

# Input Output dalam Pemrograman Java

## 1. Operasi File dan Direktori

### a. Membuat Direktori baru

- Pada jendela Source, isikan kode program berikut.

```
package fileio;
import java.io.File;
class DemoBuatFolder {
    public static void main(String[] args){
        File f = new File("D:/Winarsih");
        if (f.exists()){
            System.out.println("Direktori sudah ada!");
        }else{
            f.mkdir();
            System.out.println("Sukses membuat folder baru");
        }
    }
}
```

- Jika dijalankan, kode program di atas akan menghasilkan sebuah direktori di drive D: dengan nama Winarsih. Setelah direktori berhasil dibuat, program akan menampilkan pesan "Sukses membuat folder baru!". Gambar berikut ini menunjukkan tampilan output program tersebut dan tampilan folder yang berhasil dibuat.

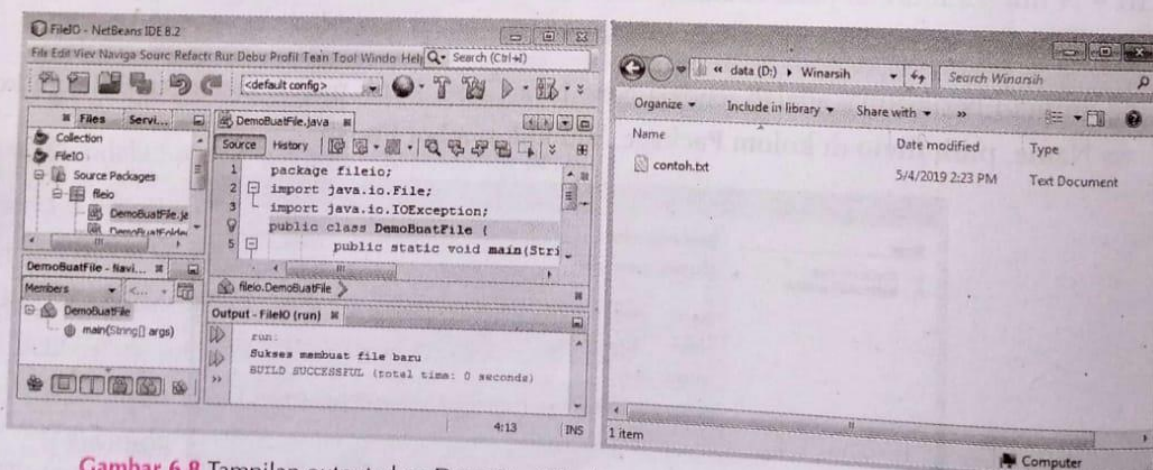
## b. Membuat File Baru

Langkah selanjutnya adalah memasukkan kode program berikut pada jendela Source.

```
package fileio;
import java.io.File;
import java.io.IOException;
public class DemoBuatFile {
```

```
    public static void main(String[] args) throws IOException{
        File f = new File("D:/Winarsih/contoh.txt");
        if(f.exists()){
            System.out.println("File sudah ada");
        }else{
            f.createNewFile();
            System.out.println("Sukses membuat file baru");
        }
    }
}
```

- Jika tidak ada kesalahan (*exception*), file dengan nama **contoh.txt** akan muncul di folder **D:/Winarsih** dan di jendela output akan muncul pesan "Sukses membuat file baru". Gambar berikut menunjukkan tampilan jendela output jika program berhasil dieksekusi dengan baik serta tampilan folder **D:/Winarsih** yang sudah berisi file kosong bernama **contoh.txt**.



**Gambar 6.8** Tampilan output class `DemoBuatFile` dan tampilan folder yang telah berisi file baru.

- Jalankan program sekali lagi dan amati output yang muncul. Pelajari blok program mana yang menyebabkan tampilan tersebut.

### c. Menulis File (Write)

- Isikan kode program berikut pada jendela Source.

```
package fileio;
import java.io.FileWriter;
import java.io.IOException;
public class DemoTulisFile {
    public static void main(String args[])throws IOException{
        FileWriter fw = new FileWriter("D:/Winarsih/contoh.txt");
        fw.write("Selamat Datang!");
        fw.close();
        System.out.println("Berhasil...");
    }
}
```



## d. Membaca Direktori & File (Read)

### c. Membaca Direktori dan File (Read)

Membaca direktori berarti menampilkan isinya, yaitu berupa daftar file maupun direktori. Membaca file berarti menampilkan isi file tersebut, misalnya tulisan pada file teks (\*.txt).

Ada banyak cara untuk membaca sebuah file dan direktori dalam pemrograman Java. Kita dapat menggunakan class **BufferedReader**, class **FileReader**, atau class **File**. Yang akan kita akan praktikkan adalah penggunaan class **File** untuk membaca direktori dan class **FileReader** untuk membaca file teks.

#### 1) Membaca Direktori

Untuk mempraktikkan proses membaca direktori atau menampilkan daftar file dan direktori dengan pemrograman Java pada NetBeans, ikuti langkah-langkah berikut.

- Buatlah class baru bernama **DemoBacaDirektori** pada proyek **FileIO** (package: **fileio**).
- Isikan kode program berikut ini pada jendela **Source**.

```
package fileio;
import java.io.File;
import java.io.IOException;
public class DemoBacaDirektori {
    public static void main(String[] args) throws IOException{
        File dir = new File("D:/Winarsih");
        if(dir.isDirectory()) {
            File[] files = dir.listFiles();
            for(File f : files)
                System.out.println(f.getName());
        }
    }
}
```

## e. Membaca File (Read)

### 2) Membaca File

Kita akan mempraktikkan cara membaca file teks (\*.txt). File teks yang akan kita baca adalah file `contoh.txt` di folder `D:/Winarsih` yang pernah kita buat dan tulisi. Ikuti langkah-langkah berikut.

- Aktifkan NetBeans, buatlah class baru dengan nama **DemoBacaFile** pada project **FileIO** (package **fileio**).
- Isikan kode program berikut pada jendela Source.

```
package fileio;
import java.io.FileNotFoundException;
import java.io.FileReader;
public class DemoBacaFile {
    public static void main(String args[]) throws FileNotFoundException{
        try {
            FileReader fr = new FileReader("D:/Winarsih/sample.txt");
            int i;
            while((i = fr.read())!= -1)
                System.out.print((char)i);
            fr.close();
        }
        catch(Exception e) {
            System.out.println("File tidak ditemukan");
        }
    }
}
```

## f. Mengganti nama File

### d. Mengganti Nama File

Untuk mengganti nama file dalam pemrograman Java kita dapat menggunakan method `renameTo(File dest)` dari class `File`. Andaikan kita akan mengubah nama file `contoh.txt` pada folder `D:/Winarsih` menjadi `sample.txt`. Ikuti langkah-langkah berikut untuk mempraktikannya.

- Aktifkan NetBeans, buatlah class baru dengan nama `DemoGantiNamaFile` pada project `FileIO` (package `fileio`).
- Isikan kode program berikut pada jendela `Source`.

```
package fileio;
import java.io.File;
public class DemoGantiNamaFile {
    public static void main(String[] args){
        File f = new File("D:/Winarsih/contoh.txt");
        if(f.exists()){
            f.renameTo(new File("D:/Winarsih/sample.txt"));
            System.out.println("Sukses mengganti nama file");
        }else{
            System.out.println("File yang dimaksud tidak ada!");
        }
    }
}
```



## g. Menghapus File

### e. Menghapus File (Delete)

Method `delete()` pada class `File` dapat digunakan untuk menghapus sebuah file atau sebuah direktori (folder) kosong. Untuk mempraktikkan proses penghapusan file pada NetBeans, ikuti langkah-langkah berikut.

- Aktifkan NetBeans, gunakan proyek yang sama, yaitu `FileIO` (package `fileio`), buatlah class baru dengan nama `DemoHapusFile`.
- Isikan kode program berikut pada jendela `Source`.

```
package fileio;
import java.io.File;
import java.io.IOException;
public class DemoHapusFile {
    public static void main(String[] args) throws Exception{
        File f = new File("D:/Winarsih/latihan.txt");
        f.delete();
        System.out.println("Sukses menghapus file latihan.txt");
    }
}
```

- Setelah program disimpan dan dijalankan, tampilan di jendela `output` dan tampilan folder setelah penghapusan file akan seperti gambar berikut.