```
public class DemoFinally{
   public static void main (String [] args){
      int x = 3;
      int [] arr = {10,11,12};

      try {
            System.out.println(arr[x]);
            System.out.println("Tidak terjadi exception");
      }catch(ArrayIndexOutOfBoundsException e) {
            System.out.println("Terjadi exception");
            System.out.println(arr[x-4]);
      }
      finally {
            System.out.println("Program Selesai");
      }
   }
}
```

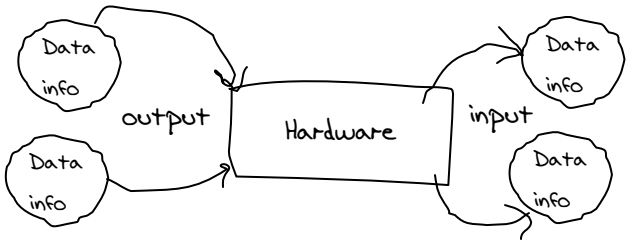
arr[3], padahal isi arr cuma ampe 2

arr[-1], kan arr ke-negative itu gak ada

Dieksekusi meskipun di catch ada exception

try this at home

I/O (Input dan Output)



Input: mengambil informasi data dari hardware
Output: mengirim informasi data ke hardware

sebuah program membuka stream dari sebuah sumber informasi (source) seperti file, memory, socket. Kemudian membaca informasi secara sekuensial.

Stream adalah aliran proses informasi data yang direpresentasikan secara abstrak untuk untuk menulis/menghasilkan/output dan membaca/mendapatkan suatu informasi/input.

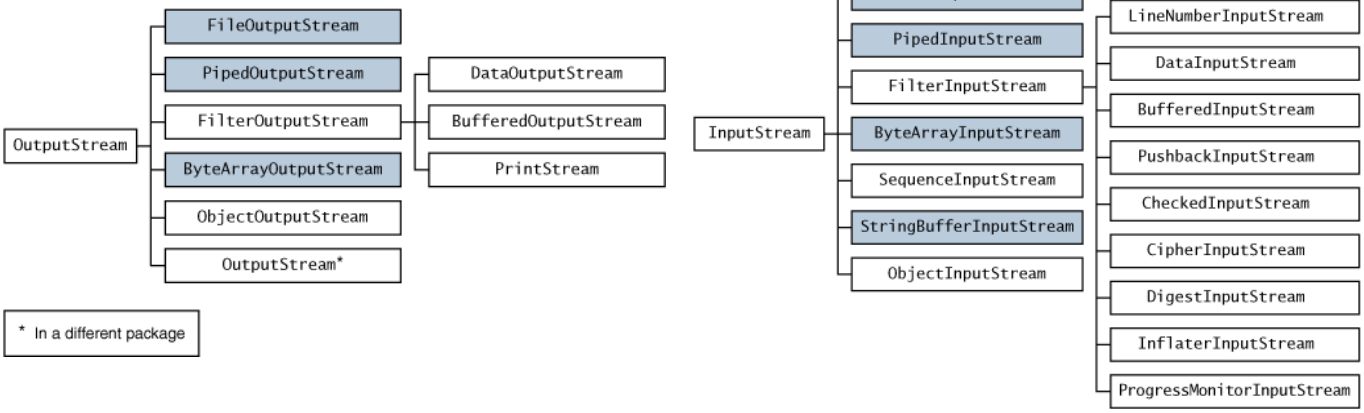
semua streams punya sifat yang sama meskipun peralatan fisik yang berhubungan dengannya beda-beda (missal: keyboard, jaringan, dll.)

sebuah program dapat mengirim informasi dengan membuka stream ke sebuah tujuan informasi (destination).Kemudian menuliskan informasi ke tujuan secara sekuensial.

Paket yang menangani pembacaan dan penulisan info untuk pemrograman java, java io. Ada 2 tipe, Byte Stream dan Character Stream

Byte Stream

Byte Streams digunakan untuk operasi I/O yang menggunakan data biner (byte). Pada java byte stream mempunyai 2 buah superclass yaitu InputStream dan OutputStream yang merupakan class abstract.



DataInputStream dalam jdk 1.5 hanya mempunyai 1 konstruktor dan 17 method yang ada dalam classnya, selebihnya turunan dari class parentnya. Berikut ini akan dijelaskan konstruktor dan method yang sering digunakan.

DataInputStream(InputStream in)
Untuk membuat sebuah objek DataInputStream dengan spesifikasi InputStream yang diinginkan.

xxx readXxx()
xxx disini dapat diganti dengan tipe data primitif seperti int, float, byte, boolean, byte, char, dll. Digunakan untuk membaca dari stream tipe data tertentu secara langsung.

DataOutputStream dalam jdk 1.5 hanya mempunyai 1 kostruktor dan 15 method yang ada dalam classnya selebihnya turunan dari class parentnya. Berikut ini akan dijelaskan konstruktor dan method yang sering digunakan.

DataOutputStream(OutputStream out)
Untuk membuat sebuah objek DataOutputStream dengan spesifikasi OutputStream yang diinginkan.

void writeXxx(xxx v)
xxx disini dapat diganti dengan tipe data primitif seperti int, float, byte, boolean, byte, char, dll.

Output untuk contoh input data

```
Masukkan data : commonlabz

Yang anda ketik : commonlabz
```

Output untuk contoh input dan output data

```
Masukkan data : common

Yang anda ketik : common
```

Kira-kira seperti apa ya bentuk kodenya?

Contoh input data

try this at home

```
import java.io.*;
public class DemoStream1
{
   public static void main(String[] args) {
      byte[] data = new byte[10];
      System.out.print("Masukkan data    : ");

      try {
            System.in.read(data);
      } catch (IOException e) {
            System.out.print("Terjadi Exception");
      }

      System.out.print("Yang anda ketik : ");

      for (int i=0;i<data.length;i++) {
            System.out.print((char)data[i]);
            //(cahr) diatas disebut posting yakni untuk
            mengubah format menjadi char
      }
   }
}
```

Contoh input dan output data

```
import java.io.*;
public class DemoStream3
{
   public static void main(String[] args) {
      byte[] data = new byte[10];
      int panjang=0;
      System.out.print("Masukkan data     : ");
      try {
            panjang=System.in.read(data);
//sebenarnya Sistem.in.read mengembalikan panjang karakter yang
//diinputkan (termasuk enter  yang dianggap 2 karakater..)
            System.out.print("Yang anda ketik  : ");
            System.out.write(data);
            System.out.println("Panjang Karakter : "+panjang);
            System.out.print("index ke-1 sebnyk 3 : ");
            System.out.write(data,1,3);
      } catch (IOException e) {
            System.out.print("Terjadi Exception");
      }
   }
}
/* write mencetak apapun tipe data yang ada, sedangkan print dan
println mencetak data ke dalam tipe string */
```

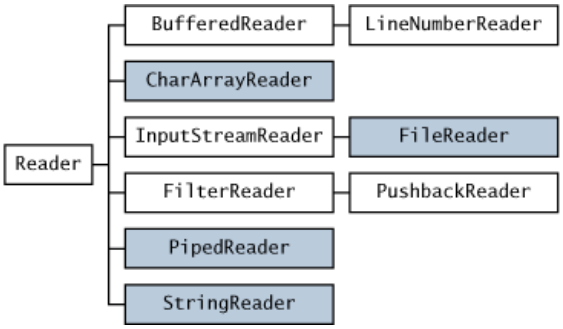
Character Stream

Apa itu character stream?

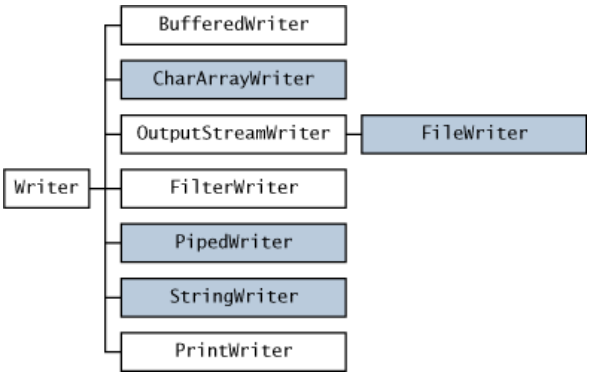
Digunakan untuk menangani operasi I/O yang menggunakan character dan merupakan sebuah objek yang dapat membaca dan menuliskan byte stream, kayak byte stream itu sendiri.

Jadi, character stream itu adalah sebuah byte stream yang diteruskan oleh class Reader dan Writer yang merupakan Abstract class.

FYI, karena karakter dalam java menggunakan Unicode (16 bit length) maka penggunaan character streams dapat digunakan untuk menangani karakter-karakter internasional (karakter diluar ASCII standar).



Hierarki Kelas Reader



Hierarki Kelas Writer

BufferedReader dalam jdk 1.5 hanya mempunyai 2 kostruktor dan 9 method yang ada dalam classnya selebihnya turunan dari class parentnya. Berikut ini akan dijelaskan konstruktor dan method yang sering digunakan.

BufferedReader(Reader in) dan
BufferedReader(Reader in,int size)
Membuat objek BufferedReader dengan karakter buffer inputstream atau objek Reader yang lain. Int sz digunakan untuk ukuran buffer yang digunakan.

String readLine()
readLine digunakan untuk membaca satu baris penuh text, yaitu mengembalikan String dari objek BufferedReader yang digunakan.

BufferedWriter dalam jdk 1.5 hanya mempunyai 2 kostruktor dan 6 method yang ada dalam classnya selebihnya turunan dari class parentnya. Berikut ini akan dijelaskan konstruktor dan method yang sering digunakan.

BufferedWriter (Writer in) dan
BufferedWriter(Writer in,int size)
Membuat objek BufferedWriter dengan karakter buffer outputstream atau objek Writer yang lain. Int sz digunakan untuk ukuran buffer yang digunakan.

void write(String s, int of, int len)
Untuk menuliskan sebuah String s ke sebuah media yang telah dispesifikasikan oleh objek BufferedWriter. Dimulai dari karakter of(integer), ke berapa banyak karakter yang akan ditulis len(integer).

Jadi jika ingin membaca dari console dengan Character Streams, dapat ditulis dengan cara :

```
InputStreamReader input = new InputStreamReader(System.in);
BufferedReader buff = new BufferedReader(input);
```

atau

```
BufferedReader buff = new BufferedReader(new InputStreamReader(System.n));
```

Contoh input:

try this at home

```
import java.io.*;
public class DemoStream6 {
      public static void main(String[] args) throws IOException {
            char data;
            String str="";
            BufferedReader buff =
                  new BufferedReader(new InputStreamReader(System.in));
            System.out.print("Ketik  : ");
            data = (char) buff.read();
            while (data!='\r') {
                  str+=data;
                  data = (char) buff.read();
            }
            System.out.println("Yang diketik : "+str);
            System.out.println("Program Selesai");
      }
}
```

Outputnya:

```
Ketik : common

Yang diketik : common
```

Contoh output:

try this at home

```
import java.io.*;
public class DemoStream8 {
      public static void main(String[] args) throws IOException {
            PrintWriter output = new PrintWriter (System.out,true);
            output.println("Hello World");
      }
}
```

Outputnya:

```
Hello World
```

additional stuff

Variabel Streams Standar

Secara default, Java telah menyediakan 3 buah variabel streams yang dapat langsung digunakan, karena variabel ini member public static dari class System, yaitu : in,out,err.
System.out : output stream standar. Secara default outputnya adalah console.
System.in : input stream standar. Secara default inputya adalah keyboard.
System.err : output stream untuk mencetak pesan kesalahan pada console (default).
(System boleh langsung diakses karena println bertipe static sama dengan main, sehingga dapat langsung dijalankan tanpa melalui instansiasi)

FYI, kalo kamu mo instan objek dari paket java io, error harus selalu ditangkap oleh Exception, seperti IOException yang digunakan untuk menangkap kesalahan dari error IO.