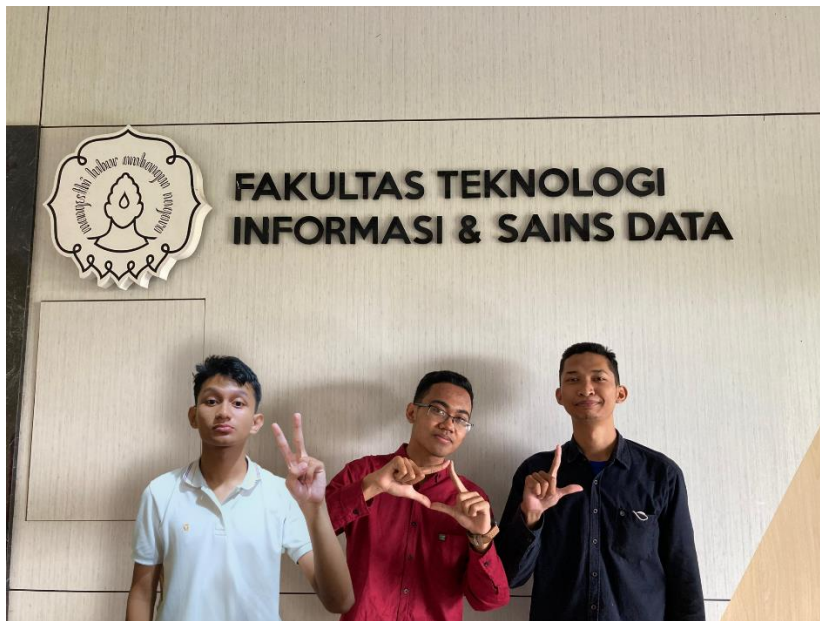


LAPORAN PROYEK ALJABAR LINIER
WEBSITE KOMPRESI CITRA MENGGUNAKAN
PRINCIPAL COMPONENT ANALYSIS (PCA)

Dosen Pengampu : Prof. Drs. Bambang Harjito, M.App.Sc., Ph.D.



Disusun Oleh:

- | | |
|-------------------------|------------|
| 1. Farras Arkan Wardana | (L0123052) |
| 2. Fathoni Nur Habibi | (L0123054) |
| 3. Ivan Wahyu Nugroho | (L0123068) |

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET

2024

Daftar Isi

BAB I DESKRIPSI MASALAH	4
1.1 Deskripsi Umum Tugas Project Pembuatan Web	4
1.2 PCA Untuk Mereduksi Dimensi	5
1.3 Penggunaan Program	6
1.4 Spesifikasi Tugas	7
BAB II LANDASAN TEORI	8
2.1 Perkalian Matriks	8
2.2 Nilai Eigen	8
2.3 Vektor Eigen	9
2.4 Matriks Singular Value Decomposition.....	9
BAB III IMPLEMENTASI PROGRAM	10
3.1 Front End.....	10
3.1.1 JavaScript (script.js).....	10
3.1.2 CSS (style.css, result.css).....	11
3.1.3 HTML (index.html, result.html)	13
3.2 Back End	15
3.2.1 ASGI Configuration (asgi.py).....	15
3.2.2 Settings Configuration (settings.py).....	16
3.2.3 URL Configuration (urls.py).....	18
3.2.4 WSGI Configuration (wsgi.py)	19
3.2.5 Management Script (manage.py)	20
3.2.6 Django Admin Configuration (admin.py)	20
3.2.7 Application Configuration (apps.py).....	21

3.2.8 Image Compression Function (compress_pca.py)	22
3.2.9 Models Definition (models.py)	23
3.2.10 Test Case (test.py)	23
3.2.11 Views Implementation (views.py)	24
3.3 Dokumentasi Website	25
BAB IV EKSPERIMEN	26
4.1. Pengujian Pertama (file .png dengan tingkat kompresi 25%)	26
4.2 Pengujian Kedua (file .jpg dengan tingkat kompresi 20%)	27
4.3 Pengujian Ketiga (file .webp dengan tingkat kompresi 15%)	29
BAB V PENUTUP	32
5.1 Kesimpulan	32
5.2 Saran	32
5.3 Refleksi	34
REFERENSI	35

BAB I

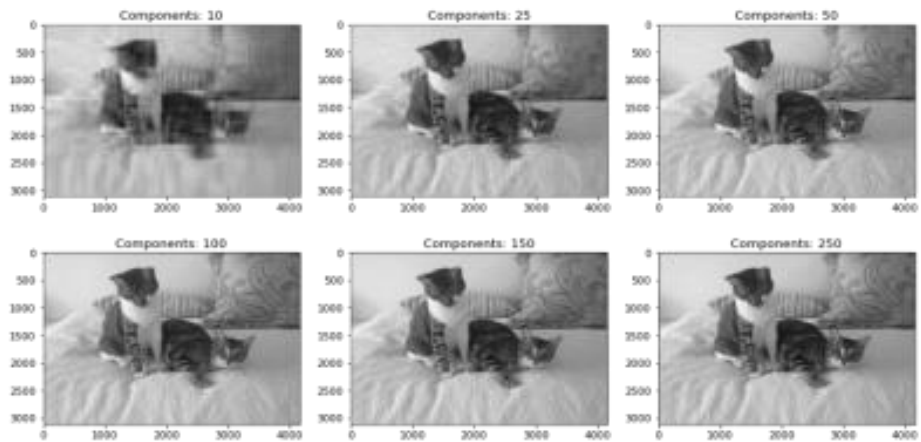
DESKRIPSI MASALAH

1.1 Deskripsi Umum Tugas Project Pembuatan Web

Principal Components Analysis (PCA) ditemukan pada tahun 1901 oleh Karl Pearson. Sebutan lain untuk PCA adalah Transformasi Karhunen-Loeve (*singular value decomposition*) pada matriks, Transformasi Hotelling atau *Proper Orthogonal Decomposition* (POD) ataupun *Empirical orthogonal Function* (EOF).

PCA merupakan suatu teknik yang digunakan untuk mereduksi data multidimensi menjadi data dengan dimensi yang lebih kecil, dengan tetap mempertahankan karakteristik dari data tersebut, untuk menganalisis data dan membuat model peramalan, dalam perhitungannya melibatkan nilai eigen dari matriks kovarians (*singular value decomposition*).

Secara matematis PCA adalah sebuah transformasi linear orthogonal, yaitu transformasi yang memberikan nilai optimal yang didapat dengan metode kuadrat terkecil. Prosedur PCA pada dasarnya adalah bertujuan untuk menyederhanakan variabel yang diamati dengan menyusutkan (mereduksi) dimensinya. Hal ini dilakukan dengan cara menghilangkan korelasi di antara variabel bebas melalui transformasi variabel bebas asal ke variabel baru yang tidak berkorelasi sama sekali atau yang biasa disebut dengan principal component. Gambar 1 menunjukkan terapan PCA yang digunakan untuk kompresi sebuah *image* dengan beberapa komponennya.



Gambar 1, Image kompresi dengan beberapa komponennya

Sumber: <https://dataknowsall.com/blog/imagepca.html>

1.2 PCA Untuk Mereduksi Dimensi

Dimensi dan kompleksitas data yang tinggi dapat mempengaruhi hasil klasifikasi. Untuk itu, perlu adanya proses reduksi dimensi dan kompleksitas dari data yang digunakan untuk mengurangi kesalahan pada proses klasifikasi. reduksi dimensi ini dapat dilakukan menggunakan algoritma PCA. PCA akan membentuk sekumpulan dimensi baru yang kemudian di ranking berdasarkan varian datanya. PCA akan menghasilkan Principal component yang didapat dari dekomposisi eigen value dan eigen vektor dari matriks kovariansi.

Langkah dari algoritma PCA adalah sebagai berikut:

1. Menghitung *mean* \bar{X} dari data pada tiap dimensi menggunakan persamaan (1)

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \quad (1)$$

Dengan:

n = jumlah data sampel

X_i = data sampel

2. Menghitung *Covariansi Matrik* C_x menggunakan persamaan (2)

$$C_x = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(X_i - \bar{X})^T \quad (2)$$

Dengan:

n = jumlah data sampel

X_i = data sampel

\bar{X} = mean

3. Menghitung *eigenvalue* (λ_m) dan *eigenvector* (v_m) dengan persamaan (3)

$$C_x v_m = \lambda_m v_m \quad (3)$$

4. Urutan nilai eigenvalue secara descending. Principal Component (PC) adalah deretan eigenvector sesuai dengan urutan eigenvalue pada tahap 3.
5. Menghasilkan dataset baru.

1.3 Penggunaan Program

Berikut ini adalah input yang dimasukkan sebagai user untuk eksekusi program

1. File image berisi file image input yang ingin dikompresi dengan format file yang bebas selama merupakan format untuk image.
2. Tingkat kompresi, berisi tingkat kompresi dari image (formatnya dibebaskan, cth: Jumlah singular value yang digunakan).



1.4 Spesifikasi Tugas

Buatlah program kompresi image dengan memanfaatkan algoritma PCA dalam bentuk website lokal sederhana. Spesifikasi website adalah sebagai berikut:

1. Website mampu menerima file image beserta input tingkat kompresi image (dibebaskan formatnya).
2. Website mampu menampilkan image input, output, runtime algoritma, dan persentase hasil kompresi image (perubahan jumlah pixel image).
3. File output hasil kompresi dapat diunduh melalui website.
4. Kompresi image tetap mempertahankan warna dari image asli.
5. Bahasa pemrograman yang boleh digunakan adalah Python, Javascript, dan Go.
6. Penggunaan framework untuk back end dan front end website dibebaskan. Contoh framework website yang bisa dipakai adalah Flask, Django, React, Vue, dan Svelte.
7. Kalian dapat menambahkan fitur fungsional lain yang menunjang program yang anda buat (unsur kreativitas diperbolehkan/dianjurkan).
8. Program harus modular dan mengandung komentar yang jelas.
9. Diperbolehkan menggunakan library pengolahan citra seperti OpenCV2, PIL, atau image dari Go.
10. Bahasa program yang digunakan adalah Python dengan menggunakan fungsi-fungsi yang sudah ada maupun dapat membuat fungsi sendiri.

BAB II

LANDASAN TEORI

2.1 Perkalian Matriks

Matriks merupakan sekumpulan bilangan yang tersusun dalam baris dan kolom tertentu. Perkalian matriks adalah proses mengalikan setiap elemen baris pada matriks pertama dengan elemen kolom pada matriks kedua. Kemudian hasilnya dijumlahkan untuk memperoleh hasil akhir perkalian dari tiap elemen matriks. Dua buah matriks dapat dilakukan operasi perkalian apabila jumlah kolom pada matriks pertama sama dengan jumlah baris pada matriks kedua. Kemudian ukuran matriks hasil perkalian memiliki jumlah baris sama dengan matriks pertama dan jumlah kolom sama dengan matriks kedua. Misalnya, matriks A berukuran 3×2 dan matriks B berukuran 2×3 , maka hasil perkalian matriks $A \times B$ akan menghasilkan sebuah matriks berukuran 3×3 . Kemudian perlu diingat bahwa hasil perkalian matriks $A \times B \neq$ matriks $B \times A$.

$$A_{(m \times n)} \times B_{(n \times s)} = C_{(m \times s)}$$

$$A.B = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} m & n \\ o & p \end{bmatrix} \\ = \begin{bmatrix} am + bo & an + bp \\ cm + do & cn + dp \end{bmatrix}$$

2.2 Nilai Eigen

Nilai eigen (eigenvalues) merupakan akar dari polinomial karakteristik yang terkait dengan matriks berukuran $n \times n$. Nilai eigen atau akar karakteristik adalah skalar yang digunakan untuk mengubah vektor eigen, sehingga terdapat suatu

hubungan antara nilai eigen dengan vektor eigen. Persamaan eigen didefinisikan sebagai $Ax = \lambda x$, dimana A merupakan matriks persegi, x adalah vektor eigen, dan λ adalah nilai eigen.

2.3 Vektor Eigen

Vektor eigen (eigenvector) merupakan vektor non nol yang tidak mengubah arah apabila diterapkan suatu transformasi linear, tetapi akan berubah dengan perkalian skalar. Persamaan eigen $Ax = \lambda x$, dimana x adalah vektor eigen. Vektor eigen x menyatakan matriks kolom yang jika dikalikan dengan sebuah matriks $n \times n$ menghasilkan vektor lain yang merupakan kelipatan dari vektor itu sendiri. Sehingga dalam persamaan eigen tersebut menyebabkan vektor x menyusut atau memanjang secara searah maupun secara terbalik terhadap nilai eigen.

2.4 Matriks Singular Value Decomposition

Singular Value Decomposition (SVD) adalah suatu metode yang memecah matriks ukuran $m \times n$ menjadi tiga bagian utama yaitu matriks singular, vektor singular kiri, dan vektor singular kanan.

$$A = U\Sigma V^T$$

Matriks U merupakan matriks vektor singular kiri yang berisi vektor singular yang menyusun sumbu utama. Matriks Σ merupakan matriks singular yang mengandung nilai-nilai singular yang merepresentasikan pentingnya setiap vektor singular dalam matriks asli. Dan matriks V^T adalah matriks vektor singular kanan yang berisi vektor singular yang mempengaruhi atau dipengaruhi oleh matriks asli. SVD berguna dalam pemampatan data karena dapat mengidentifikasi pola dan struktur penting pada matriks asli, serta menghapus informasi yang kurang penting. Dalam pengaplikasiannya, SVD sering digunakan untuk mereduksi dimensi, memproses citra, pengenalan pola, analisis data, kompresi data, dan rekomendasi sistem.

BAB III

IMPLEMENTASI PROGRAM

3.1 Front End

3.1.1 JavaScript (script.js)

Deskripsi:

Kode JavaScript ini mengatur interaktivitas zona drop gambar, mengelola event drag-and-drop serta input file, dan menampilkan thumbnail gambar yang diunggah.

Detail Analisis:

- **Event Listener DOMContentLoaded:** Fungsi ini memastikan bahwa seluruh elemen DOM telah dimuat sebelum menjalankan script, menjamin bahwa elemen HTML seperti dropZone dan fileInput tersedia.
- **Zona Drop (Drag and Drop):** Menggunakan .drop-zone sebagai area di mana pengguna dapat menyeret dan melepaskan gambar untuk diunggah.
 - **click Event:** Ketika zona drop diklik, input file akan terbuka, memungkinkan pengguna memilih gambar secara manual.
 - **change Event:** Memperbarui thumbnail jika ada file yang dipilih melalui dialog file.
 - **dragover dan dragleave Event:** Mengelola gaya visual saat file sedang diseret ke atas zona drop.
 - **drop Event:** Menangani file yang dijatuhkan ke dalam zona drop, mengatur file input dan memperbarui thumbnail.
- **Fungsi updateThumbnail:** Menampilkan thumbnail dari gambar yang dipilih, dengan menggunakan FileReader untuk membaca file gambar sebagai URL data dan kemudian menampilkan URL tersebut sebagai latar belakang elemen thumbnail.

Kelebihan:

- **Interaktivitas:** Memberikan pengalaman pengguna yang intuitif dengan zona drop yang interaktif.
- **Visual Feedback:** Menampilkan thumbnail membantu pengguna memverifikasi bahwa gambar yang diunggah sudah benar.

Potensi Peningkatan:

- **Validasi File:** Bisa ditambahkan validasi untuk memastikan hanya file gambar yang diterima, dan memberikan feedback jika file yang tidak valid diunggah.

3.1.2 CSS (style.css, result.css)**Deskripsi:**

CSS digunakan untuk mendesain tampilan dan tata letak halaman, mencakup styling untuk berbagai elemen seperti header, footer, zona drop, dan tombol.

Detail Analisis:

- **Global Styling:**
 - **body:** Menggunakan font keluarga 'Times New Roman' dan menyesuaikan tata letak dengan margin dan padding yang diatur untuk sentralisasi konten.
 - **.container:** Memberikan batas maksimum lebar, latar belakang putih, padding, border-radius, dan shadow untuk membuat elemen terlihat seperti kotak card yang rapi.
- **Upload Section:**
 - **.drop-zone:** Memiliki batasan maksimal lebar, tinggi, dan border yang terdefinisi dengan baik. Warna dan gaya dashed memberikan indikasi visual area unggah.

- **.drop-zone--over:** Menambahkan perubahan warna latar belakang saat file sedang diseret di atasnya untuk memberikan feedback visual.
- **.drop-zone__thumb:** Thumbnail ditampilkan dalam kotak dengan latar belakang yang diatur untuk menampilkan gambar yang dipilih.
- **Form dan Input:**
 - **.upload-section:** Menyediakan styling untuk form dan input, memastikan bahwa komponen form terlihat rapi dan seimbang.
 - **#n_components:** Mengatur margin untuk elemen input persentase kompresi, sehingga lebih nyaman digunakan.
- **Header dan Footer:**
 - **header dan footer:** Memberikan warna latar belakang biru, padding yang cukup, dan teks yang dipusatkan. Desain ini memastikan bahwa elemen header dan footer konsisten di seluruh halaman.
- **Tombol:**
 - **#download:** Memberikan warna latar belakang biru dengan perubahan warna saat hover, memberikan indikasi interaktif kepada pengguna.

Kelebihan:

- **Konsistensi Desain:** CSS memastikan bahwa elemen-elemen di seluruh halaman memiliki gaya yang konsisten.
- **Responsif:** Menyediakan tata letak yang bisa menyesuaikan dengan berbagai ukuran layar, terutama dengan penggunaan Bootstrap.

Potensi Peningkatan:

- **Media Queries:** Bisa ditambahkan untuk memastikan bahwa semua elemen tampak baik pada layar yang lebih kecil atau lebih besar.
- **Animasi:** Penggunaan animasi CSS bisa meningkatkan pengalaman pengguna saat berinteraksi dengan elemen-elemen di halaman.

3.1.3 HTML (index.html, result.html)

Deskripsi:

- HTML menyediakan struktur dasar halaman untuk kompresi gambar dan menampilkan hasil kompresi.

Detail Analisis:

- **Header dan Footer:**
 - **header:** Menampilkan judul proyek dan informasi dosen pengampu, memberikan konteks mengenai proyek ini.
 - **footer:** Menampilkan informasi kontak dan hak cipta, memastikan bahwa halaman memiliki identitas dan informasi kepemilikan yang jelas.
- **Main Section:**
 - **Form untuk Upload:** Menggunakan form dengan metode POST dan encoding multipart/form-data untuk mengunggah gambar.
 - **drop-zone:** Tempat untuk pengguna mengunggah gambar, baik dengan cara menyeret gambar atau dengan mengklik area tersebut untuk memilih file secara manual.
 - **Input dan Slider untuk Kompresi:** Memberikan opsi kepada pengguna untuk memilih tingkat kompresi gambar, memastikan fleksibilitas dalam penggunaan aplikasi.
 - **Tombol Kompresi:** Mengirimkan gambar dan parameter kompresi untuk diproses di backend.

- **Halaman Hasil:**
 - **Menampilkan Hasil:** Menampilkan gambar sebelum dan sesudah kompresi, memberikan perbandingan visual yang jelas bagi pengguna.
 - **Informasi Kompresi:** Menampilkan informasi tentang tingkat kompresi, waktu kompresi, dan perubahan ukuran file, memberikan informasi yang relevan kepada pengguna.
 - **Tombol Unduh:** Memberikan akses mudah bagi pengguna untuk mengunduh gambar yang sudah dikompresi.

Kelebihan:

- **Struktur yang Jelas:** HTML dirancang dengan struktur yang jelas dan terorganisir, memudahkan dalam pemeliharaan dan pengembangan lebih lanjut.
- **Penggunaan Template Engine:** Menggunakan Django template engine untuk memungkinkan dinamisasi konten halaman seperti URL gambar dan informasi hasil kompresi.

Potensi Peningkatan:

- **SEO Optimization:** Bisa ditambahkan elemen seperti meta tag untuk meningkatkan optimisasi mesin pencari.
- **Aksesibilitas:** Bisa ditingkatkan dengan menambahkan atribut ARIA dan teks alternatif yang lebih lengkap untuk elemen-elemen penting.

3.2 Back End

3.2.1 ASGI Configuration (asgi.py)

Fungsi Utama:

- asgi.py menyediakan konfigurasi untuk menjalankan aplikasi Django menggunakan ASGI (Asynchronous Server Gateway Interface).
- Menyediakan titik masuk untuk aplikasi web yang memungkinkan dukungan protokol lain seperti WebSocket selain HTTP.

Detail Implementasi:

- `os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'imageCompression.settings')`:
 - Mengatur variabel lingkungan untuk menunjuk ke modul pengaturan proyek imageCompression.
 - Ini memastikan bahwa Django tahu di mana menemukan pengaturan untuk proyek ini.
- `get_asgi_application()`:
 - Menghasilkan objek aplikasi ASGI yang digunakan oleh server ASGI untuk menangani request.

Analisis:

- Penggunaan ASGI memungkinkan aplikasi untuk menangani komunikasi asinkron yang efisien, cocok untuk kebutuhan real-time atau jika proyek di masa depan memerlukan dukungan untuk WebSocket atau komunikasi non-HTTP lainnya.

3.2.2 Settings Configuration (settings.py)

Fungsi Utama:

- Menyediakan konfigurasi global untuk proyek Django, termasuk pengaturan untuk keamanan, database, aplikasi yang diinstal, dan pengaturan untuk file statis.

Detail Implementasi:

- **BASE_DIR:**
 - Mendefinisikan direktori dasar proyek menggunakan `Path(__file__).resolve().parent.parent`.
 - Memudahkan pengelolaan path relatif di seluruh proyek.
- **SECRET_KEY:**
 - Kunci rahasia yang digunakan oleh Django untuk menyediakan kriptografi dan menjaga keamanan.
 - **Penting:** Jangan pernah membagikan kunci ini secara publik dan pastikan untuk menggantinya dengan nilai yang aman di lingkungan produksi.
- **DEBUG = True:**
 - Mengaktifkan mode debug yang membantu pengembang dengan memberikan detail error yang rinci.
 - **Peringatan:** Mode ini harus dimatikan (`DEBUG = False`) di lingkungan produksi untuk mencegah kebocoran informasi sensitif.
- **ALLOWED_HOSTS = ['.vercel.app']:**
 - Mengizinkan domain tertentu (dalam hal ini, `.vercel.app`) untuk mengakses aplikasi.
 - **Pengaturan ini penting** untuk mencegah Host Header attacks.

- **Aplikasi yang Diinstal (INSTALLED_APPS):**
 - Daftar aplikasi Django yang diinstal dan aktif dalam proyek, termasuk aplikasi compression yang dibuat khusus untuk proyek ini.
- **Middlewares (MIDDLEWARE):**
 - Berbagai middleware yang mengelola request/response, keamanan, sesi, dan lainnya.
 - Contoh penting termasuk SecurityMiddleware untuk keamanan dan CsrfViewMiddleware untuk melindungi dari serangan CSRF.
- **Pengaturan Database (DATABASES):**
 - Menggunakan SQLite ('ENGINE': 'django.db.backends.sqlite3') yang ringan dan mudah digunakan untuk pengembangan.
 - Di masa depan, database yang lebih kuat mungkin diperlukan untuk lingkungan produksi.
- **Static Files (STATICFILES_DIRS):**
 - Menentukan direktori static yang digunakan untuk menyimpan file statis seperti CSS, JavaScript, dan gambar.

Analisis:

- Konfigurasi ini memastikan bahwa aplikasi memiliki dasar yang kuat untuk beroperasi dengan aman dan efisien.
- Beberapa pengaturan harus diubah atau ditingkatkan ketika mengembangkan aplikasi lebih lanjut atau saat memasuki fase produksi.

3.2.3 URL Configuration (urls.py)

Fungsi Utama:

- Mengatur routing untuk aplikasi web, memastikan bahwa URL yang diminta pengguna diteruskan ke tampilan atau logika yang sesuai.

Detail Implementasi:

- **ROOT_URLCONF:**
 - Menunjukkan modul URL utama untuk proyek, yaitu `imageCompression.urls`.
- **urlpatterns:**
 - Daftar pola URL dan tampilan yang sesuai.
 - Setiap pola URL dipetakan ke fungsi tampilan atau aplikasi lain yang akan menangani request tersebut.

Analisis:

- Pengaturan URL yang baik sangat penting untuk memastikan pengguna dapat mengakses fitur aplikasi dengan mudah dan URL yang diminta diproses dengan benar.
- URL disusun dengan baik memungkinkan penambahan fitur baru tanpa mengganggu fungsionalitas yang sudah ada.

3.2.4 WSGI Configuration (wsgi.py)

Fungsi Utama:

- Menyediakan konfigurasi untuk menjalankan aplikasi Django menggunakan WSGI (Web Server Gateway Interface).
- Memungkinkan aplikasi untuk beroperasi dengan server web yang mendukung protokol WSGI, seperti Gunicorn atau uWSGI.

Detail Implementasi:

- **get_wsgi_application():**
 - Menghasilkan objek aplikasi WSGI yang digunakan oleh server WSGI untuk menangani request.
- **Penggunaan:**
 - WSGI adalah protokol standar untuk aplikasi web Python dan berguna untuk deployment aplikasi ke server yang tidak mendukung ASGI.

Analisis:

- Penggunaan WSGI adalah standar untuk banyak server web dan penting untuk aplikasi yang tidak memerlukan komunikasi real-time atau protokol non-HTTP.
- Menyediakan fondasi yang solid untuk deployment aplikasi di berbagai lingkungan server.

3.2.5 Management Script (manage.py)

Fungsi Utama:

- Menyediakan antarmuka baris perintah untuk berbagai tugas manajemen dalam proyek Django, seperti menjalankan server pengembangan, migrasi database, dan lainnya.

Detail Implementasi:

- **main():**
 - Fungsi utama yang menjalankan tugas manajemen, termasuk menjalankan server atau melakukan migrasi database.
- **os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'imageCompression.settings'):**
 - Mengatur modul pengaturan untuk proyek.

Analisis:

- manage.py adalah skrip penting untuk mengelola proyek Django dan mempermudah berbagai tugas pengembangan dan administrasi.

3.2.6 Django Admin Configuration (admin.py)

Fungsi Utama:

- Menyediakan konfigurasi untuk Django Admin, yang memungkinkan pengelolaan data model melalui antarmuka web yang disediakan oleh Django.

Detail Implementasi:

- **Saat ini, admin.py kosong**, yang menunjukkan bahwa tidak ada model yang didaftarkan untuk pengelolaan melalui Django Admin.

Analisis:

- Meskipun tidak ada model yang didaftarkan, pengaturan ini menyediakan kerangka untuk menambahkan model di masa depan jika pengelolaan data melalui antarmuka admin diperlukan.
- Mengaktifkan fitur admin ini dapat mempermudah pengelolaan data dan pengujian model.

3.2.7 Application Configuration (apps.py)

Fungsi Utama:

- Mengonfigurasi aplikasi compression dengan pengaturan dasar yang diperlukan untuk integrasi dengan proyek Django.

Detail Implementasi:

- **CompressionConfig:**
 - Kelas konfigurasi aplikasi dengan name yang diatur ke compression dan default_auto_field diatur ke BigAutoField.
 - Pengaturan ini memastikan bahwa primary key untuk model akan otomatis menggunakan tipe BigAutoField, yang dapat menangani jumlah data yang besar.

Analisis:

- Pengaturan ini memastikan aplikasi compression terintegrasi dengan benar dalam proyek Django dan siap digunakan untuk pengembangan fitur yang lebih lanjut.

3.2.8 Image Compression Function (compress_pca.py)

Fungsi Utama:

- Mengimplementasikan fungsi kompresi gambar menggunakan algoritma PCA untuk mengurangi ukuran gambar tanpa mengorbankan terlalu banyak kualitas.

Detail Implementasi:

- **compress_image(image_path, compress_percentage):**
 - Membaca gambar dari path yang diberikan, memisahkan ke dalam kanal warna (R, G, B), dan melakukan kompresi menggunakan PCA sesuai persentase yang ditentukan.
 - Fungsi ini juga mengembalikan gambar yang dikompresi dalam bentuk objek PIL.
- **Penggunaan PCA:**
 - **PCA (Principal Component Analysis)** digunakan untuk mengurangi dimensi data dengan mengidentifikasi komponen utama dari gambar.
 - Setiap kanal warna diperlakukan sebagai matriks dua dimensi yang kemudian dikompresi dengan mempertahankan komponen utama yang sesuai dengan persentase kompresi.

Analisis:

- Penggunaan PCA sangat efektif untuk kompresi gambar karena dapat mengurangi ukuran gambar dengan tetap menjaga kualitas gambar yang cukup baik.
- Pendekatan ini cocok untuk aplikasi web yang membutuhkan pengurangan ukuran gambar untuk mempercepat waktu pemuatan dan menghemat ruang penyimpanan.

3.2.9 Models Definition (models.py)

Fungsi Utama:

- Menyediakan definisi untuk model data yang akan digunakan dalam aplikasi.

Detail Implementasi:

- **Saat ini models.py kosong**, yang berarti tidak ada model data yang didefinisikan untuk aplikasi ini.

Analisis:

- Meskipun tidak ada model saat ini, file ini menyediakan tempat untuk mendefinisikan model di masa depan jika diperlukan untuk menyimpan informasi atau metadata terkait gambar yang diunggah dan dikompresi.
- Menggunakan model dapat membantu dalam mengelola data dan memberikan struktur yang lebih baik untuk aplikasi.

3.2.10 Test Case (test.py)

Fungsi Utama:

- Menyediakan kerangka untuk pengujian aplikasi guna memastikan bahwa fungsi-fungsi bekerja sesuai yang diharapkan.

Detail Implementasi:

- **Saat ini test.py kosong**, yang berarti tidak ada pengujian yang didefinisikan untuk aplikasi ini.

Analisis:

- Menambahkan pengujian di masa depan sangat penting untuk memastikan aplikasi berfungsi dengan baik dan membantu dalam mendeteksi bug atau masalah.
- Pengujian dapat mencakup berbagai skenario seperti pengujian kompresi gambar, pengujian ukuran file, dan penanganan berbagai format gambar.

3.2.11 Views Implementation (views.py)

Fungsi Utama:

- Mengelola logika terkait dengan request pengguna dan menghasilkan respon yang sesuai untuk berbagai URL yang diminta.

Detail Implementasi:

- **index View:**
 - **index(request):**
 - Mengelola request GET dan POST.
 - Jika POST, menerima gambar yang diunggah, menyimpannya, dan memanggil fungsi kompresi PCA.
 - Menghitung waktu kompresi dan perubahan ukuran file.
 - Menyimpan gambar sebelum dan sesudah kompresi.
 - Menggunakan render untuk menghasilkan template index.html dengan data yang diperlukan.

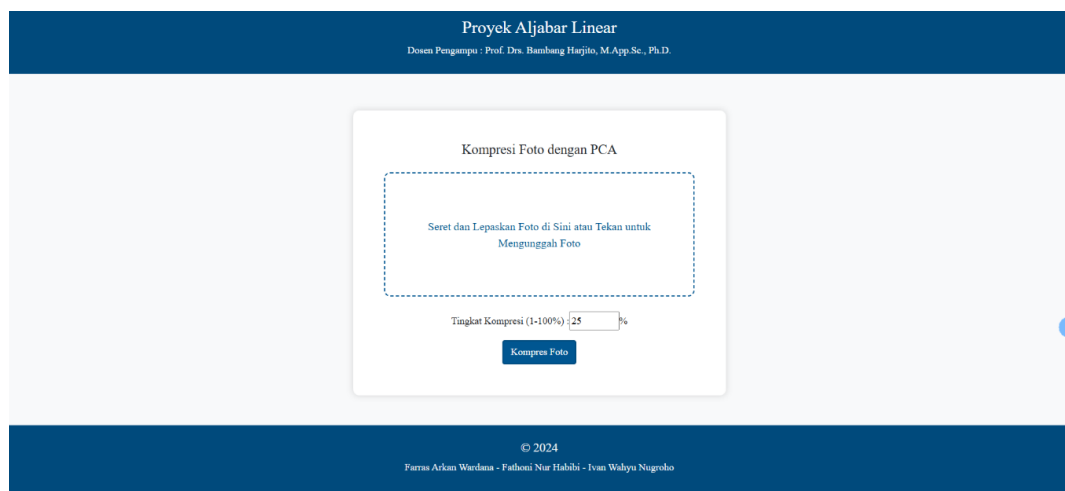
- **download View:**
 - **download(request):**
 - Menyediakan gambar yang dikompresi untuk diunduh melalui URL yang ditentukan.
 - Menggunakan FileResponse untuk mengirimkan gambar sebagai file respons.

Analisis:

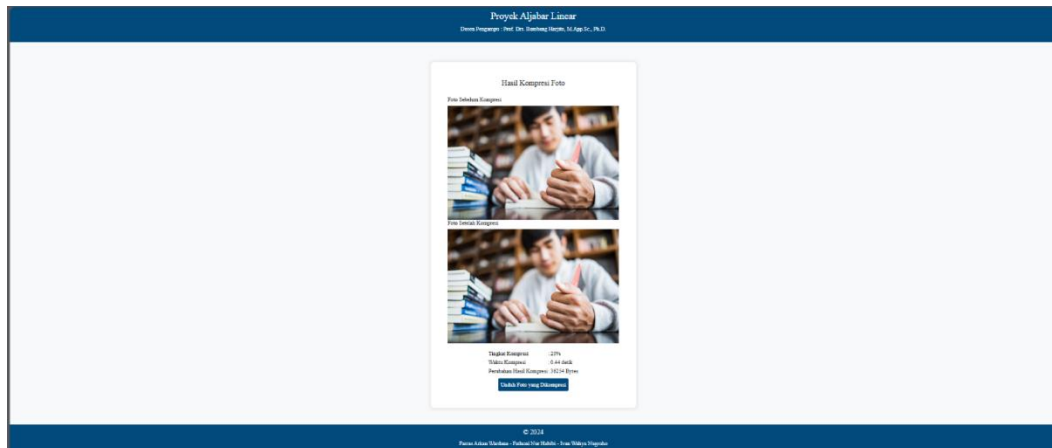
- Tampilan index dan download dirancang untuk memberikan alur kerja yang mulus bagi pengguna yang ingin mengunggah dan mengompresi gambar serta mengunduh hasilnya.
- Mengukur waktu eksekusi dan perubahan ukuran file memberikan informasi yang berharga bagi pengguna dan dapat digunakan untuk mengoptimalkan kinerja di masa mendatang.

3.3 Dokumentasi Website

Berikut beberapa dokumentasi implementasi dari program yg sudah dijabarkan sebelumnya :



Gambar 3.3.1 Tampilan awal website



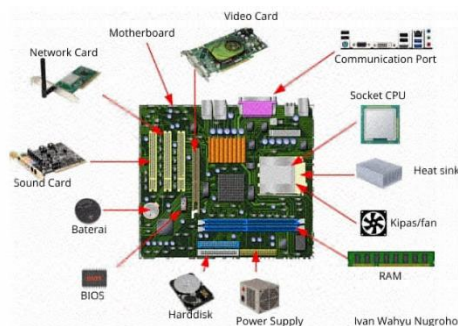
Gambar 3.3.2 Tampilan akhir website

BAB IV EKSPERIMEN

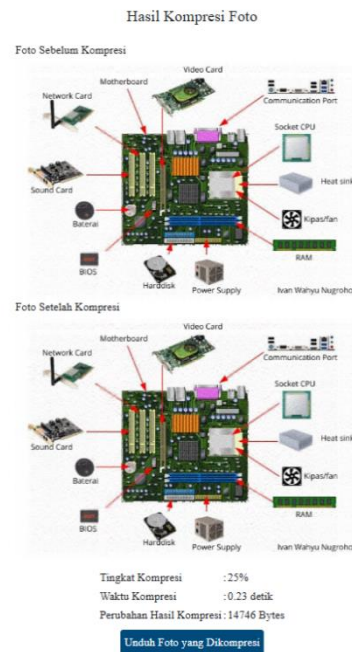
Dalam folder *test*, terdapat beberapa pengujian dari *website* ini. Pengujian dilakukan dengan masukan *file image* dengan tingkat kompresi berbeda-beda. Selain itu, pengujian dilakukan dengan masukan *file image* berbagai ekstensi, seperti .png, .jpg, dan .webp.

4.1. Pengujian Pertama (file .png dengan tingkat kompresi 25%)

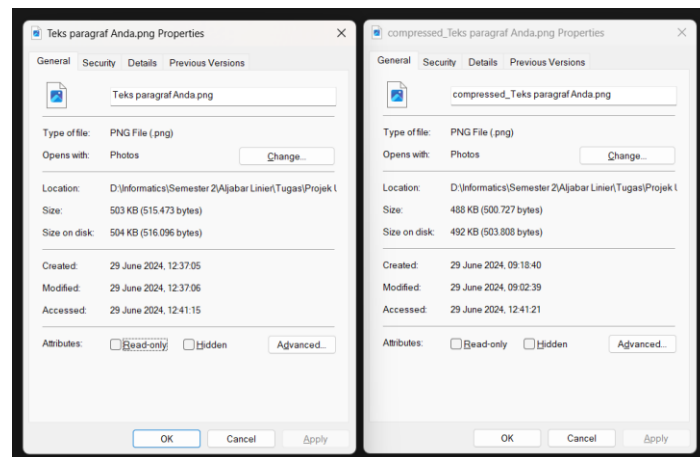
Berikut ini file masukan yang diberikan :



Berikut ini tampilan *website* pasca kompresi :



Berikut ini ukuran yang file sebelum dan sesudah kompresi :



Hasil yang diperoleh adalah file mengalami penurunan setelah kompresi. Sebelum kompresi, ukuran file original adalah 503 kB (foto kiri) dan setelah kompresi, ukuran file adalah 488 kB (foto kanan). Hal ini membuktikan kompresi berjalan dengan baik.

4.2 Pengujian Kedua (file .jpg dengan tingkat kompresi 20%)

Berikut ini adalah file masukan yang digunakan untuk pengujian kedua :



Berikut ini tampilan *website* pasca kompresi :

Hasil Kompresi Foto

Foto Sebelum Kompresi



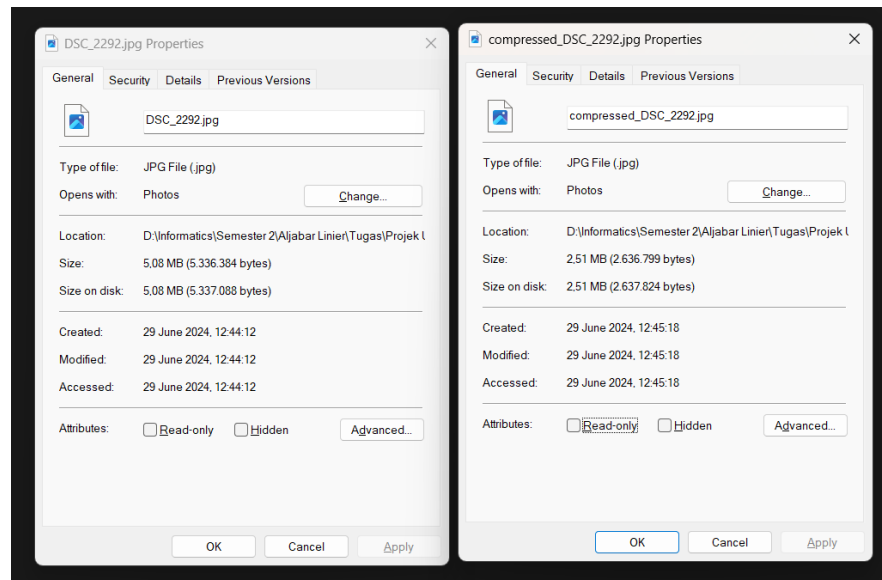
Foto Setelah Kompresi



Tingkat Kompresi : 20%
 Waktu Kompresi : 66.11 detik
 Perubahan Hasil Kompresi: 2699585 Bytes

[Unduh Foto yang Dikompresi](#)

Berikut ini ukuran yang file sebelum dan sesudah kompresi :



Hasil yang diperoleh adalah file mengalami penurunan setelah kompresi. Sebelum kompresi, ukuran file original adalah 5,08 MB (foto kiri) dan setelah kompresi, ukuran file adalah 2,51 MB (foto kanan). Hal ini membuktikan kompresi berjalan dengan baik.

4.3 Pengujian Ketiga (file .webp dengan tingkat kompresi 15%)

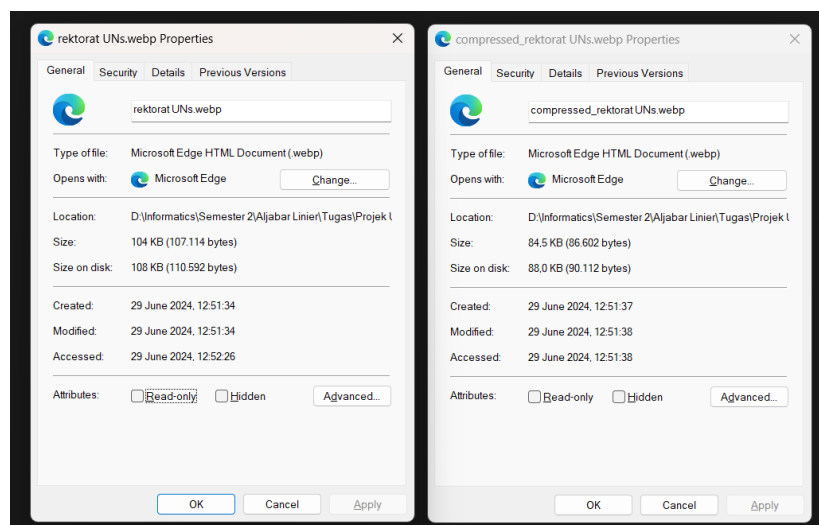
Berikut ini file masukan yang diberikan :



Berikut ini tampilan *website* pasca kompresi :



Berikut ini ukuran yang file sebelum dan sesudah kompresi :



Hasil yang diperoleh adalah file mengalami penurunan setelah kompresi. Sebelum kompresi, ukuran file original adalah 104 kB (foto kiri) dan setelah kompresi, ukuran file adalah 84,5 kB (foto kanan). Hal ini membuktikan kompresi berjalan dengan baik.

Berdasarkan beberapa sampel pengujian, diperoleh kesimpulan uji sebagai berikut :

1. Besarnya file *input* dan waktu lama berjalannya algoritma PCA adalah berbanding lurus.
2. Dimensi foto sebelum dan sesudah kompresi dapat dipertahankan.
3. Warna foto sebelum dan sesudah kompresi dapat dipertahankan.
4. Ukuran foto hasil kompresi akan menyesuaikan terhadap besarnya tingkat kompresi yang diberikan.

BAB V

PENUTUP

5.1 Kesimpulan

Penelitian ini berhasil menunjukkan efektivitas Principle Component Analysis (PCA) dalam kompresi gambar, dengan hasil yang menunjukkan pengurangan dimensi data yang signifikan tanpa mengorbankan kualitas visual. Metode PCA mampu mengidentifikasi komponen utama yang paling berpengaruh dalam gambar, memungkinkan penghapusan komponen yang kurang signifikan dan menghasilkan ukuran file yang lebih kecil. Pengujian pada berbagai format gambar menunjukkan bahwa metode ini tidak hanya efisien dalam mengurangi ukuran file tetapi juga mempertahankan detail dan ketajaman gambar yang baik. Dengan implementasi yang tepat, PCA dapat menjadi solusi efektif untuk mengatasi masalah penyimpanan dan pengiriman gambar berukuran besar. Selain itu, penggunaan PCA juga mempermudah dalam manajemen data gambar yang besar, khususnya dalam aplikasi yang memerlukan efisiensi ruang penyimpanan dan kecepatan transmisi. Penerapan PCA dalam kompresi gambar juga memberikan fleksibilitas dalam mengatur tingkat kompresi sesuai dengan kebutuhan, sehingga dapat disesuaikan untuk berbagai skenario penggunaan. Kesimpulannya, metode PCA menawarkan kombinasi yang baik antara efisiensi kompresi dan kualitas gambar, menjadikannya pilihan yang layak untuk aplikasi kompresi gambar modern. Dengan hasil yang memuaskan ini, diharapkan penelitian lebih lanjut dapat mengembangkan metode ini untuk aplikasi yang lebih luas, termasuk dalam sistem manajemen gambar dan transmisi data digital.

5.2 Saran

1. Penambahan Format Gambar

Meskipun aplikasi saat ini mendukung format gambar populer seperti JPEG dan PNG, penambahan dukungan untuk format gambar lain seperti BMP dan GIF dapat membuat aplikasi lebih fleksibel dan berguna untuk berbagai kebutuhan pengguna.

2. Fitur Pratinjau Kompresi

Mengembangkan fitur pratinjau yang memungkinkan pengguna untuk melihat hasil kompresi sebelum mereka mengunduh gambar akan meningkatkan pengalaman pengguna dan memberikan mereka gambaran yang lebih jelas tentang perubahan yang terjadi.

3. Optimisasi Kinerja:

Meskipun kompresi citra dengan PCA sudah efisien, penambahan algoritma kompresi lain atau teknik optimisasi lebih lanjut dapat membantu dalam mempercepat proses kompresi dan membuat aplikasi lebih responsif, terutama untuk gambar dengan ukuran besar.

4. Peningkatan Keamanan

Memperkuat aspek keamanan aplikasi seperti validasi file yang diunggah untuk mencegah potensi ancaman seperti injeksi file berbahaya dan memastikan bahwa hanya format gambar yang valid yang dapat diproses oleh aplikasi.

5. Integrasi *Cloud Storage*

Menambahkan fitur untuk menyimpan gambar yang diunggah dan hasil kompresi di layanan penyimpanan awan (*cloud storage*) akan meningkatkan fleksibilitas dan aksesibilitas aplikasi, memungkinkan pengguna untuk mengakses gambar mereka dari mana saja.

6. *User Experience* (UX) dan *User Interface* (UI)

Pengembangan lebih lanjut pada tampilan dan pengalaman pengguna dengan menambahkan animasi, feedback saat proses kompresi berlangsung, dan panduan pengguna yang lebih intuitif dapat membuat aplikasi lebih ramah pengguna.

5.3 Refleksi

Pengerjaan tugas ini memberikan pemahaman yang mendalam tentang bagaimana algoritma PCA dapat diaplikasikan dalam dunia nyata untuk memecahkan masalah kompresi citra. Penggunaan Django sebagai framework pengembangan web memberikan pengalaman berharga dalam mengelola backend dan frontend aplikasi secara lebih terstruktur. Selain itu, integrasi dengan Bootstrap memperkaya kemampuan dalam merancang antarmuka pengguna yang modern dan responsif.

Selama pengerjaan, tantangan utama yang dihadapi adalah dalam optimasi kinerja kompresi dan pengelolaan ukuran file yang besar. Hal ini memacu untuk belajar lebih dalam mengenai manajemen memori dan efisiensi algoritma. Pengalaman ini juga menyoroti pentingnya memahami kebutuhan pengguna dan bagaimana menyediakan solusi yang tidak hanya fungsional, tetapi juga mudah digunakan.

Secara keseluruhan, tugas ini memperkaya pengetahuan dalam pemrograman dan pengembangan aplikasi berbasis web, serta memberikan wawasan praktis tentang bagaimana teknologi seperti PCA dapat digunakan untuk menghasilkan solusi yang inovatif dan bermanfaat dalam kehidupan sehari-hari.

REFERENSI

- Nicholson, W. Keith. (2021). Linear Algebra with Application. Alberta: Lyryx Learning Inc.
- Nuraini, R. (2015). Desain algoritma operasi perkalian matriks menggunakan metode flowchart. *Jurnal Teknik Komputer Amik Bsi*, 1(1), 144-151.