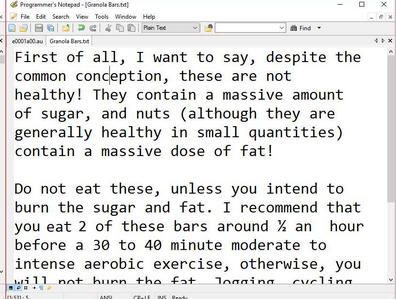
**MODUL 2**

**LOCAL FILE STORAGE**

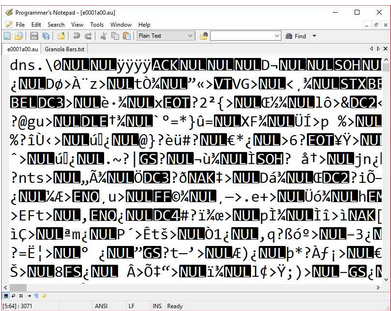
1. TUJUAN
   1. Mahasiswa mengenali ragam file dalam local computer dengan Java
   2. Mahasiswa dapat melakukan penyimpanan file dalam local computer dengan Java
   3. Mahasiswa dapat melakukan pembacaan file dalam local computer dengan Java
2. ALOKASI WAKTU 1 x 50 menit
3. DASAR TEORI
4. **File**

File adalah kumpulan data yang disimpan dalam sebuah perangkat penyimpanan seperti hard drive, flash drive, atau media penyimpanan lainnya. File dapat berisi berbagai jenis informasi, seperti teks, gambar, audio, video, program komputer, dan sebagainya. File digunakan untuk menyimpan dan mengorganisir data agar dapat diakses dan digunakan oleh perangkat lunak dan pengguna komputer.

Dalam dunia komputer terdapat dua kategori file, yakni plaintext file dan binary file. Plaintext file merupakan file yang umumnya dapat dibaca dengan jelas hanya dengan menggunakan text editor saja. Plaintext file biasa ditulis dengan menggunakan ASCII. Contoh dari plaintext file adalah file dengan format .txt dan .ini.



Selanjutnya, file binary adalah file yang pada dasarnya juga terdiri dari kumpulan text, namun tidak dapat dengan mudah dibaca oleh manusia karena sudah terstandar secara khusus tergantung dengan format filenya. Sebaliknya, binary file justru lebih mudah untuk dibaca bagi komputer itu sendiri. Namun demikian, untuk dapat membacanya perlu menggunakan tools khusus. Contohnya adalah file music seperti .mp3 dan .wav yang dapat dibaca hanya dengan music player, atau .jpg dan .png yang dapat dibuka oleh photo viewer. Jika binary file dibaca dengan text editor biasa, maka hasilnya menjadi karakter dan symbol saja.



Perbedaan utama antara plaintext file dan binary file adalah cara data disimpan di dalamnya:

* 1. Plaintext File
* File teks berisi karakter yang dapat dibaca oleh manusia, seperti huruf, angka, simbol, dan whitespace (spasi, tab, newline).
* Data dalam file teks disimpan dalam format teks atau teks biasa, yang berarti tidak ada enkripsi atau kompresi khusus yang digunakan.
* File teks umumnya digunakan untuk menyimpan teks sederhana, seperti dokumen, kode sumber, atau data tabular dalam format yang dapat dibaca manusia, seperti CSV (Comma-Separated Values) atau TXT (Plain Text).
  1. Binary File
* File biner berisi data dalam bentuk biner atau kode mesin yang tidak dapat dibaca langsung oleh manusia.
* Data dalam file biner seringkali disusun dalam format yang lebih efisien, yang dapat mencakup berbagai jenis informasi, seperti gambar, audio, video, atau data yang digunakan oleh program komputer.
* File biner bisa jauh lebih kompleks dan fleksibel dalam hal jenis data yang dapat disimpan, tetapi tidak dapat dibaca dengan mudah tanpa pemahaman yang mendalam tentang struktur dan formatnya.

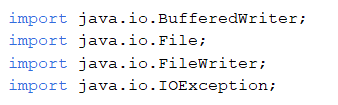
1. **Text File**

**Write Text File**

Untuk menulis text file dengan Java, lakukan langkah berikut:

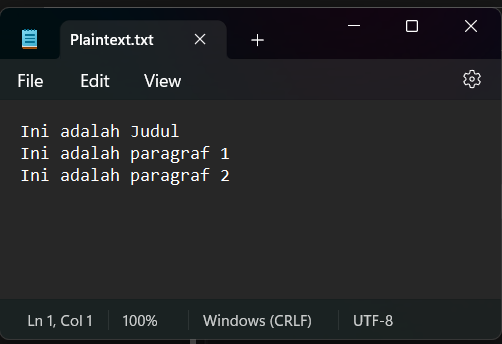
1. Buat objek File yang mewakili file teks.
2. Buat objek FileWriter untuk merujuk pada file.
3. Buat objek BufferedWriter untuk membuat buffer untuk FileWriter.
4. Masukkan tulisan ke file menggunakan method write().
5. Tutup objek BufferedWriter.

Perhatikan contoh berikut:

* Impor library yang dibutuhkan   
   
* Tulis fungsi menulis file

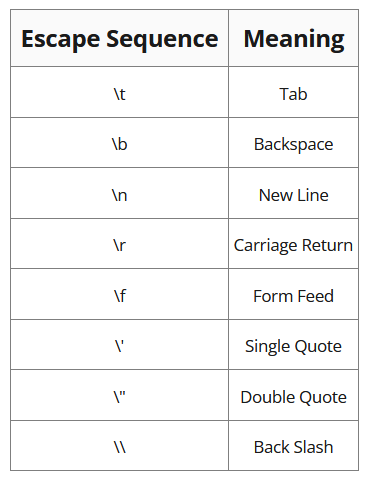


* Pada folder project kita akan muncul file dengan nama “Plaintext.txt” dan jika dibuka akan berisi data berikut



**Note:**

Dalam menulis file kita dapat memberikan escape sequence seperti baris baru, tab atau backspace. Berikut tabel beberapa escape sequence.

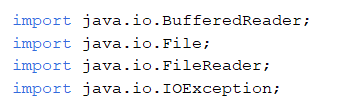


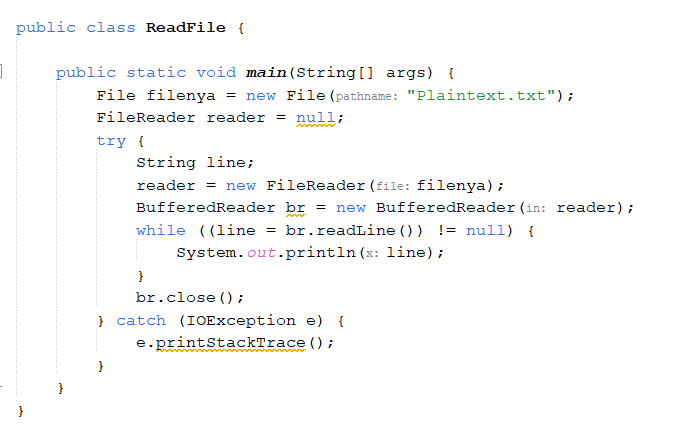
**Read Text File**

Untuk membaca text file, lakukan langkah berikut:

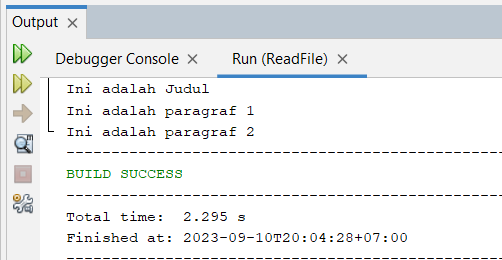
1. Buat objek File yang mewakili file teks.
2. Buat objek FileReader untuk membaca file.
3. Buat objek BufferedReader untuk membaca teks melalui buffer.
4. Baca file menggunakan method readLine().
5. Tutup objek BufferedReader.

Perhatikan contoh berikut:

* + Impor library yang dibutuhkan   
     
  + Pastikan kita menyediakan file yang akan dibaca. Pada contoh menggunakan file “Plaintext.txt” yang baru saja kita buat.
  + Tulis fungsi membaca file



* + Pada output Netbeans project kita akan muncul isi file “Plaintext.txt” kita



1. **Binary File**

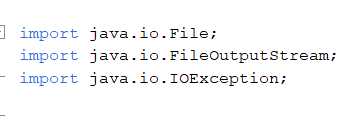
**Write Binary File**

Untuk menulis binary file dengan Java, lakukan langkah berikut:

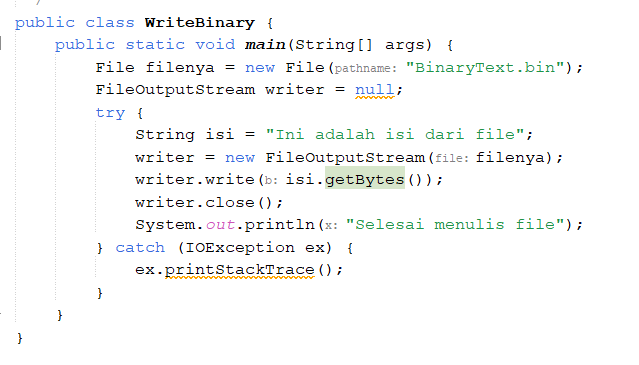
1. Buat objek File yang mewakili binary file.
2. Buat objek FileOutputStream untuk merujuk ke file.
3. Masukkan tulisan ke file menggunakan method write().
4. Tutup objek FileOutputStream.

Perhatikan contoh berikut:

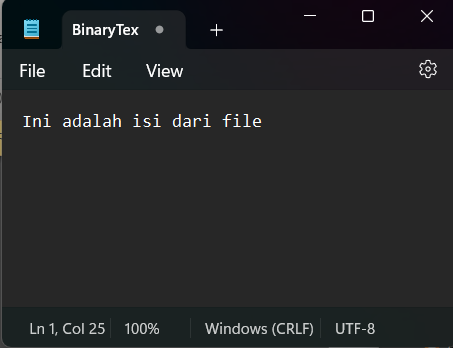
* Impor library yang dibutuhkan



* Tulis fungsi menulis file



Pada folder project kita akan muncul file dengan nama “Binary.bin” dan jika dibuka dengan text editor akan berisi data berikut

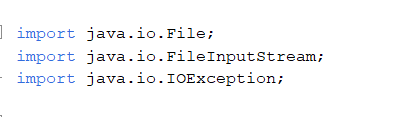


**Read Text File**

Untuk membaca binary file, lakukan langkah berikut:

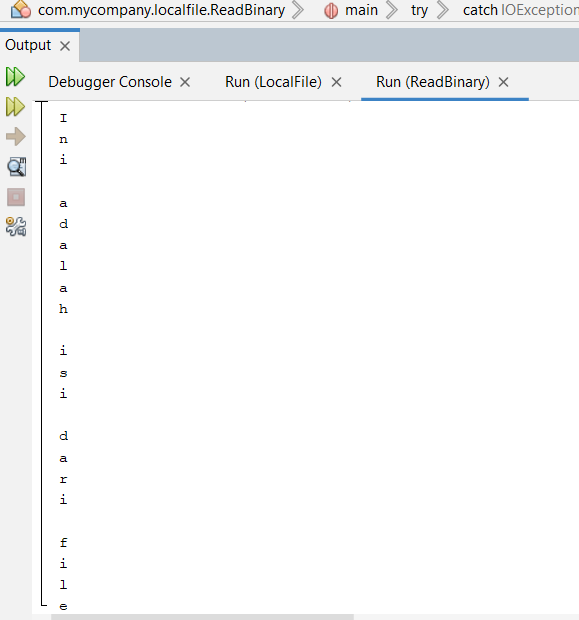
1. Buat objek File yang mewakili binary file.
2. Buat objek FileInputStream untuk membaca file.
3. Baca file menggunakan method read().
4. Tutup objek FileInputStream

Perhatikan contoh berikut:

* + Impor library yang dibutuhkan   
     
  + Pastikan kita menyediakan file yang akan dibaca. Pada contoh menggunakan file “Binary.bin” yang baru saja kita buat.
  + Tulis fungsi membaca file



* + Pada output Netbeans project kita akan muncul isi file “Binary.bin” kita



# PRAKTIKUM :

# Praktikum 1

* Buatlah class Print.java

# 

* Jalankan lalu apa hasilnya?
* Apa perbedaan antara penyimpanan teks menggunakan PrintWriter dan BufferedWriter?

# 

# 

# Praktikum 2

* Buatlah class Mahasiswa.java



* Buatlah kelas main dengan nama SimpanMahasiswa.java

# 

# 

* Jalankan program tersebut! Apa hasilnya?
* Jika file Objek.bin dibuka dengan text editor, apakah yang terjadi?
* Buatlah class baru yang dapat membaca file Objek.bin tersebut!

# LATIHAN

# Buatlah sebuah file yang dapat menulis sebuah file dengan nama “input.txt”. File tersebut kemudian dikonversi sedemikian rupa sehingga semua teks lowercase berubah menjadi uppercase dan begitu pula sebaliknya. Simpan hasil konversi pada file yang diberi nama “output.txt”. Simpan kode ini dengan nama latihan1.java!

# Buatlah sebuah file yang dapat membuat duplikasi dari sebuah gambar. Gambar yang dipilih bebas namun ubah namanya menjadi “gambarasli.jpg”. Gambar hasil duplikasi nantinya harus sama dengan gambar awal dan diberi nama “gambarduplikat.jpg”. Simpan kode ini dengan nama latihan2.java!

# Buatlah sebuah program yang dapat melakukan manajemen file ASCII/plaintext. Operasi yang perlu tersedia adalah membuat file, mengganti nama file dan menghapus file. Gunakan method file renameTo() untuk mengganti nama file dan delete() untuk menghapus file. Simpan dengan nama latihan3.java!

Manajemen File

1. Buat file
2. Ubah nama file
3. Hapus file
4. Keluar

===============================

Masukkan perintah : 1

Buat nama file : baru.txt

Isi file : ((isi file dari baru.txt))

File berhasil dibuat

===============================

Masukkan perintah : 2

Nama file yang akan diubah : baru.txt

Nama file yang baru : lebih\_baru.txt

Nama file berhasil diubah

===============================

Masukkan perintah : 3

Nama file yang akan dihapus : lebih\_baru.txt

File berhasil dihapus

===============================

Masukkan perintah : 4

Sampai jumpa!