
Nama : Akhmad Syafiul Anam
NIM : 245150707111012
Kelas : TI - A
Bab : BAB 10 (OPERASI FILE)
Asisten : 1. Ketut Bagus Wedanta Ananda Murti
2. Gantang Satria Yudha.

LANGKAH 1

A. Soal

Tugas Praktikum

1. Jalankan kode TulisFile1.java beberapa kali dan amati yang terjadi pada file yang ditulis. Kemudian, Pada baris 13, ubah parameter `false` menjadi `true`. Kemudian jalankan kode tersebut beberapa kali dan amati yang terjadi pada *file* yang ditulis.
2. Buat kode program untuk mendapatkan ukuran *file* dalam satuan KB jika ukuran *file* tersebut < 1 MB dan dalam satuan MB jika ukuran *file* tersebut >= 1 MB.
3. Buat kode program untuk menampilkan nama dari semua *file* yang ada di dalam suatu direktori. Petunjuk: gunakan perulangan dan *method* `list()` atau `listFiles()`.
4. Buat kode program untuk menghapus suatu direktori beserta semua *file* yang ada di dalamnya. Asumsi: di dalam direktori tersebut, hanya ada *file-file* saja, tidak ada subdirektori.
5. Apakah yang salah dengan *statement* berikut? Berikan penjelasan.

```
var file = new File("C:\Data\Java\teks.txt");
```
6. Apa yang akan terjadi jika kita mencoba untuk membaca isi dari suatu *file* tetapi *file* tersebut tidak ada dan kita tidak melakukan pengecekan lebih dulu?

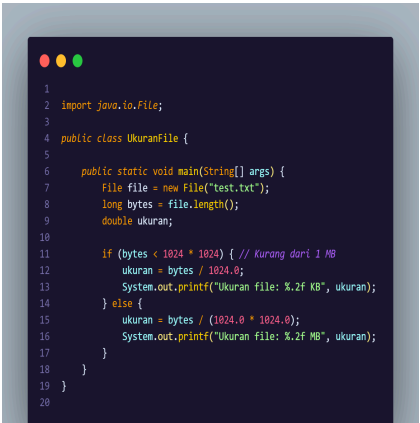
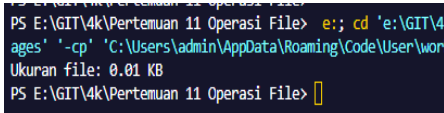
B. Screenshoot

TulisFile.Java	Output
----------------	--------

The image is a composite of two screenshots. The left screenshot shows a Java IDE with a dark theme. It displays a Java class named `TulisFile` with a `main` method. The code uses `Scanner` to read input from the user and `FileWriter` to write the input to a file named `test.txt`. The code is as follows:

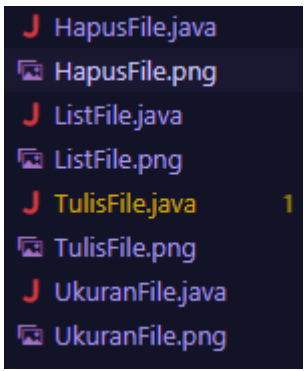
```
1
2 import java.io.FileWriter;
3 import java.io.IOException;
4 import java.util.Scanner;
5
6 public class TulisFile {
7
8     public static void main(String[] args) {
9         var keyboard = new Scanner(System.in);
10         System.out.print("Masukkan teks yang akan disimpan: ");
11         var text = keyboard.nextLine();
12
13         try (var writer = new FileWriter("test.txt", true)) {
14             writer.write(text);
15         } catch (IOException e) {
16             System.err.println("gagal menulis ke file");
17         }
18     }
19 }
20
```

The right screenshot shows a Windows command prompt with a dark background. It shows the execution of the Java program. The prompt is at `PS E:\GIT\4k\Pertemuan 11 Operasi File>`. The user enters `java -cp 'C:\Users\admin\AppData\Roaming\Code\User' HapusFile`. The prompt then shows `PS E:\GIT\4k\Pertemuan 11 Operasi File>` with a cursor. Below the command prompt, there is a list of files in a dark-themed window, showing `HapusFile.java`, `HapusFile.png`, `ListFile.java`, `ListFile.png`, `test.txt`, `TulisFile.java`, `TulisFile.png`, `UkuranFile.java`, and `UkuranFile.png`. The file `TulisFile.java` is highlighted with a yellow background.

<h2>UkuranFile.java</h2>	<h2>Output</h2>
 <pre>1 2 import java.io.File; 3 4 public class UkuranFile { 5 6 public static void main(String[] args) { 7 File file = new File("test.txt"); 8 long bytes = file.length(); 9 double ukuran; 10 11 if (bytes < 1024 * 1024) { // Kurang dari 1 MB 12 ukuran = bytes / 1024.0; 13 System.out.printf("Ukuran file: %.2f KB", ukuran); 14 } else { 15 ukuran = bytes / (1024.0 * 1024.0); 16 System.out.printf("Ukuran file: %.2f MB", ukuran); 17 } 18 } 19 } 20</pre>	 <pre>PS E:\GIT\4k\Pertemuan 11 Operasi File> PS E:\GIT\4k\Pertemuan 11 Operasi File> e:: cd 'e:\GIT\4k\ ages' -cp 'C:\Users\admin\AppData\Roaming\code\User\work Ukuran file: 0.01 KB PS E:\GIT\4k\Pertemuan 11 Operasi File></pre>

ListFile.java	Output
---------------	--------

<pre> 1 2 import java.io.File; 3 4 public class ListFile { 5 6 public static void main(String[] args) { 7 File dir = new File("."); 8 String[] files = dir.list(); 9 10 System.out.println("Daftar file:"); 11 for (String file : files) { 12 System.out.println(file); 13 } 14 } 15 } 16 </pre>	<pre> PS E:\GIT\4k\Pertemuan 11 Operasi File> cd .. PS E:\GIT\4k\Pertemuan 11 Operasi File> PS E:\GIT\4k\Pertemuan 11 Operasi File> e;; cd 'e:\ ages' '-cp' 'C:\Users\admin\AppData\Roaming\Code\Use Daftar file: HapusFile.java HapusFile.png ListFile.java ListFile.png test.txt TulisFile.java TulisFile.png UkuranFile.java UkuranFile.png PS E:\GIT\4k\Pertemuan 11 Operasi File> </pre>
---	---

HapusFile.java	Output
<pre> 1 import java.io.File; 2 3 public class HapusFile { 4 public static void main(String[] args) { 5 File file = new File("test.txt"); // Ganti dengan path Lengkap jika perlu 6 7 if (file.exists() && file.isFile()) { 8 boolean berhasil = file.delete(); 9 if (berhasil) { 10 System.out.println("File test.txt berhasil dihapus."); 11 } else { 12 System.out.println("Gagal menghapus file test.txt."); 13 } 14 } else { 15 System.out.println("File test.txt tidak ditemukan atau bukan file."); 16 } 17 } 18 } </pre>	<pre> ages' '-cp' 'C:\Users\admin\AppData\Roaming\Code\Use File test.txt berhasil dihapus. PS E:\GIT\4k\Pertemuan 11 Operasi File> ^C PS E:\GIT\4k\Pertemuan 11 Operasi File> PS E:\GIT\4k\Pertemuan 11 Operasi File> e;; cd 'e:\ ages' '-cp' 'C:\Users\admin\AppData\Roaming\Code\Use File test.txt berhasil dihapus. PS E:\GIT\4k\Pertemuan 11 Operasi File> </pre> 

C. Syntax

	<pre> //TulisFile.java import java.io.FileWriter; import java.io.IOException; import java.util.Scanner; public class TulisFile { </pre>
--	---

```

    public static void main(String[] args) {
        var keyboard = new Scanner(System.in);
        System.out.print("Masukkan teks yang akan disimpan:");
        var text = keyboard.nextLine();

        try (var writer = new FileWriter("test.txt", true)) {
            writer.write(text);
        } catch (IOException e) {
            System.err.println("Gagal menulis ke file");
        }
    }
}

```

```

//UkuranFile.java
import java.io.File;

public class UkuranFile {

    public static void main(String[] args) {
        File file = new File("test.txt");
        long bytes = file.length();
        double ukuran;

        if (bytes < 1024 * 1024) { // Kurang dari 1 MB
            ukuran = bytes / 1024.0;
            System.out.printf("Ukuran file: %.2f KB", ukuran);
        } else {
            ukuran = bytes / (1024.0 * 1024.0);
            System.out.printf("Ukuran file: %.2f MB", ukuran);
        }
    }
}

```

```

//ListFile.java
import java.io.File;

public class ListFile {

```

```

    public static void main(String[] args) {
        File dir = new File(".");
        String[] files = dir.list();

        System.out.println("Daftar file:");
        for (String file : files) {
            System.out.println(file);
        }
    }
}

```

```

//HapusFile.java
import java.io.File;

public class HapusFile {
    public static void main(String[] args) {
        File file = new File("test.txt"); // Ganti dengan path
        Lengkap jika perlu

        if (file.exists() && file.isFile()) {
            boolean berhasil = file.delete();
            if (berhasil) {
                System.out.println("File test.txt berhasil
dihapus.");
            } else {
                System.out.println("Gagal menghapus file
test.txt.");
            }
        } else {
            System.out.println("File test.txt tidak ditemukan
atau bukan file.");
        }
    }
}

```

D. Penjelasan

Penjelasan TulisFile.java

```

//TulisFile.java

```

```

1. import java.io.FileWriter;

```

```

2. import java.io.IOException;
3. import java.util.Scanner;
4.
5. public class TulisFile {
6.
7.     public static void main(String[] args) {
8.         var keyboard = new Scanner(System.in);
9.         System.out.print("Masukkan teks yang akan disimpan: ");
10.        var text = keyboard.nextLine();
11.
12.        try (var writer = new FileWriter("test.txt", true)) {
13.            writer.write(text);
14.        } catch (IOException e) {
15.            System.err.println("Gagal menulis ke file");
16.        }
17.    }
18.}

```

- **Baris 1:** Mengimpor class `FileWriter` dari package `java.io` yang digunakan untuk menulis karakter ke dalam file.
- **Baris 2:** Mengimpor `IOException` yang digunakan untuk menangani error yang berkaitan dengan operasi input/output.
- **Baris 3:** Mengimpor `Scanner` dari package `java.util`, digunakan untuk membaca input dari pengguna (dalam hal ini dari keyboard).
- **Baris 5:** Mendeklarasikan class publik dengan nama `TulisFile`. Nama ini juga harus sama dengan nama file Java (`TulisFile.java`).
- **Baris 6:** Metode utama yang akan dijalankan ketika program dimulai. `main` adalah entry point untuk program Java.
- **Baris 7:**
 - Membuat objek `Scanner` bernama `keyboard` untuk membaca input dari pengguna melalui `System.in` (keyboard).
 - `var` digunakan untuk mendeklarasikan variabel tanpa menyebutkan tipe eksplisitnya, Java akan menyimpulkan tipe sebagai `Scanner`.
- **Baris 8:** Menampilkan pesan kepada pengguna agar memasukkan teks yang ingin disimpan ke dalam file.
- **Baris 9:**
 - Membaca satu baris penuh yang diketik oleh pengguna dan menyimpannya dalam variabel `text`.
 - `nextLine()` membaca sampai pengguna menekan Enter.
- **Baris 11:**
 - Membuka file `test.txt` menggunakan `FileWriter` untuk menulis data ke dalamnya.
 - `true` berarti mode **append**, sehingga data akan ditambahkan di akhir file, bukan menimpa isi sebelumnya.

- `try(...)` adalah **try-with-resources**: otomatis menutup file setelah digunakan, meskipun terjadi error.
- **Baris 12:** Menulis teks dari pengguna ke file `test.txt`. (Jika ingin menulis ke baris baru setiap kali, tambahkan `"\n"`.)
- **Baris 13:** Jika terjadi error saat membuka atau menulis file (misalnya file tidak bisa dibuat), maka akan masuk ke blok `catch`.
- **Baris 14:** Menampilkan pesan error ke layar jika gagal menulis ke file. Menggunakan `System.err` agar jelas bahwa itu error.
- **Baris 15:** Menutup blok `try-catch`.
- **Baris 16-17:** Menutup metode `main` dan class `TulisFile`

Penjelasan UkuranFile.java.

```
//UkuranFile.java
1. import java.io.File;
2.
3. public class UkuranFile {
4.
5.     public static void main(String[] args) {
6.         File file = new File("test.txt");
7.         long bytes = file.length();
8.         double ukuran;
9.
10.        if (bytes < 1024 * 1024) { // Kurang dari 1 MB
11.            ukuran = bytes / 1024.0;
12.            System.out.printf("Ukuran file: %.2f KB", ukuran);
13.        } else {
14.            ukuran = bytes / (1024.0 * 1024.0);
15.            System.out.printf("Ukuran file: %.2f MB", ukuran);
16.        }
17.    }
18. }
```

- **Baris 1:** Mengimpor class `File` dari package `java.io`, yang digunakan untuk merepresentasikan file dan direktori di Java.
- **Baris 4:** Mendefinisikan kelas publik `UkuranFile`, yang akan digunakan untuk menjalankan program.
- **Baris 6:** Method `main()` adalah titik masuk utama program.
- **Baris 7:** Membuat objek `File` yang menunjuk ke file bernama `test.txt`. Program akan mencari file ini di direktori kerja saat ini (biasanya folder project).
- **Baris 8:** Mendapatkan ukuran file `test.txt` dalam satuan **byte** dan menyimpannya ke variabel `bytes`.
- **Baris 9:** Mendeklarasikan variabel `ukuran` yang akan menyimpan ukuran file dalam KB atau MB sebagai `double`.
- **Baris 11:** Mengecek apakah ukuran file kurang dari 1 MB (1 MB = 1024 * 1024 byte = 1.048.576 byte).

- **Baris 12-13:** Jika ukuran file kurang dari 1 MB, konversi ke **kilobyte (KB)** dan cetak hasilnya dengan 2 angka di belakang koma.
- **Baris 14-17:** Jika ukuran file lebih dari atau sama dengan 1 MB, konversi ke **megabyte (MB)** dan tampilkan.
- **Baris 18-19:** Menutup method `main()` dan class `UkuranFile`.

Penjelasan ListFile.java

```
//ListFile.java
1. import java.io.File;
2.
3. public class ListFile {
4.
5.     public static void main(String[] args) {
6.         File dir = new File(".");
7.         String[] files = dir.list();
8.
9.         System.out.println("Daftar file:");
10.        for (String file : files) {
11.            System.out.println(file);
12.        }
13.    }
14. }
```

- **Baris 1:** Mengimpor class `File` dari package `java.io` untuk mengakses file dan direktori.
- **Baris 4:** Mendefinisikan kelas `ListFile`, nama kelas bebas, tapi harus sama dengan nama file `.java`.
- **Baris 6:** Fungsi `main()` adalah titik awal eksekusi program.
- **Baris 7:** Membuat objek `File` yang menunjuk ke direktori saat ini (`."` artinya direktori tempat program dijalankan).
- **Baris 8:** Mengambil **daftar nama file dan folder** dalam direktori yang dimaksud, dan menyimpannya dalam array `files`.
- **Baris 10:** Menampilkan judul daftar file.
- **Baris 11-13:** Melakukan perulangan terhadap setiap elemen dalam array `files` dan mencetak nama file atau folder ke layar.

Penjelasan HapusFile.java

```
//HapusFile.java
1. import java.io.File;
2.
3. public class HapusFile {
4.     public static void main(String[] args) {
5.         File file = new File("test.txt"); // Ganti dengan path lengkap
        jika perlu
6.    }
```



```

7.         if (file.exists() && file.isFile()) {
8.             boolean berhasil = file.delete();
9.             if (berhasil) {
10.                System.out.println("File test.txt berhasil dihapus.");
11.            } else {
12.                System.out.println("Gagal menghapus file test.txt.");
13.            }
14.        } else {
15.            System.out.println("File test.txt tidak ditemukan atau
            bukan file.");
16.        }
17.    }
18.}

```

- **Baris 1:** Mengimpor class `File` dari package `java.io` untuk mengakses dan memanipulasi file atau direktori.
- **Baris 2–3:** Baris kosong untuk kerapian dan keterbacaan kode.
- **Baris 4:** Mendefinisikan class `ListFile`, yang merupakan nama class utama dari program (harus sama dengan nama file `ListFile.java`).
- **Baris 6:** Mendeklarasikan method `main`, titik awal eksekusi program Java.
- **Baris 7:** Membuat objek `File` bernama `dir` yang merepresentasikan direktori saat ini ("`.`" artinya folder tempat file `.class` dijalankan).
- **Baris 8:** Mengambil daftar nama semua file dan folder dalam direktori tersebut dan menyimpannya dalam array `files`.
- **Baris 10:** Mencetak tulisan "Daftar file:" sebagai heading sebelum menampilkan isi direktori.
- **Baris 11–13:** Melakukan loop pada setiap elemen array `files` dan mencetak nama masing-masing file/folder.
- **Baris 14–15:** Penutup blok fungsi dan kelas.

Jawaban untuk soal no. 5 dan 6

5. Apakah yang salah dengan statement berikut? Berikan penjelasan.

```
var file = new File("C:\Data\Java\teks.txt");
```

Statement tersebut salah karena:

- **Tanda kutip yang digunakan adalah tanda kutip miring (“ ”), bukan tanda kutip lurus (").** Java hanya mengenali tanda kutip **lurus (")** untuk string.
- **Backslash (\) harus di-escape** karena merupakan karakter khusus dalam string Java. Jika kamu menulis `\D`, Java akan menganggap itu sebagai escape sequence yang tidak valid.

Perbaikan yang benar:

```
var file = new File("C:\\Data\\Java\\teks.txt");
```

Atau bisa juga menggunakan slash biasa (/), yang juga valid di Java:

```
var file = new File("C:/Data/Java/teks.txt");
```

6. Apa yang akan terjadi jika kita mencoba untuk membaca isi dari suatu file tetapi file tersebut tidak ada dan kita tidak melakukan pengecekan lebih dulu?

Jika mencoba membaca file yang tidak ada tanpa melakukan pengecekan lebih dulu, maka program akan melempar exception seperti:

```
java.io.FileNotFoundException
```

Ini akan menyebabkan program berhenti (crash) jika tidak menangkap exception tersebut menggunakan `try-catch`.

Contoh Kode Tanpa Pengecekan:

```
Scanner input = new Scanner(new File("data.txt")); // jika file tidak ada,  
FileNotFoundException terjadi
```

Solusi yang Benar:

Gunakan pengecekan:

```
File file = new File("data.txt");

if (file.exists()) {

    Scanner input = new Scanner(file);

    // lanjut baca file

} else {

    System.out.println("File tidak ditemukan.");
```

```
}
```

Atau tangani dengan `try-catch`:

```
try {  
    Scanner input = new Scanner(new File("data.txt"));  
    // lanjut baca file  
} catch (FileNotFoundException e) {  
    System.out.println("File tidak ditemukan!");  
}
```