VERSI 0.1

AGUSTUS 2025

PEMROGRAMAN LANJUT

MODUL 5 - File Handling (Java & Python)

DISUSUN OLEH:

Ir. Wildan Suharso, M.Kom.

Muhammad Ega Faiz Fadlillah

M. Ramadhan Titan Dwi .C

TIM LABORATORIUM INFORMATIKA

UNIVERSITAS MUHAMMADIYAH MALANG

PENDAHULUAN

TUJUAN

1. Mahasiswa mampu memahami konsep dasar File Handling dalam konteks pemrograman.
2. Mahasiswa mampu menyadari pentingnya penyimpanan dan pengelolaan data dalam bentuk file di aplikasi.
3. Mahasiswa mampu melakukan operasi CRUD pada file menggunakan Bahasa Pemrograman Java dan Python.
4. Mahasiswa mampu mengembangkan keterampilan praktis dalam manipulasi data menggunakan operasi CRUD yang melibatkan pengelolaan file.

TARGET MODUL

1. Mahasiswa mampu merancang dan membuat program yang mampu membuat, membaca, memperbarui, dan menghapus data dalam file.
2. Memahami dan dapat mengimplementasikan operasi CRUD dalam file yang efektif dan efisien.
3. Mampu mengatasi masalah umum terkait dengan pengelolaan file seperti error saat membuka file, penutupan file yang tidak benar, serta kesalahan dalam penulisan dan pembacaan data.

PERSIAPAN

1. Java Development Kit
2. Python v3.13 (Optional), untuk instalasi bisa mengikuti panduan ini.
3. Text Editor / IDE (**Preferably** Intellij IDEA, Visual Studio Code, Netbeans, etc).

KEYWORDS

File Handling, Create, Read, Update, Delete

TABLE OF CONTENTS

TEORI

File Handling pada Pyhton

Python adalah bahasa pemrograman yang sangat populer dan mudah dipelajari, terkenal karena sintaksisnya yang sederhana serta mudah dipahami. Bahasa ini mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek, fungsional, dan prosedural. Python menjadi pilihan utama bagi banyak pengembang di seluruh dunia karena pustaka standar yang sangat lengkap, yang memungkinkan pengembang untuk menyelesaikan berbagai macam tugas, seperti analisis data, pengembangan web, kecerdasan buatan, dan pengolahan gambar. Selain itu, Python juga mendukung integrasi dengan berbagai bahasa pemrograman lainnya dan memiliki komunitas yang sangat aktif.

Python mampu menangani berbagai jenis file, baik itu file teks, file biner, maupun format khusus seperti CSV, JSON, dan EML. Program yang ditulis dalam Python dapat melakukan berbagai operasi pada file, seperti membuka file, membaca isi file, menulis data ke dalam file, serta menutup file setelah penggunaannya selesai.

File Handling dan CRUD

File Handling adalah proses pengelolaan data dengan menyimpan, mengakses, memperbarui, dan menghapus data dalam file. File dapat berupa teks, gambar, atau bahkan format data lainnya seperti CSV, JSON, dan XML. Operasi dalam File Handling antara lain :

1. Create = Membuat file baru
2. Read = Membaca isi file
3. Update = Memperbarui data dalam file
4. Delete = Menghapus data dari file

Di dalam materi ini, kita akan mempelajari implementasi File Handling dalam dua bahasa pemrograman, yaitu Java dan Python, serta cara kedua bahasa tersebut menangani file melalui operasi CRUD.

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')

A. CREATE

Dalam pengelolaan file, operasi Create mengacu pada pembuatan file baru di dalam sistem file. Dengan menggunakan proses ini, program atau aplikasi dapat menyimpan data dalam file yang baru dibuat, baik dalam format teks maupun biner. File yang dibuat dapat digunakan untuk berbagai jenis data, seperti teks dalam format txt, data terstruktur dalam format csv, atau data biner untuk gambar, video, dan audio.

Setelah file dibuat, sistem operasi akan memberikan ruang di disk untuk menyimpan file, dan pengguna dapat memilih lokasi penyimpanan file. File baru yang dibuat dapat diakses dan diubah sesuai kebutuhan aplikasi. Dalam beberapa kasus, jika file dengan nama yang sama sudah ada, sistem mungkin memberi tahu Anda atau bahkan menimpa file lama. Ini bergantung pada mode akses yang digunakan, apakah itu tulis atau tambah. Aplikasi dapat menyimpan data secara permanen, memungkinkan pengolahan lanjutan, atau berbagi informasi antar aplikasi, sambil mengelola file dengan format yang sesuai dengan jenis data yang akan disimpan dengan operasi Create.

Contoh Program Java :

```java
import java.io.FileWriter;
import java.io.IOException;

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

Pada kode Java ini, FileWriter digunakan untuk membuat file baru bernama data_java.txt dan menulis string "Nama, NIM, Semester\n" ke dalamnya. Setelah itu, file ditutup menggunakan fileWriter.close(), dan pesan "File berhasil dibuat." ditampilkan di konsol.

Contoh Program Python :

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

Pada kode Python ini, fungsi open() digunakan untuk membuka file baru dengan nama data_python.txt dalam mode tulis 'w'. Kemudian, program menulis string "Nama, NIM, Semester\n" ke dalam file tersebut dan menutupnya dengan file.close(). Setelah itu, Python juga menampilkan pesan "File berhasil dibuat.".

B. READ

Operasi Read adalah proses mengambil atau membaca isi file yang sudah ada di dalam sistem file. Proses ini memungkinkan program untuk mengakses data yang telah disimpan sebelumnya dalam file dan menggunakannya untuk berbagai tujuan. Bergantung pada jenis file yang dibuka dan aplikasi yang digunakan, program dapat membaca konten file secara bertahap atau sekaligus, tergantung pada metode yang digunakan. File yang dibaca dapat berisi data dalam berbagai format, seperti teks, angka, atau data biner.

Operasi Read memungkinkan data yang dibaca diproses dalam program. Sementara program biasanya membaca baris demi baris atau seluruh konten file sekaligus dalam file teks, data biasanya dibaca dalam bentuk byte dalam file biner. Operasi Read adalah tahap penting dalam pengelolaan data, yang memungkinkan aplikasi untuk melakukan berbagai tugas, seperti analisis data, pencarian informasi, atau pengolahan konten file. Setelah data dibaca, file biasanya ditutup untuk membebaskan sumber daya yang digunakan, tetapi beberapa bahasa pemrograman, seperti Python, mendukung penutupan otomatis file ketika menggunakan struktur with.

Contoh Program Java :

```java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class ReadFileExample {
    public static void main(String[] args) {
        String fileName = "data_java.txt";
        try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {
            String line;
            int lineNumber = 1;
            while ((line = reader.readLine()) != null) {
                System.out.println("Baris " + lineNumber + ": " + line);
                lineNumber++;
            }
        } catch (IOException e) {
            System.out.println("Gagal membaca file: " + e.getMessage());
        }
    }
}
```

Pada contoh program Java ini, kita menggunakan BufferedReader dan FileReader untuk membuka dan membaca file data_java.txt. Program membaca file baris demi baris menggunakan readLine(). Setiap baris yang dibaca dicetak dengan nomor baris untuk memudahkan pemantauan progres pembacaan file. Program ini juga menangani pengecualian IOException yang dapat terjadi saat file tidak ditemukan atau ada kesalahan dalam membaca file.

Contoh Program Python :

```python
def read_file(file_name):
    try:
        with open(file_name, "r") as file:
            content = file.readlines()
            for i, line in enumerate(content, 1):
                print(f"Baris {i}: {line.strip()}")
    except FileNotFoundError:
        print("File tidak ditemukan!")

# Memanggil fungsi untuk membaca file
read_file("data_python.txt")
```

Pada contoh program Python ini, kita menggunakan with open() untuk membuka file data_python.txt dalam mode baca ('r'). Fungsi readlines() digunakan untuk membaca seluruh baris dalam file dan menyimpannya dalam bentuk list. Program kemudian

mencetak setiap baris beserta nomor urutannya dengan menggunakan enumerate(), yang juga menghilangkan karakter newline menggunakan .strip(). Jika file tidak ditemukan, program menangani pengecualian FileNotFoundError dan menampilkan pesan kesalahan.

C. UPDATE

Operasi Update mengacu pada proses yang digunakan untuk memperbarui atau mengubah data yang sudah ada dalam suatu file. Dalam kasus ini, operasi update dilakukan pada file teks atau file biner untuk memodifikasi bagian tertentu dari data yang disimpan sebelumnya. Proses ini memungkinkan pengguna untuk mengubah informasi dalam file sesuai dengan kebutuhan aplikasi, seperti mengganti kata-kata tertentu atau menambahkan data baru pada bagian akhir file.

Proses update file dapat dilakukan dengan membuka file dalam mode yang memungkinkan perubahan, seperti mode ‘r+’ untuk file teks atau mode ‘wb’ untuk file biner. Ketika file dibuka, program dapat membaca data yang ada, memodifikasi bagian yang diperlukan, dan kemudian menyimpan perubahan tersebut ke dalam file. Jika file yang ingin diubah sangat besar, teknik seperti penulisan data baru sementara dan menggantikan file asli dapat digunakan untuk menghindari kehilangan data.

Contoh Program Java :

```java
import java.io.*;

class UpdateFile {
    public static void main(String[] args) {
        try {
            // Membuka file dalam mode append untuk menambahkan data
            FileWriter fileWriter = new FileWriter("data_file.txt", true);
            // Menulis data baru ke file
            fileWriter.write("Ken Aryo, 2004, 8\n");
            fileWriter.close();
            // Menampilkan pesan konfirmasi
            System.out.println("Data berhasil ditambahkan.");
        } catch (IOException e) {
            System.out.println("Terjadi kesalahan.");
        }
    }
}
```

Pada contoh Java, program membuka file "data_file.txt" dalam mode append (true pada konstruktor FileWriter). Program menambahkan data baru ("Ken Aryo, 2004, 8") ke dalam file dan kemudian menutup file dengan fileWriter.close(). Jika terjadi kesalahan saat menulis data, program akan menangani pengecualian IOException.

Contoh Program Python :

```python
# Membuka file dalam mode append
with open("data_file_python.txt", "a") as file:
    # Menulis data baru ke file
    file.write("Ken Aryo, 2004, 8\n")
# Menampilkan pesan konfirmasi
print("Data berhasil ditambahkan.")
```

Di contoh Python, file "data_file_python.txt" dibuka dalam mode append menggunakan perintah with open(). Program kemudian menulis data baru ("Ken Aryo, 2004, 8") ke dalam file, dan secara otomatis menutup file setelah operasi selesai. Program ini juga menampilkan pesan konfirmasi bahwa data telah berhasil ditambahkan.

D. DELETE

Operasi *Delete* digunakan untuk menghapus file atau bagian dari file yang sudah tidak dibutuhkan lagi. Dalam sistem manajemen file, penghapusan ini tidak hanya menghilangkan file tersebut, tetapi juga mengembalikan ruang penyimpanan yang sebelumnya digunakan oleh file tersebut. Proses penghapusan membantu menjaga sistem agar tetap terorganisir, menghindari penumpukan file yang tidak relevan yang dapat mengganggu kinerja sistem.

Ketika sebuah file dihapus, beberapa sistem operasi langsung menghapus file tersebut dari direktori, sementara sistem lain mungkin menyimpannya terlebih dahulu di tempat sampah (recycle bin), memberikan kesempatan untuk pemulihan jika file tersebut terhapus secara tidak sengaja. Untuk penghapusan permanen, teknik-teknik khusus dapat digunakan untuk memastikan data tersebut benar-benar hilang, terutama jika file tersebut mengandung informasi sensitif.

Contoh Program Java :

```java
import java.io.*;

class DeleteFile {
    public static void main(String[] args) {
        try {
            // File yang akan dibaca dan dihapus
            File inputFile = new File("data_java.txt");
            // File sementara untuk menampung data
            File tempFile = new File("temp.txt");

            BufferedReader bufferedReader = new BufferedReader(new FileReader(inputFile));
            PrintWriter printWriter = new PrintWriter(new FileWriter(tempFile));

            String lineToRemove = " Ken Aryo, 2004, 3";
            String currentLine;
            while ((currentLine = bufferedReader.readLine()) != null) {
                if (!currentLine.equals(lineToRemove)) {
                    printWriter.println(currentLine);
                }
            }
            printWriter.close();
            bufferedReader.close();

            // Menghapus file asli dan mengganti dengan file sementara
            if (inputFile.delete()) {
                tempFile.renameTo(inputFile);
                System.out.println("Data berhasil dihapus.");
            }
        } catch (IOException e) {
            System.out.println("Exception Occurred");
        }
    }
}
```

Pada program ini, file "data_java.txt" dibaca menggunakan BufferedReader dan setiap baris yang cocok dengan data yang akan dihapus ("Ken Aryo, 2004, 3") diabaikan. Baris lain yang tidak cocok ditulis ke file sementara ("temp.txt"). Setelah itu, file asli dihapus menggunakan inputFile.delete(), dan file sementara diganti namanya menjadi nama file asli menggunakan tempFile.renameTo(inputFile).

Contoh Program Python :

```python
line_to_remove = "Ken Aryo, 2004, 3"
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.")
```

Pada contoh Python, file "data_python.txt" dibaca menggunakan metode readlines() yang menyimpan semua baris dalam file. Program kemudian menulis kembali ke file tersebut, hanya menyertakan baris yang tidak sesuai dengan data yang ingin dihapus ("Ken Aryo, 2004, 3"). Setelah selesai, program menampilkan pesan konfirmasi bahwa data telah berhasil dihapus.

E. IMAGE PROCESSING

Pemrosesan gambar di Java dan Python memungkinkan kita untuk memanipulasi gambar dengan cara membaca, menulis, atau mengubah format gambar. Dalam contoh kali ini, kita akan belajar bagaimana cara membaca gambar, mengubah format gambar, dan menyimpannya ke file baru.

1. Java

a. Membaca dan Mengubah Format Gambar

Kita akan menggunakan ImageIO untuk membaca gambar dan mengubah formatnya dari JPG menjadi PNG. Proses ini berguna jika kita ingin mengganti format gambar tanpa harus mengubah isi gambar itu sendiri.

```java
import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

public class GambarFormatConverter {
    public static void main(String[] args) {
        try {
            // Membaca gambar dalam format JPG
            File inputFile = new File("gambar_asli.jpg");
            BufferedImage image = ImageIO.read(inputFile);
            System.out.println("Gambar berhasil dibaca!");

            // Mengubah format gambar dan menyimpannya dalam format PNG
            File outputFile = new File("gambar_baru.png");
            ImageIO.write(image, "png", outputFile);
            System.out.println("Gambar berhasil disimpan dalam format PNG!");

        } catch (IOException e) {
            System.out.println("Terjadi kesalahan: " + e.getMessage());
        }
    }
}
```

Dalam program ini kita menggunakan ImageIO.read(inputFile) untuk membaca gambar dari file. Gambar yang dibaca dalam format JPG akan dimuat ke dalam objek BufferedImage. Kemudian menyimpan gambar ke dalam format PNG menggunakan ImageIO.write(image, "png", outputFile). Ini adalah contoh pengubahan format gambar tanpa memodifikasi gambar itu sendiri.

b. Menyalin atau Memindahkan File Gambar

Pada contoh kali ini, kita akan menggunakan Java untuk menyalin dan memindahkan gambar dengan cara yang berbeda. Alih-alih menggunakan FileUtils dari Apache Commons IO, kita akan menggunakan Java java.nio.file.Files yang lebih sederhana dan sudah tersedia di dalam Java SDK.

```java
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.StandardCopyOption;

public class GambarFileHandling {
    public static void main(String[] args) {
        try {
            // Menyalin gambar
            File sourceFile = new File("original_image.jpg");
            File destFile = new File("copied_image.jpg");
            Files.copy(sourceFile.toPath(), destFile.toPath(), StandardCopyOption.REPLACE_EXISTING);
            System.out.println("Gambar berhasil disalin.");

            // Memindahkan gambar
            File movedFile = new File("moved_image.jpg");
            Files.move(sourceFile.toPath(), movedFile.toPath(), StandardCopyOption.REPLACE_EXISTING);
            System.out.println("Gambar berhasil dipindahkan.");

        } catch (IOException e) {
            System.out.println("Terjadi kesalahan: " + e.getMessage());
        }
    }
}
```

Program ini menggunakan kelas java.nio.file.Files yang merupakan bagian dari Java NIO untuk menyalin dan memindahkan file. Pada contoh ini, kita pertama-tama menyalin gambar dari file original_image.jpg ke file baru bernama copied_image.jpg menggunakan metode Files.copy(). Setelah itu, gambar yang sama dipindahkan ke file baru dengan nama moved_image.jpg menggunakan Files.move().

c. Manipulasi Gambar

Pada contoh kali ini, kita akan menggunakan pustaka Java AWT untuk melakukan manipulasi gambar, seperti mengubah ukuran gambar.

```java
import java.awt.Image;
import java.awt.Toolkit;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;

public class ResizeImage {
    public static void main(String[] args) {
        try {
            // Membaca gambar
            File inputFile = new File("example.jpg");
            Image image = ImageIO.read(inputFile);

            // Mengubah ukuran gambar
            Image resizedImage = image.getScaledInstance(100, 100, Image.SCALE_SMOOTH);

            // Menyimpan gambar yang sudah diubah ukuran
            File outputFile = new File("resized_example.jpg");
            ImageIO.write((java.awt.image.BufferedImage) resizedImage, "jpg", outputFile);

            System.out.println("Gambar berhasil diubah ukurannya.");
        } catch (IOException e) {
            System.out.println("Terjadi kesalahan: " + e.getMessage());
        }
    }
}
```

Program ini menggunakan Java AWT untuk memanipulasi gambar. Pertama, kita membaca gambar menggunakan ImageIO.read(), yang mengembalikan objek Image.

Kemudian, kita mengubah ukuran gambar tersebut menggunakan metode getScaledInstance(), yang memungkinkan kita untuk menentukan lebar dan tinggi gambar yang baru, dalam hal ini 100x100 piksel.

Setelah gambar diubah ukurannya, gambar tersebut disimpan menggunakan ImageIO.write(). Gambar hasil resize akan disimpan dengan nama resized_example.jpg.

2. Python

a. Membaca dan Menulis File Gambar

Untuk bekerja dengan file gambar di Python, kita bisa menggunakan pustaka Pillow. Pustaka ini memungkinkan kita untuk membuka, mengubah, dan menyimpan file gambar dalam berbagai format.

Pip install Pillow

```python
from PIL import Image

# Membaca gambar
image = Image.open("example.jpg")
print(f"Format: {image.format}, Ukuran: {image.size}, Mode: {image.mode}")

# Menyimpan gambar dengan nama baru
image.save("example_copy.jpg")
print("Gambar berhasil disalin.")
```

Program ini menggunakan pustaka Pillow untuk membuka gambar example.jpg, kemudian menampilkan informasi seperti format gambar, ukuran, dan mode gambar. Setelah itu, gambar disalin dengan nama baru menjadi example_copy.jpg menggunakan metode save() dari objek gambar.

b. Menyalin atau Memindahkan File Gambar

Untuk menyalin atau memindahkan file gambar di Python, kita bisa menggunakan pustaka shutil yang menyediakan berbagai metode untuk bekerja dengan file.

```python
import shutil

# Menyalin gambar
shutil.copy("example.jpg", "copy_example.jpg")
print("Gambar berhasil disalin.")

# Memindahkan gambar
shutil.move("copy_example.jpg", "moved_example.jpg")
print("Gambar berhasil dipindahkan.")
```

Pada bagian ini, kita menggunakan pustaka shutil untuk menyalin gambar example.jpg ke file baru copy_example.jpg dengan menggunakan metode copy(). Kemudian, gambar yang telah disalin dipindahkan dari copy_example.jpg ke moved_example.jpg menggunakan metode move().

c. Manipulasi Gambar

Untuk manipulasi gambar, kita dapat menggunakan pustaka **Pillow (PIL)**. Dalam contoh kali ini, kita akan melakukan dua operasi: mengubah ukuran gambar dan memotong bagian tertentu dari gambar.

```python
from PIL import Image

# Membuka gambar
image = Image.open("example.jpg")

# Mengubah ukuran gambar
resized_image = image.resize((100, 100))
resized_image.save("example_resized.jpg")
print("Gambar berhasil diubah ukurannya.")

# Memotong gambar
cropped_image = image.crop((50, 50, 200, 200))
cropped_image.save("example_cropped.jpg")
print("Gambar berhasil dipotong.")
```

Program ini menggunakan pustaka Pillow untuk membuka gambar example.jpg. Kemudian, gambar tersebut diubah ukurannya menjadi 100x100 piksel menggunakan metode resize(). Hasilnya disimpan sebagai example_resized.jpg.

Selanjutnya, gambar dipotong menggunakan koordinat (50, 50, 200, 200) yang menunjukkan area gambar yang akan diambil. Area ini diwakili oleh koordinat kiri atas (50, 50) dan koordinat kanan bawah (200, 200). Hasil potongan gambar disimpan sebagai example_cropped.jpg.

F. FILE HANDLING MENGGUNAKAN XLSX / DOCX

1. **Java**

   Untuk bekerja dengan file .docx di Java, kita memerlukan pustaka Apache POI. Dengan pustaka ini, kita bisa membuka, membaca, dan menulis dokumen Word dalam format .docx.

```xml
<dependency>
<groupId>org.apache.poi</groupId>
<artifactId>poi-ooxml</artifactId>
<version>5.2.3</version>
</dependency>
```

a. .docx

```java
import org.apache.poi.xwpf.usermodel.XWPFDocument;
import org.apache.poi.xwpf.usermodel.XWPFParagraph;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class DocxFileHandling {
    public static void main(String[] args) {
        try {
            // Membaca file .docx
            FileInputStream fis = new FileInputStream("example.docx");
            XWPFDocument document = new XWPFDocument(fis);

            // Membaca teks dari paragraf pertama
            XWPFParagraph paragraph = document.getParagraphs().get(0);
            System.out.println("Teks pertama dalam file .docx: " + paragraph.getText());

            // Menulis ke file .docx
            FileOutputStream fos = new FileOutputStream("example_copy.docx");
            document.write(fos);
            fos.close();
            System.out.println("File .docx berhasil disalin.");
            fis.close();
        } catch (IOException e) {
            System.out.println("Terjadi kesalahan: " + e.getMessage());
        }
    }
}
```

Program ini menggunakan pustaka Apache POI untuk memanipulasi file .docx. Pertama, file example.docx dibuka menggunakan FileInputStream, dan kemudian sebuah objek XWPFDocument digunakan untuk membaca file tersebut.

Setelah itu, teks dari paragraf pertama diambil menggunakan getParagraphs() yang mengembalikan daftar semua paragraf dalam dokumen. Kemudian, teks dari paragraf pertama dicetak di konsol.

Pada bagian kedua, file .docx disalin menggunakan FileOutputStream. Setelah file ditulis, program menutup stream dan mencetak pesan bahwa file berhasil disalin.

b. .xlsx

Untuk bekerja dengan file .xlsx, kita dapat menggunakan pustaka Apache POI, yang menyediakan kelas seperti XSSFWorkbook untuk membuka dan memanipulasi file Excel. Program berikut ini akan membuka file .xlsx, membaca nilai dari sebuah sel, dan menyimpan file yang telah dimodifikasi.

```java
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class XlsxFileHandling {
    public static void main(String[] args) {
        try {
            // Membaca file .xlsx
            FileInputStream fis = new FileInputStream("example.xlsx");
            XSSFWorkbook workbook = new XSSFWorkbook(fis);
            XSSFSheet sheet = workbook.getSheetAt(0);

            // Menampilkan nilai pada sel pertama (baris 0, kolom 0)
            System.out.println("Nilai di sel pertama: " + sheet.getRow(0).getCell(0).getStringCellValue());

            // Menulis ke file .xlsx
            FileOutputStream fos = new FileOutputStream("example_copy.xlsx");
            workbook.write(fos);
            System.out.println("File .xlsx berhasil disalin.");
            fos.close();
            fis.close();
        } catch (IOException e) {
            System.out.println("Terjadi kesalahan: " + e.getMessage());
        }
    }
}
```

Program ini menggunakan pustaka Apache POI untuk menangani file .xlsx. Pertama, kita membuka file example.xlsx menggunakan FileInputStream, lalu membuat objek XSSFWorkbook untuk mewakili workbook. Kita mengakses lembar kerja pertama dengan getSheetAt(0). Setelah itu, program membaca nilai dari sel pertama (baris 0, kolom 0) menggunakan metode getRow(0).getCell(0).getStringCellValue() dan mencetak nilai tersebut ke konsol. Di bagian selanjutnya, kita menulis ulang workbook ke dalam file baru menggunakan FileOutputStream. Program ini menyimpan file sebagai example_copy.xlsx.

2. Python
    a. .docx

    Untuk bekerja dengan file .docx di Python, kita bisa menggunakan pustaka python-docx. Pustaka ini memungkinkan kita untuk mengelola elemen-elemen dokumen seperti paragraf, teks, dan lainnya.

    pip install python-docx

```python
from docx import Document

# Membaca file .docx
doc = Document("example.docx")

# Menampilkan teks dari paragraf pertama
print(f"Teks pertama dalam file .docx: {doc.paragraphs[0].text}")

# Menulis ke file .docx
doc.add_paragraph("Ini adalah teks baru.")
doc.save("example_copy.docx")
print("File .docx berhasil disalin.")
```

Program ini menggunakan pustaka python-docx untuk membuka file example.docx. Setelah file dibuka, kita dapat mengakses teks dari paragraf pertama dengan menggunakan doc.paragraphs[0].text. Teks dari paragraf pertama kemudian ditampilkan di konsol. Selanjutnya, kita menambahkan teks baru ke dalam dokumen menggunakan metode add_paragraph(). Setelah itu, file yang telah dimodifikasi disimpan dengan nama baru example_copy.docx menggunakan metode save().

b. .xlsx

Untuk bekerja dengan file .xlsx di Python, kita menggunakan pustaka openpyxl. Pustaka ini memungkinkan kita untuk membuka, membaca, dan memodifikasi file Excel.

pip install openpyxl

```python
import openpyxl

# Membaca file .xlsx
wb = openpyxl.load_workbook("example.xlsx")
sheet = wb.active

# Menampilkan nilai pada sel pertama (baris 1, kolom 1)
print(f"Nilai di sel pertama: {sheet.cell(row=1, column=1).value}")

# Menulis ke file .xlsx
sheet.cell(row=2, column=1, value="Teks baru")
wb.save("example_copy.xlsx")
print("File .xlsx berhasil disalin.")
```

Program ini menggunakan pustaka openpyxl untuk memanipulasi file .xlsx. Pertama, kita membuka file example.xlsx menggunakan metode load_workbook(), yang memungkinkan kita untuk mengakses workbook dan

memilih lembar kerja yang aktif dengan wb.active. Selanjutnya, kita membaca nilai dari sel pertama (baris 1, kolom 1) menggunakan sheet.cell(row=1, column=1).value, dan menampilkan nilai tersebut ke konsol. Di bagian berikutnya, program menulis teks baru ke sel (baris 2, kolom 1) menggunakan sheet.cell(row=2, column=1, value="Teks baru"), dan menyimpan file dengan nama baru example_copy.xlsx menggunakan metode save().

REFRENSI

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/
https://www.baeldung.com/java-images
https://www.geeksforgeeks.org/working-images-python/

CODELAB

Codelab 1

Buatlah sebuah program di Java yang dapat menerima input nama, semester, dan mata kuliah dari pengguna. Inputkan data tidak diperbolehkan nama yang sama. Jika selesai menginput, data tersebut akan tersimpan ke dalam file dengan nama data_mahasiswa.xlsx.

Contoh Output :

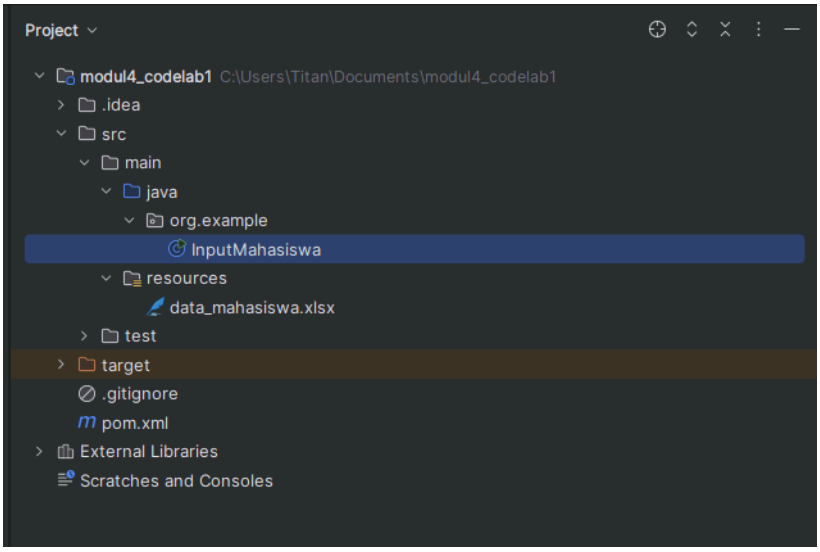
example.java

```
Masukkan data mahasiswa. Ketik 'selesai' pada nama untuk mengakhiri
Masukkan Nama: Ken
Masukkan Semester: 5
Masukkan Mata Kuliah: Pemrograman Mobile
Data berhasil disimpan ke dalam file data_mahasiswa.xlsx !

Masukkan Nama: Ega
Masukkan Semester: 5
Masukkan Mata Kuliah: Pemrograman Web
Data berhasil disimpan ke dalam file data_mahasiswa.xlsx !

Masukkan Nama: Ega
Nama sudah ada, masukkan nama yang berbeda !
Masukkan Nama: selesai
Terima kasih !
```

Contoh letak file data_mahasiswa.xlsx :



Contoh Isi file data_mahasiswa.xlsx :

Nama	Semester	Mata Kuliah
Ken	5	Pemrograman Mobile
Ega	5	Pemrograman Web

Tugas

Tugas 1

Buatlah sebuah program CRUD yang mengelola data buku di perpustakaan. Program harus dapat:

Note : Jika dikerjakan dengan Python, akan mendapatkan nilai tambah

1. Menambahkan buku baru ke dalam file .json.
2. Menampilkan data buku yang ada.
3. Mengupdate data buku (misalnya mengganti judul atau pengarang).
4. Menghapus buku yang sudah tidak diperlukan.

Contoh Ouput :

```
--- Menu Perpustakaan ---
1. Menambahkan Buku
2. Menampilkan Buku
3. Mengupdate Buku
4. Menghapus Buku
5. Keluar
Pilih menu (1/2/3/4/5): 1

Menambahkan Buku Baru
Masukkan judul buku: sore pacar masa depan
Masukkan pengarang buku: wira
Masukkan tahun terbit buku: 2035
```

```
--- Menu Perpustakaan ---
1. Menambahkan Buku
2. Menampilkan Buku
3. Mengupdate Buku
4. Menghapus Buku
5. Keluar
Pilih menu (1/2/3/4/5): 2

Daftar Buku:
1. sore pacar masa depan oleh wira (2025)

--- Menu Perpustakaan ---
1. Menambahkan Buku
2. Menampilkan Buku
3. Mengupdate Buku
4. Menghapus Buku
5. Keluar
Pilih menu (1/2/3/4/5): 3

Daftar Buku:
1. sore pacar masa depan oleh wira (2025)

Masukkan nomor buku yang ingin diupdate: 1

Update Buku:
Masukkan judul baru (kosongkan untuk mempertahankan 'sore pacar masa depan'):
Masukkan pengarang baru (kosongkan untuk mempertahankan 'wira'): ken
Masukkan tahun terbit baru (kosongkan untuk mempertahankan '2025'): 2024
Buku berhasil diupdate.
```

```
--- Menu Perpustakaan ---
1. Menambahkan Buku
2. Menampilkan Buku
3. Mengupdate Buku
4. Menghapus Buku
5. Keluar
Pilih menu (1/2/3/4/5): 4

Daftar Buku:
1. sore pacar masa depan oleh ken (2024)

Masukkan nomor buku yang ingin dihapus: 1
Buku berhasil dihapus.
```

Contoh isi file .json :

books.json > ...

```json
[
    {
        "title": "harry proter",
        "author": "ken",
        "year": "2030"
    },
    {
        "title": "sore pacar masa depan",
        "author": "wira",
        "year": "2035"
    },
    {
        "title": "si pembalas",
        "author": "titan",
        "year": "2045"
    }
]
```

Tugas 2

Buatlah program dalam Java yang melakukan manipulasi gambar dan penyimpanan file, termasuk membuka, mengubah ukuran gambar, memindahkan, menyalin, serta mengubah format gambar sesuai dengan instruksi berikut :

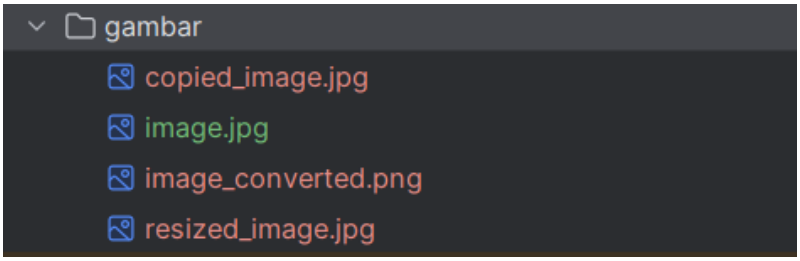
Note : Jika dikerjakan dengan Python, akan mendapatkan nilai tambah

1. Program harus membuka gambar yang disimpan di file image.jpg.
2. Program akan mengubah ukuran gambar tersebut menjadi 200x200 piksel.
3. Setelah itu, simpan gambar yang telah diubah ukurannya dengan nama resized_image.jpg.
4. Setelah gambar diubah ukurannya, buat salinan gambar tersebut ke file baru dengan nama copied_image.jpg.
5. Salinan gambar ini harus dipindahkan ke folder yang berbeda dengan menggunakan teknik yang sesuai (misalnya, menggunakan metode copy atau move).
6. Ubah format gambar yang telah disalin dari JPG ke PNG dan simpan hasilnya dengan nama image_converted.png.

Contoh Output :

```
Gambar berhasil disimpan sebagai resized_image.jpg
Salinan gambar berhasil dibuat dengan nama copied_image.jpg
Format gambar berhasil diubah menjadi PNG dan disimpan dengan nama image_converted.png
```

Letak file image :



KRITERIA & DETAIL PENILAIAN

KRITERIA PENILAIAN	POIN (TOTAL 100%)
CODELAB 1	**Total 25%**
Kesesuaian Output	10%
Pemahaman Materi	15%
TUGAS 1	**Total 35%**
Kesesuaian Output	10%
Pemahaman Materi	15%
Penggunaan Python	10%
TUGAS 2	**Total 40%**
Kesesuaian Output	15%
Pemahaman Materi	15%
Penggunaan Python	10%