah dalam pemrograman Java angka, tanggal, karakter dan juga string.

1. Import java.net.\*;

Import java.net.\*; Merupakan perintah Program Java berisi kelas-kelas  yang memperbolehkan program berkomunikasi melewati jaringan.  
Fungsi Import java.net.\*; Digunakan dalam Membuat suatu Komunikasi Jaringan dalam sebuah program yang dibuat dengan bahasa pemrograman [Java](http://www.java.com/).

1. Import.javax.swing.JOptionPane.\* berfungsi untuk membuat satu kelas terlihat.
2. Import.java.util.\* digunakan untukstruktur data (collections), waktu, scanner, kelas, dll.
3. Import.java.util.regex.\* berfungsi untuk membuat regular expression classes.
4. Import.java.io.BufferedReader.\* digunakan untuk mendapatkan input dari user.
5. Import.java.util.Scanner.\* digunakan untuk mengimpor scanner ke program.
6. Import.java.io.InputStreamReader.\*digunakan untuk menyangga input dan mendukung metode penandaan dan reset. Saat Buffered Input Stream dibuat, array buffer internal dibuat.
7. Import.java.io.IOException.\*dgunakan untuk menyediakan penanganan terhadap error atau kesalahan.
8. Import.java.awt.event.\* : the most common GUI event listeners
9. Import.java.io.console.\* berfungsi untuk mengakses perangkat console berbasis karakter.
10. Import.java.io.FileInputStream.\* berfungsi untuk membaca data daris ebuah file yang berupa urutan byte.
11. Import.java.io.BufferedOutputStream.\* berfungsi untuk mengimplementasikan aliran output yang ada di buffer.
12. Import.java.io.CharArrayReader.\* berfungsi untuk membuat buffer karakter menggunakan array karakter
13. Import.java.io.InputStream.\* berfungsi untuk mewakili aliran input byte
14. Import.java.io.BufferedInputStream.\*berfungsi untuk menyangga input dan mendukung metode penandadan reset itu saja
15. Import.java.io.FilterReader berfungsi untuk membaca aliran karakter yang difilter
16. Import.java.io.Writer berfungsi untuk menulis aliran karakter
17. Import.java.io.Reader berfungsi untuk membaca aliran karakter
18. Import.java.io.FilePermission berfungsi untuk mewakili akses ke file atau direktori
19. Import.java.util.JarEntry berfungsi untuk mewakili entri file JAR
20. Import.java.io.RandomAccessFile berfungsi untuk mengakses file secara acak menggunakan operasi membaca dan menulis
21. Import.java.io.SequenceInputStream berfungsi untuk memungkinkan untukmenggabungkan beberapa input stream
22. Format Penulisan pada Java:

Struktur dalam bahasa pemrograman Java merupakan sebuah format standar yang dikeluarkan oleh developer bagi para programmer. Struktur tersebut berbentuk serpti dibawah ini

package details

class className

{

Data members;

user\_defined method;

public static void main (String args[])

{

Block of Statements

}

}

Penjelasan tiap struktur dalam bahasa pemrograman java adalah sebagai berikut

* Package. Merupakan kumpulan dari class, interace dan sub-packages. Sub package terdiri dari kumpulan class, interface dan sub-sub packages etc.java.lang.\*; Package diimpor secara default dan disebut dengan default package
* Class. Merupakan keyword yang digunakan untuk mengembangkan tipe data user defined. Setiap program java harus dimulai dengan konsep class
* "ClassName" merupakan nama variabel valid dalam java yang dianggap sebagai nama dari setiap class. Setiap nama class dalam pemrograman Java dianggap sebagau tipe data user-defined.
* Data member. Melambangkan instance maupun static yang akan dipilih berdasarkan dari nama class
* User-defined. Melambangkan instance maupun static yang akan digunakan untuk melakukan operasi baik satu kali maupun setiap kali.
* Setiap program Java memulai eksekusi dari main method. Oleh sebab itu, main method disebut juga sebagai program driver.
* Karena sifat main method yang tidak mengembalikan nilai, maka return type harus dihindari.
* Karena sifat main method yang hanya dijalankan sekali sepanjang program, maka main method harus static.
* Karena sifat main method yang harus dapat diakses oleh setiap programmer java, maka access nya harus public.
* Setiap main method dalam java harus memakai array String
* Block of Statement. Melambangkan barisan program yang dapat dieksekusi yang memanggil user-defined method yang memiliki business-logic.
* Penamaan dalam pemrograman Java adalah pada setiap class yang mengandung main method, maka nama file harus sama dengan nama class dengan ditambahkan ekstensi .java