

VERSION 1.0
AGUSTUS 20, 2023



PEMROGRAMAN LANJUT

MODUL 5 – FILE HANDLING (JAVA & PYTHON)

TIM PENYUSUN:

- WILDAN SUHARSO, S.KOM., M.KOM
- LARYNT SAWFA KENANGA
- AHYA NIKA SALSABILA

LAB. INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN LANJUT

PERSIAPAN MATERI

Praktikan harus memahami konsep dasar tentang CRUD dan File Handling dan konsep dasar Bahasa pemrograman Java dan Python.

TUJUAN

1. Memahami konsep dasar File Handling dalam pemrograman.
 2. Mengerti pentingnya penyimpanan dan pengelolaan data dalam file.
 3. Mampu menerapkan operasi CRUD (Create, Read, Update, Delete) pada file.
 4. Memahami bagaimana file handling dapat digunakan dalam aplikasi sehari-hari.
 5. Mengembangkan keterampilan dalam manipulasi data melalui operasi CRUD.
-

TARGET MODUL

1. Mampu merancang dan membuat program yang mampu membuat, membaca, memperbarui, dan menghapus data dalam file.
 2. Memiliki pemahaman yang kuat tentang konsep operasi CRUD dalam konteks file handling.
-

PERSIAPAN SOFTWARE/APLIKASI

1. Java Development Kit.
2. Text Editor / IDE (**Preferably** IntelliJ IDEA, Visual Studio Code, Netbeans, Pycharm etc).
3. Notepad (File txt)

TEORI

A. File Handling dan CRUD

File handling adalah kemampuan suatu bahasa pemrograman untuk membaca dan menulis data dari dan ke file. Dalam konteks CRUD (Create, Read, Update, Delete), file handling memungkinkan kita untuk mengelola data dalam bentuk file. Dalam materi ini, kita akan menjelaskan konsep dasar file handling serta penggunaannya untuk operasi CRUD dalam bahasa pemrograman Java dan Python.

Keyword	Fungsi	Dalam Java	Dalam Python
fopen	Membuka file	FileReader reader = new FileReader("data.txt");	file = open('data.txt', 'r')
fclose	Menutup file	reader.close();	file.close()
fread	Membaca data dari file	BufferedReader bufferedReader = new BufferedReader(reader);	lines = file.readlines()
fwrite	Menulis data ke file	FileWriter writer = new FileWriter("data.txt", true);	file = open('data.txt', 'a')
fprintf	Menulis format data ke file	PrintWriter writer = new PrintWriter("data.txt");	file.write(data)
fscanf	Membaca format data dari file	Scanner scanner = new Scanner(reader);	data = file.read()
fseek	Mengatur posisi pembacaan/tulisan dalam file	-	file.seek(offset)

ftell	Memberikan posisi saat ini dalam file	-	position = file.tell()
rewind	Mengembalikan posisi pembacaan/tulisan ke awal	-	file.seek(0)
remove	Menghapus file	File file = new File("data.txt");	import os os.remove('data.txt')
rename	Mengganti nama file	File newFile = new File("new_data.txt")	os.rename('data.txt', 'new_data.txt')

1. Create

Operasi Create melibatkan pembuatan file baru di dalam sistem operasi. File baru dapat digunakan untuk menyimpan data dalam bentuk teks, biner, atau format lainnya.

Contoh Program Java:

```
import java.io.*;

class create {
    public static void main(String[] args) {
        try {
            FileWriter fileWriter = new FileWriter( fileName: "data_java.txt");
            fileWriter.write( str: "Nama, NIM, Semester\n");
            fileWriter.close();
            System.out.println("File berhasil dibuat.");
        } catch (IOException e) {
            System.out.println("Exception Occurred");
        }
    }
}
```

Contoh Program Python:

```

file = open("data_python.txt", "w")
file.write("Nama, NIM, Semester\n")
file.close()
print("File berhasil dibuat.")

```

2. Read

Operasi Read melibatkan membaca isi dari sebuah file yang sudah ada. Ini memungkinkan program untuk mengambil data dari file dan memprosesnya.

Contoh Program Java:

```

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

class read {
    public static void main(String[] args) {
        try {
            File file = new File( pathname: "data_java.txt");
            Scanner dataReader = new Scanner(file);

            while (dataReader.hasNextLine()) {
                String fileData = dataReader.nextLine();
                System.out.println(fileData);
            }
            dataReader.close();
        } catch (FileNotFoundException exception) {
            System.out.println("Exception Occurred");
        }
    }
}

```

Contoh Program Python:

```

file = open("data_python.txt", "r")
for line in file:
    print(line, end="")
file.close()

```

3. Update

Operasi Update melibatkan pengeditan isi file yang sudah ada. Data dalam file dapat diubah atau ditambahkan dengan informasi baru.

Contoh Program Java:

```
import java.io.*;

class update {
    public static void main(String[] args) {
        try {
            FileWriter fileWriter = new FileWriter("data_java.txt", append: true);
            fileWriter.write(str: "John Doe, 12345, 3\n");
            fileWriter.close();
            System.out.println("Data berhasil ditambahkan.");
        } catch (IOException e) {
            System.out.println("Exception Occurred");
        }
    }
}
```

Contoh Program Python:

```
file = open("data_python.txt", "a")
file.write("Jane Doe, 543222, 2\n")
file.close()
print("Data berhasil ditambahkan.")
```

4. Delete

Operasi Delete melibatkan penghapusan file dari sistem operasi. File yang sudah tidak diperlukan atau usang dapat dihapus untuk menghemat ruang penyimpanan.

Contoh Program Java:

```

import java.io.*;

class delete {
    public static void main(String[] args) {
        try {
            File inputFile = new File( pathname: "data_java.txt");
            File tempFile = new File( pathname: "temp.txt");
            BufferedReader bufferedReader = new BufferedReader(new FileReader(inputFile));
            PrintWriter printWriter = new PrintWriter(new FileWriter(tempFile));

            String lineToRemove = "John Doe, 12345, 3";
            String line;
            while ((line = bufferedReader.readLine()) != null) {
                if (!line.equals(lineToRemove)) {
                    printWriter.println(line);
                }
            }
            printWriter.close();
            bufferedReader.close();
            inputFile.delete();
            tempFile.renameTo(inputFile);
            System.out.println("Data berhasil dihapus.");
        } catch (IOException e) {
            System.out.println("Exception Occurred");
        }
    }
}

```

Contoh Program Python

```

line_to_remove = "Jane Doe, 54321, 2"
with open("data_python.txt", "r") as file:
    lines = file.readlines()
with open("data_python.txt", "w") as file:
    for line in lines:
        if line.strip() != line_to_remove:
            file.write(line)
print("Data berhasil dihapus.")

```

LATIHAN

Terdapat data seperti berikut:

Nama	NIM	Semester
Andi	11119	1
Bima	11112	1
Rahma	11131	3
Zeno	11198	9
Rahma	11131	3
Andi	11119	1

Saya ingin menyimpan file tersebut dengan nama file data_mahasiswa.txt

Saya harus membuat code agar dapat menginput data tersebut sekaligus. (Hint: list, looping)

Kemudian saya sadar ada data yang double/duplikat. Saya ingin menghapus data yang duplikat tersebut. Sehingga datanya akan menjadi seperti ini:

Nama	NIM	Semester
Andi	11119	1
Bima	11112	1
Rahma	11131	3
Zeno	11198	9

Notes: Jika dikerjakan menggunakan bahasa pemrograman **Python** akan menjadi nilai tambah.

TUGAS

BUATLAH APLIKASI SEDERHANA UNTUK DATA MAHASIWA UMM, TERAPKAN PENGGUNAAN FILE HANDLING YANG BISA MELAKUKAN BEBERAPA FITUR SEPERTI BERIKUT:

1. Bisa Create Data
2. Bisa Read Data
3. Bisa Update Data

4. Bisa Delete Data
5. Bisa Search Data (Opsional)

Contoh output:

```
=====APLIKASI KELOLA DATA MAHASISWA=====
```

1. Tambah Data
2. Tampilkan Data
3. Update Data
4. Delete Data
5. Search Data
6. Keluar

Pilih menu: 1

Masukkan Nama: Ahya Nika Salsabila

Masukkan NIM: 2021-187

Masukkan Mata Kuliah: Proglan

Masukkan Semester: 3

Data mahasiswa berhasil ditambahkan.

```
=====APLIKASI KELOLA DATA MAHASISWA=====
```

1. Tambah Data
2. Tampilkan Data
3. Update Data
4. Delete Data
5. Search Data
6. Keluar

Pilih menu: 1

Masukkan Nama: Larynt Sawfa Kenanga

Masukkan NIM: 2021-189

Masukkan Mata Kuliah: Komdat

Masukkan Semester: 3

=====APLIKASI KELOLA DATA MAHASISWA=====

1. Tambah Data
2. Tampilkan Data
3. Update Data
4. Delete Data
5. Search Data
6. Keluar

Pilih menu: 2

Menampilkan Data ke-1:

Nama: Ahya Nika Salsabila

NIM: 2021-187

Mata Kuliah: Proglan

Semester: 3

Menampilkan Data ke-2:

Nama: Larynt Sawfa Kenanga

NIM: 2021-189

Mata Kuliah: Komdat

Semester: 3

=====APLIKASI KELOLA DATA MAHASISWA=====

1. Tambah Data
2. Tampilkan Data
3. Update Data
4. Delete Data
5. Search Data

6. Keluar

Pilih menu: 3

Masukkan NIM dari data yang ingin diubah: 2021-187

Masukkan Nama Baru: Ahya

Masukkan NIM Baru: 2021-187

Masukkan Matkul Baru: Komdat

Masukkan Semester Baru: 5

Data berhasil diupdate.

=====APLIKASI KELOLA DATA MAHASISWA=====

1. Tambah Data

2. Tampilkan Data

3. Update Data

4. Delete Data

5. Search Data

6. Keluar

Pilih menu: 4

Masukkan NIM data yang ingin dihapus: 2021-189

Data berhasil didelete.

=====APLIKASI KELOLA DATA MAHASISWA=====

1. Tambah Data

2. Tampilkan Data

3. Update Data

4. Delete Data

5. Search Data

6. Keluar

Pilih menu: 5

Masukkan NIM dari data yang ingin dicari: 2021-187

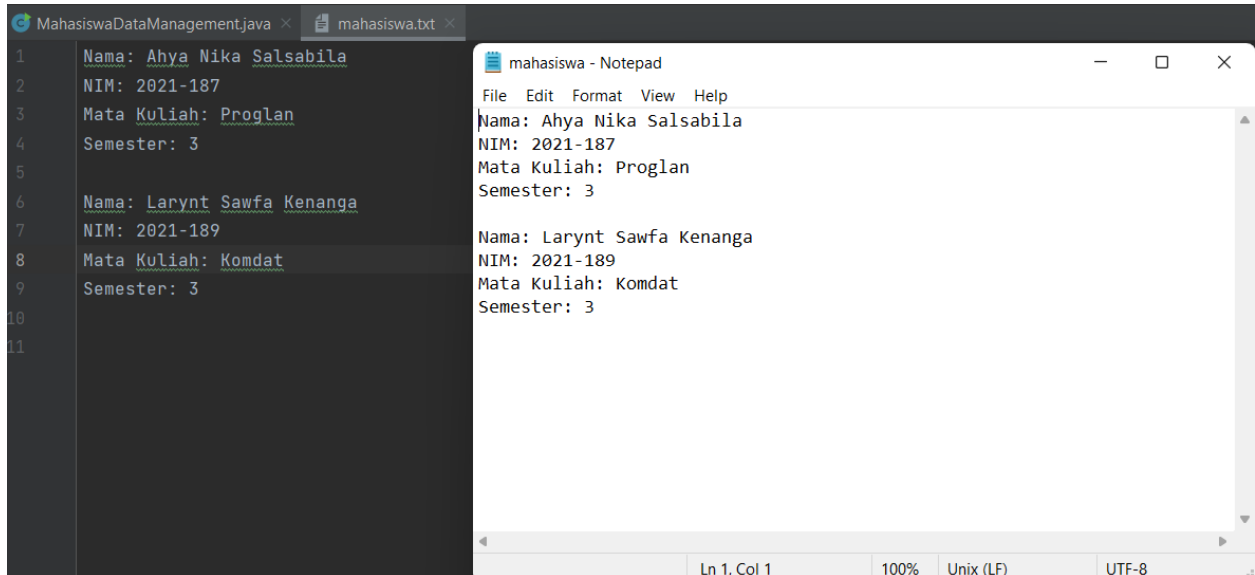
Nama: Ahya

NIM: 2021-187

Mata Kuliah: Komdat

Semester: 5

Dan pastikan data tersimpan dalam file txt:



The screenshot shows a Java IDE with a file named 'MahasiswaDataManagement.java' open. The code in the IDE is as follows:

```
1 Nama: Ahya Nika Salsabila
2 NIM: 2021-187
3 Mata Kuliah: Proglan
4 Semester: 3
5
6 Nama: Larynt Sawfa Kenanga
7 NIM: 2021-189
8 Mata Kuliah: Komdat
9 Semester: 3
10
11
```

Next to the IDE is a Notepad window titled 'mahasiswa - Notepad'. It contains the same text as the code in the IDE, representing the data that has been saved to a file:

```
File Edit Format View Help
Nama: Ahya Nika Salsabila
NIM: 2021-187
Mata Kuliah: Proglan
Semester: 3

Nama: Larynt Sawfa Kenanga
NIM: 2021-189
Mata Kuliah: Komdat
Semester: 3
```

The status bar at the bottom of the Notepad window indicates 'Ln 1, Col 1', '100%', 'Unix (LF)', and 'UTF-8'.

Notes: Jika dikerjakan menggunakan bahasa pemrograman **Python** akan menjadi nilai tambah.

REFERENSI

[Java File I/O Tutorial](#)

[Java FileReader and FileWriter Tutorial](#)

[Java BufferedReader and BufferedWriter Tutorial](#)

[Java File CRUD Operations Example](#)

[Python File Handling Tutorial](#)

[Python File I/O](#)

[Python File Handling: Read, Write, Append](#)

[Python File CRUD Operations Example](#)

https://www.tutorialspoint.com/java/java_files_io.htm

<https://www.geeksforgeeks.org/file-handling-java/>

<https://www.jetbrains.com/idea/documentation/>

https://www.w3schools.com/python/python_file_handling.asp

<https://www.geeksforgeeks.org/file-handling-python/>

<https://www.jetbrains.com/pycharm/documentation/>

KRITERIA & DETAIL PENILAIAN

Kriteria Penilaian		Nilai
Java		55
Latihan	20	
Tugas	30	
Tugas dengan search (Opsional)	5	
Pemahaman		30
Total		85

Kriteria Penilaian		Nilai
Python		70
Latihan	25	
Tugas	40	
Tugas dengan search (Opsional)	5	
Pemahaman		30
Total		100