1. 50 pesan error pada Bahasa pemrograman Java dan penjelasannya

Jawab:

1. **“…Expected”**

Kesalahan ini terjadi ketika ada sesuatu yang hilang dari kode program. Seringkali ini terjadi jika tidak ada titik koma atau tanda tutup kurung.

1. **“Unclosed String Literal”**

Pesan error ini muncul apabila *string literal* dibuat tanpa diakhiri dengan tanda kutip, dan pesan akan muncul di baris yang sama dengan kesalahan.

1. **“Illegal Start of an Expression”**

Pesan error ini biasa terjadi dengan berbagai alasan. Oleh karena itu pesan error ini anggap kurang membantu. Beberapa developer mengatakan bahwa pesan error ini disebabkan oleh kode yang buruk. Biasanya *Expression* dibuat untuk menghasilkan nilai baru atau menetapkan nilai ke variable. *Compiler* mengharapkan untuk menemukan *Expression* tetapi tidak dapat menemukannya karena sintaksnya tidak sesuai dengan harapan atau kodenya buruk.

1. **“Cannot Find Symbol”**

Ini adalah masalah yang sangat umum karena semua pengenal di Java harus dideklarasikan sebelum digunakan. Ketika kode sedang dikompilasi, *Compiler* tidak mengerti apa yang dimaksud dengan pengenal.

1. **“Public Class XXX Should Be in File”**

Pesan error ini muncul ketika class XXX dan nama file program Java tidak cocok. Kode hanya akan di kompilasi jika class dan file Java sama.

1. **“Incompatible Types”**

Pesan error ini sering muncul ketika kode mencoba menempatkan teks string menjadi integer atau sebaliknya. Tapi itu bukanlah kesalahan sintaks Java

1. **“Invalid Method Declaration; Return Type Required”**

Pesan error ini berarti pengembalian tipe suatu metode tidak secara eksplisit dinyatakan dalam tanda tangan metode.

1. **“Method <X> in Class <Y> Cannot Be Apllied to Given Types”**

Pesan error ini menjelaskan tanda tangan metode yang memanggil parameter yang salah.

1. **“Missing Return Statement”**

Ini terjadi ketika metode tidak memiliki pernyataan pengembalian. Setiap metode yang mengembalikan nilai (tipe non-void) harus memiliki pernyataan yang secara harfiah mengembalikan nilai itu sehingga bisa dipanggil diluar metode.

1. **“Possible Loss of Precision”**

Ini terjadi ketika lebih banyak informasi yang diberikan ke variable daripada yang apat ditampungnya.

1. **“Reached End of File While Parsing”**

Pesan error ini biasanya terjadi di Java ketika program kehilangan tanda kurung kurawal tutup (“}”).

1. **“Unreachable Statement”**

Ini terjadi ketika sebuah pernyataan ditulis di tempat yang mencegahnya untuk dieksekusi. Biasanya, ini setelah pernyataan break atau return.

1. **“Variable <X> Might Not Have Been Initialized”**

Error ini terjadi ketika variable local yang dideklarasikan dalam metode belum diinisialisasikan. Hal itu bisa terjadi ketika variable tanpa nilai awal adalah bagian dari pernyataan if.

1. **“Operator … Cannot be Applied to <X>”**

Masalah ini terjadi ketika operator digunakan untuk tipe yang tidak memiliki definisi.

1. **“Inconvertible Types”**

Pesan error ini muncul saat kode Java mencoba melakukan konversi illegal.

1. **“Missing Return Value”**

Kita akan mendapatkan pesan seperti ini jika pernyataan return menyertakan type yang salah.

1. **“Cannot Return a Value From Method Whose Result Type Is Void”**

Error Java ini terjadi saat metode void mencoba mengembalikan nilai apapun.

1. **“Non-Static Variable … Cannot Be Referenced From a Static Context”**

Kesalahan ini terjadi ketika *Compiler* mencoba mengakses variable non-static dari metode static.

1. **“Non-Static Method … Cannot Be Referenced From a Static Context”**

Masalah ini terjadi saat kode Java mencoba memanggil metode non-static di kelas non-static.

1. **“(array) <X> Not Initialized”**

Kita akan mendapatkan pesan error seperti ini ketika array telah dideklarasikan tetapi tidak diinisialisasikan.

1. **“Array Index Out Of Bounds Exception”**

Ini adalah pesan error runtime yang terjadi saat kode mencoba mengakses indeks array yang tidak ada di dalam nilai.

1. **“String Index Out Of Bounds Exception”**

Ini adalah masalah yang terjadi saat kode mencoba mengakses bagian dari string yang tidak berada dalam batas-batas string. Biasanya, ini terjadi ketika kode mencoba membuat substring dari string yang parameternya tidak disetel.

1. **“Null Pointer Exception”**

Pesan error ini akan muncul saat program mencoba menggunakan referensi objek yang tidak memiliki nilai yang ditetapkan untuk itu.

1. **“No Class Def Founder Error”**

Pesan error ini akan muncul ketika interpreter tidak dapat menemukan file yang berisi class dan main method.

1. **“No Such Method Found Error”**

Pesan eeror ini akan terjadi ketika Java mencoba memanggil class method dan method tersebut tidak lagi memiliki definisi.

1. **“No Such Provider Found Error”**

Pesan error ini akan muncul saat penyedia keamanan yang diminta tidak tersedia.

1. **Access Control Exception**

*Access Control Exception* menunjukkan bahwa akses yang diminta ke sumber daya system seperti system file atau jaringan di tolak.

1. **“Array Store Exception”**

Error ini terjadi saat aturan elemen casting dalam array Java dilanggar. Array sangat berhati-hati tentang apa yang bisa masuk ke dalamnya.

1. **“Bad Magic Number”**

Pesan error ini berarti ada yang salah dengan file define kelas di jaringan.

1. **“Broken Pipe”**

Pesan error ini mengacu pada aliran data dari file atau soket jaringan telah berhenti atau ditutup dari ujung lain.

1. **“Could Not Create Java Virtual Machine”**

Pesan error ini biasanya terjadi ketika kode mencoba memanggil Java dengan argument yang salah.

1. **“Class File Contains Wrong Class”**

Error ini terjadi saat kode Java mencoba menemukan file class di direktori yang salah.

1. **“Class Cast Exception”**

Pesan error ini menunjukkan kode Java mencoba mentransmisikan objek ke kelas yang salah.

1. **“Class Format Error”**

Pesan error ini menunjukkan kesalahan penautan dan terjadi ketika file class tidak dapat dibaca atau ditafsirkan sebagai file class.

1. **“Class Not Found Exception”**

Error ini hanya terjadi pada waktu proses. Artinya class yang ada disana selama kompilasi hilang pada waktu proses. Ini adalah kesalahan tautan.

1. **“Exception In Initializer Error”**

Error Java ini akan terjadi ketika terjadi kesalahan dengan inisialisasi statis. Ketika kode Java nanti menggunakan kelas, *Exception In Initializer Error* akan terjadi.

1. **“Illegal Block Size Exception”**

Error ini akan terjadi selama deskripsi apabila panjang pesan bukan kelipatan 8 byte.

1. **“Bad Padding Exception”**

Error ini akan terjadi selama deskripsi saat padding digunakan untuk membuat pesan daripada yang dapat diukur dengan kelipatan 8 byte.

1. **“Incompatible Class Change Error”**

*Incompatible Class Change Error* adalah bentuk Linkage Error yang bisa terjadi ketika base class berubah setelah kompilasi dari child class.

1. **“File Not Found Exception”**

Pesan error ini muncul ketika file dengan nama jalur yang ditentukan tidak ada.

1. **“EOF Exception”**

Pesan error ini muncul ketika akhir file telah dicapai secara tidak terduga selama input.

1. **“Unsupported Encoding Exception”**

Pesan error ini muncul saat pengkodean karakter tidak didukung.

1. **“Socket Exception”**

Error ini menunjukkan ada kesalahan saat membuat atau mengakses soket.

1. **“SLL Exception”**

Pesan error ini terjadi ketika ada kegagalan dalam operasi terkait SLL.

1. **“Missing Resource Exception”**

Error ini terjadi saat sumber daya tidak ada. Jika sumber daya berada di jalur class yang benar, ini biasanya karena file property tidak dikonfigurasi dengan benar.

1. **“No Initial Context Exception”**

Error ini terjadi ketika aplikasi Java ingin melakukan operasi penamaan tetapi tidak dapat membuat koneksi.

1. **“No Such Element Exception”**

Error ini terjadi saat iterasi (seperti loop “for”) mencoba mengakses elemen berikutnya tetapi tidak ada.

1. **“No Such Field Error”**

Pesan error ini muncul saat aplikasi mencoba mengakses bidang di objek, tetapi bidang yang ditentukan tidak lagi ada di object.

1. **“Number Format Exception”**

Pesan error ini terjadi saat aplikasi mencoba mengonversi string menjadi tipe numerik, tetapi nomor tersebut bukan string digit yang valid.

1. **“Time Out Exception”**

Pesan error ini terjadiketika waktu operasi pemblokiran habis.

1. 30 file header java (import java.io,?) Dan fungsinya untuk apa
   1. Import java.io.\*; untuk memasukan dan mengeluarkan data.
   2. Import java.util.Date.\*; Untuk Memasukan dan Memanggil Inputan yang berupa DATE
   3. Import java.awt digunakan untuk proses inputan dalam sebuah program java yang telah jalankan tersebut.
   4. 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.
   5. Import javax.swing.\*; Mendeklarasikan Sebuah Komponen yang Berupa Grafis dengan perintah yang dilakukan dalam pembuatan Program.
   6. Import java.lang.\*; untuk menghubungkan penrintah - perintah dalam program yang yang digunakan untuk kompiler ke semua program java tersebut.
   7. Import java.applet.\*; Biasanya digunakan untuk pembuatan sebuah Website dengan bahasa pemrograman Java dan biasanya digunakan dalam tampilan Mobile.
   8. Import java.text.\*; Memanipulasi Perintah perintah dalam pemrograman Java angka, tanggal, karakter dan juga string.
   9. Import java.net.\*; Digunakan dalam Membuat suatu Komunikasi Jaringan dalam sebuah program yang dibuat dengan bahasa pemrograman Java.
   10. Import javax.swing.JOptionPane; Membuat satu kelas terlihat
   11. Import java.awt.event.\*; The most common GUI event listeners.
   12. Import java.util.\*; Data structures (Collections), time, Scanner, etc classes.
   13. Import java.util.regex.\*; Regular expression classes.
   14. Import java.io.BufferedReader; Mendapatkan input dari user.
   15. Import java.util.Scanner; Mengimport scanner ke program.
   16. Import java. .io.InputStreamReader; Digunakan untuk sebuah variabel dalam memasukkan inputan ke dalam sebuah program.
   17. Import java.io.IOException; untuk menyediakan penanganan terhadap error atau kesalahan.
   18. Import java.io.BufferedInputStream; untuk menyangga input dan mendukung metode penanda dan reset itu aja.
   19. Import java.io.console; untuk mengakses perangkat konsol berbasis karakter.
   20. Import java.io.FileInputStream; untuk membaca data dari sebuah file yang berupa urutan byte.
   21. Import java.io.BufferedOutputStream; mengimplementasikan aliran keluaran yang di buffer.
   22. Import java.io.CharArrayReader; membuat buffer karakter menggunakan array karakter.
   23. Import java.io.InputStream; mewakili aliran input byte.
   24. Import java.io.filterreader; membaca aliran karakter yang difilter.
   25. Import java.io.Writer; menulis aliran karakter.
   26. Import java.io.Reader; membaca aliran karakter.
   27. Import java.io.FilePermission; mewakili akses ke file atau direktori.
   28. Import java.util.JarEntry; mewakili entri file JAR.
   29. Import java.io.RandomAccessFile; mengakses file secara acak menggunakan operasi membaca dan menulis.
   30. Import java.io.SequenceInputStream; memungkinkan untuk menggabungkan beberapa inputStream.
2. Format penulisan bahasa pemrograman java
3. Harus disimpan berekstensi \*.java
4. Nama file harus sama dengan nama class public, misalnya nama filenya myHello.java, maka  nama  class public juga harus myHello

Contoh:

public class myHello     {

  //harus sama dengan nama file (myHello.java)

     }

1. Comment sebaiknya sebaiknya ditulis untuk menjelaskan sebuah class atau method. Comment ditulis  untuk tujuan dokumentasi.

Contoh:

public static void main(String[]args){

//untuk menampilkan kata myfirs java

System.out.println(“Hello this is my first java”);

}

1. Java statement adalah suatu baris yang diakhiri dengan titik koma.

Contoh:

System.out.println(“Hello this is my first java”);

1. Block adalah satu atau beberapa statement yang berada diantara tanda kurung kurawal { dan diakhiri dengan kurung kurawal }.

Contoh:

public static void main(String[]args){

System.out.println(“Hello this is my first java”);

System.out.println(“Java is very good”);

}