

## LAPORAN HASIL PRAKTIKUM

### Aplikasi Kompresi Data Berbasis Web Menggunakan Flask dan Tailwind CSS

#### 1 . Tujuan Praktikum

Praktikum ini bertujuan untuk membangun aplikasi kompresi data berbasis web yang mampu mengompresi berbagai jenis file (gambar, dokumen, dll) menggunakan Python Flask sebagai backend, serta HTML dan Tailwind CSS sebagai frontend. Aplikasi ini juga menampilkan hasil kompresi berupa ukuran sebelum dan sesudah kompresi.

#### 2. Alat dan Bahan

- Bahasa Pemrograman: Python 3.x
- Framework Backend: Flask
- Frontend: HTML, Tailwind CSS
- Library Kompresi: Pillow (untuk gambar), PyPDF2 (untuk PDF), dll (jika ada)
- Editor: VS Code / PyCharm
- Browser: Chrome / Firefox

#### 3. Desain dan Alur Aplikasi

Pengguna memilih file yang ingin dikompres → file dikirim ke server Flask → proses kompresi dijalankan → hasil ditampilkan dalam bentuk preview nama file, ukuran sebelum dan sesudah → pengguna dapat menyimpan atau membatalkan penyimpanan file hasil kompresi.



#### 4. Implementasi Program

```
> __pycache__  
> env  
└── templates  
    ├── index.html  
    ├── uploads  
    ├── app.py  
    ├── backup.py  
    └── kompres.py
```

##### a. Struktur Folder Proyek

- `app.py`: file utama untuk menjalankan server Flask.
- `kompres.py`: berisi fungsi-fungsi untuk mengompresi file.
- `templates/index.html`: tampilan antarmuka web pengguna.
- `uploads/`: direktori untuk menyimpan file yang diunggah.
- `env/`: virtual environment project.
- `__pycache__/`: folder otomatis dari Python,

##### b. File `app.py` — Logika Utama Aplikasi

File ini adalah jantung aplikasi web yang mengatur:

- Routing halaman,
- Upload file dari pengguna,
- Memanggil fungsi kompresi berdasarkan file,
- Menampilkan ukuran sebelum dan sesudah kompresi,
- Menyediakan link untuk mengunduh hasil kompresi.

```

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        file = request.files.get('file')
        kualitas = request.form.get('kualitas')

        if not file or not kualitas:
            return render_template('index.html', error="File dan kualitas wajib diisi.")

        # Simpan file sementara
        original_path = os.path.join(UPLOAD_FOLDER, file.filename)
        file.save(original_path)

        # Kompres file
        hasil_path = kompres_file(original_path, kualitas)
        if not hasil_path:
            return render_template('index.html', error="Kompresi gagal atau format tidak didukung.")

        # Hitung ukuran sebelum dan sesudah
        before = get_file_size(original_path)
        after = get_file_size(hasil_path)

        return render_template('index.html',
                               nama_file=file.filename,
                               before_size=f"{before:.2f} KB",
                               after_size=f"{after:.2f} KB",
                               download_url=url_for('download_file', filename=os.path.basename(hasil_path)))
    return render_template('index.html')

```

### Penjelasan:

- `request.files.get('file')`: mengambil file dari form HTML.
- `kompres_file(...)`: fungsi utama untuk melakukan proses kompresi.
- Ukuran file dihitung sebelum dan sesudah menggunakan `get_file_size`.
- Hasil dikirim kembali ke halaman utama (`index.html`) untuk ditampilkan.

### c. File kompres.py — Fungsi Inti Kompresi

File ini berisi seluruh logika kompresi untuk berbagai format file: .jpg, .png, .pdf, .docx, dan .pptx.

```
def kompres_file(path, kualitas):
    ext = os.path.splitext(path)[1].lower()
    try:
        if ext in ['.jpg', '.jpeg']:
            kualitas_map = {"Tinggi": 80, "Sedang": 50, "Rendah": 30}
            img = Image.open(path)
            output_path = os.path.join(tempfile.gettempdir(), "compressed_" + os.path.basename(path))
            img.convert("RGB").save(output_path, format='JPEG', quality=kualitas_map[kualitas], optimize=True)
    return output_path
```

### Penjelasan:

- Untuk file .jpg, kualitas ditentukan pengguna (80, 50, 30).
- Gambar dikonversi ke RGB dan disimpan ulang dengan kualitas lebih rendah.

```
def compress_office_file(original_path, kualitas):
    kualitas_map = {"Tinggi": 80, "Sedang": 60, "Rendah": 40}

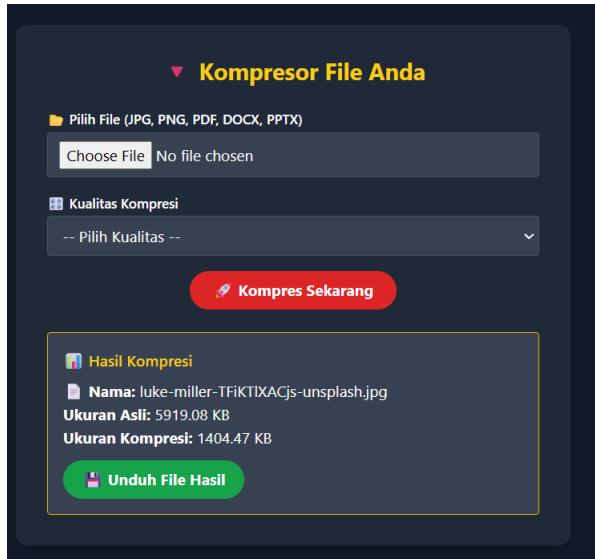
    # Ekstrak isi file Office
    with zipfile.ZipFile(original_path, 'r') as zip_ref:
        zip_ref.extractall(extract_dir)

    # Cari gambar dalam folder /media
    for img_file in os.listdir(media_dir):
        img_path = os.path.join(media_dir, img_file)
        img = Image.open(img_path)
        img.convert("RGB").save(img_path, optimize=True, quality=kualitas_map[kualitas])
```

### Penjelasan:

- File Office (.docx, .pptx) sebenarnya adalah file ZIP → diekstrak.
- Gambar-gambar dalam folder media/ dikompres.
- Lalu file disusun kembali menjadi dokumen utuh hasil kompresi.

## 5. Hasil dan Tampilan Aplikasi



## Panduan Menggunakan Aplikasi

### Deskripsi singkat aplikasi

Aplikasi ini dibuat untuk mengompres ukuran file dengan mudah melalui tampilan web. File yang didukung meliputi gambar (.jpg, .png), dokumen (.pdf, .docx, .pptx), dan file lainnya. Aplikasi dibangun menggunakan Python Flask sebagai backend dan Tailwind CSS untuk tampilan antarmuka.

### Fitur Utama

- Upload file dari komputer
- Pilih kualitas kompresi (Tinggi, Sedang, Rendah)
- Proses kompresi otomatis di server
- Menampilkan ukuran sebelum dan sesudah
- Opsi untuk mengunduh hasil kompresi

### Cara Menggunakan Aplikasi

Langkah 1: Buka Aplikasi

Buka terminal dan jalankan perintah berikut :

```
PS C:\Users\user\Desktop\flask> python app.py
```

Langkah 2 : Buka Browser

Buka browser dan jalankan aplikasi di alamat ketik <http://localhost:5000>

Setelah itu akan masuk ke tampilan awal :

▼ Kompresor File Anda

Pilih File (JPG, PNG, PDF, DOCX, PPTX)

Choose File No file chosen

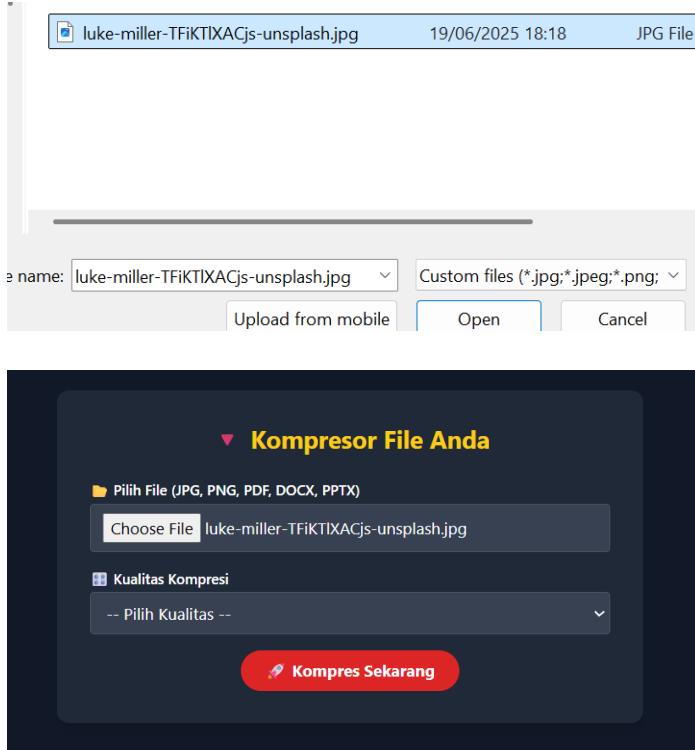
Kualitas Kompresi

-- Pilih Kualitas --

**Kompres Sekarang**

### Langkah 3 : Pilih File

Klik choose file untuk memilih file yang akan dikompres :



### Langkah 4 : pilih kualitas kompresi

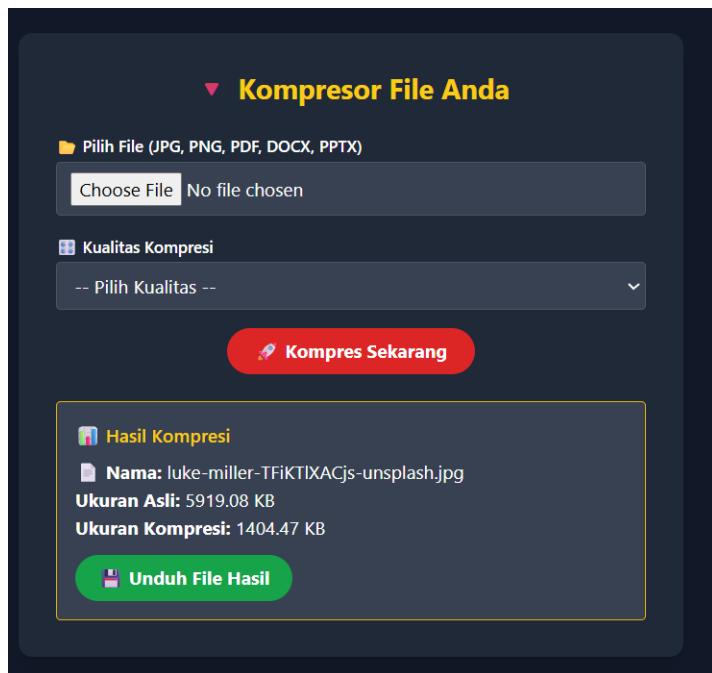
Pilih salah satu dari 3 opsi:

- **Tinggi:** Kualitas bagus, ukuran besar
- **Sedang:** Keseimbangan
- **Rendah:** Ukuran paling kecil, kualitas menurun



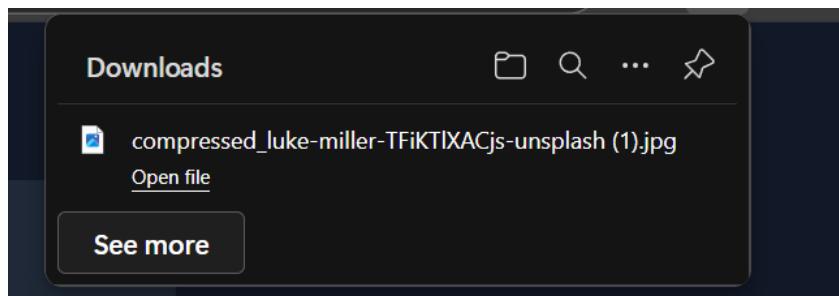
Langkah 4 : klik tombol kompres sekarang

Klik tombol "Kompres Sekarang" untuk memulai proses kompresi, dan hasil ukuran sebelum dan sesudah kompresi akan muncul



Langkah 5 : Unduh file hasil kompresi

Setelah itu klik unduh file hasil maka file hasil kompresi akan terdownload



## Format File yang Didukung dan Metode Kompresi

Aplikasi ini mendukung beberapa jenis file untuk dikompresi, masing-masing dengan metode kompresi yang berbeda tergantung pada format file yang dipilih pengguna. Berikut adalah detail jenis file dan cara kompresinya:

### 1. Format: .jpg, .jpeg

- **Jenis Kompresi:** *Lossy Compression (JPEG)*

- **Metode:** Gambar dikonversi ke mode RGB dan disimpan ulang dengan pengaturan kualitas yang lebih rendah, sesuai pilihan pengguna (Tinggi = 80, Sedang = 50, Rendah = 30).
- **Efek:** Mengurangi ukuran file secara signifikan, dengan kemungkinan penurunan kualitas gambar yang sedikit terlihat tergantung level yang dipilih.

## 2. Format: .png

- **Jenis Kompresi:** *Lossless Optimization*
- **Metode:** Gambar PNG dikompresi menggunakan optimasi internal (`optimize=True`) tanpa mengubah kualitas visualnya. Tidak menggunakan pengaturan kualitas karena PNG adalah format lossless.
- **Efek:** Ukuran file berkurang tanpa kehilangan kualitas gambar.

## 3. Format: .pdf

- **Jenis Kompresi:** *Deflate Compression* menggunakan pustaka **PyMuPDF (fitz)**
- **Metode:** File PDF dikompresi dengan opsi `deflate=True`, yang mengurangi redundansi internal dalam dokumen. Tidak mengubah isi visual atau teks.
- **Efek:** Ukuran file lebih kecil, konten tetap sama.

## 4. Format: .docx, .pptx

- **Jenis Kompresi:** *Image Recompression in Archive (Lossy)*
- **Metode:** File Office (.docx/.pptx) diekstrak karena strukturnya adalah file ZIP. Gambar yang terdapat di dalam folder `word/media` atau `ppt/media` kemudian dikompres ulang menggunakan kualitas:
  - Tinggi = 80
  - Sedang = 60
  - Rendah = 40
 Setelah dikompresi, file dikemas ulang menjadi file Office yang utuh.
- **Efek:** Ukuran file dokumen berkurang, kualitas gambar menurun sesuai level.

## 5. Format: Lainnya

- **Status:** Tidak didukung
- **Penanganan:** Jika pengguna memilih format file yang tidak didukung (seperti `.mp4`, `.zip`, `.xlsx`, dll), maka proses kompresi akan gagal dan muncul pesan kesalahan.