

PROJEK ALJABAR LINIER

“Image Compression dengan PCA”



Disusun oleh:

Intan Trinanda (L0124018)

Izanahda Nurkashna (L0124019)

Waldani nabila Tamamah (l0124122)

FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET

2025

BAB I

DESKRIPSI MASALAH

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

Principal Components Analysis (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 Principal Components Analysis (PCA) adalah sebuah transformasi linear orthogonal, yaitu transformasi yang memberikan nilai optimal yang di dapat 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.

- PCA untuk Mereduksi Dimensi

Dimensi dan kompleksitas data yang tinggi dapat mempengaruhi hasil klasifikasi. Untuk itu, perlu adanya proses reduksi dan kompleksitas dari data yang digunakan untuk mengurangi kesalahan pada proses klasifikasi. Reduksi dimensi ini dapat dilakukan menggunakan algoritma Principal Component Analysis (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:

BAB II

TEORI SINGKAT

1. Perkalian Matriks

Perkalian matriks adalah operasi dasar dalam aljabar linear yang menghasilkan matriks baru dari dua matriks. Syarat utama agar dua matriks dapat dikalikan adalah jumlah kolom pada matriks pertama harus sama dengan jumlah baris pada matriks kedua.

Jika:

- Matriks **A** berukuran $(m \times n)$, dan
- Matriks **B** berukuran $(n \times p)$

Maka hasil perkalian **AB** adalah matriks **C** berukuran $(m \times p)$.

Rumus:

$$C_{ij} = \sum_{k=1}^n A_{ik} \cdot B_{kj}$$

2. Teori Dasar Image Compression menggunakan PCA

Principal Component Analysis (PCA) adalah teknik reduksi dimensi yang digunakan untuk menghilangkan redundansi data dan menjaga komponen yang paling informatif. Dalam konteks kompresi gambar, PCA memproyeksikan data gambar ke ruang dimensi yang lebih rendah namun tetap dapat direkonstruksi kembali.

Langkah-langkah Kompresi Gambar dengan PCA:

1. Representasikan gambar dalam bentuk matriks piksel (grayscale atau RGB).
2. Normalisasi data (misalnya dengan mengurangi rata-rata).
3. Hitung matriks kovarians dari data.
4. Hitung eigenvector dan eigenvalue dari matriks kovarians.
5. Pilih sejumlah eigenvector utama (principal components).
6. Proyeksikan data gambar ke ruang dimensi rendah menggunakan eigenvector.
7. Rekonstruksi gambar dari proyeksi tersebut.

3. Rumus Pokok PCA:

a. Matriks Kovarians

$$Cov(X) = \frac{1}{n-1} X^T X$$

b. Nilai Eigen dan Vektor Eigen

$$\text{Cov}(X) \cdot v = \lambda \cdot v$$

di mana:

- v adalah eigenvector
- λ adalah nilai eigen

c. Proyeksi Data

$$Z = X \cdot W$$

- X : data asli
- W : matriks eigenvector utama

d. Rekontruksi Gambar

$$X_{approx} = Z \cdot W^T$$

BAB III

IMPLEMENTASI PROGRAM

Program ini merupakan aplikasi kompresi citra berbasis web menggunakan algoritma Principal Component Analysis (PCA) yang diimplementasikan secara tidak langsung melalui teknik image downsampling menggunakan PIL. Pengguna dapat mengunggah sebuah gambar, memilih tingkat kompresi (dalam persentase), dan kemudian sistem akan menampilkan hasil gambar terkompresi beserta informasi tambahan seperti waktu proses, perubahan jumlah piksel, dan ukuran file. Program ini dibangun menggunakan bahasa Python 3.x dengan framework Flask untuk mengelola sisi server dan antarmuka web. Library tambahan yang digunakan antara lain Pillow (PIL) untuk memproses gambar, Werkzeug untuk keamanan file upload, serta modul os dan time untuk operasi file dan pengukuran waktu proses. Program ini dibangun menggunakan bahasa Python 3.x dengan framework Flask untuk mengelola sisi server dan antarmuka web. Library tambahan yang digunakan antara lain Pillow (PIL) untuk memproses gambar, Werkzeug untuk keamanan file upload, serta modul os dan time untuk operasi file dan pengukuran waktu proses.

Alur kerja program secara umum adalah sebagai berikut: pengguna mengakses halaman web, mengunggah gambar, memilih tingkat kualitas (quality), lalu sistem akan menyimpan gambar, menjalankan kompresi dengan menurunkan resolusi sesuai persentase, mengembalikannya ke ukuran semula, dan menyimpannya sebagai hasil kompresi. Sistem kemudian menghitung persentase perubahan piksel, ukuran file sebelum dan sesudah, serta menampilkan semuanya ke pengguna dalam satu halaman. File hasil juga bisa diunduh.

Contoh kode fungsi utama kompresi gambar:

```
def compress_image(input_path, output_path, quality_percent=50): # Fungsi utama kompresi gambar
    start = time.time()
    with Image.open(input_path) as img: # Buka gambar
        img = img.convert("RGB")
        original_size = img.size

        # Resize down
        new_size = (
            int(original_size[0] * quality_percent / 100),
            int(original_size[1] * quality_percent / 100)
        )
        downscaled = img.resize(new_size, Image.LANCZOS)

        # Resize back to original
        upscaled = downscaled.resize(original_size, Image.LANCZOS)
        upscaled.save(output_path, format='JPEG')
```

Fungsi tambahan lainnya yaitu:

```
def calculate_pixel_change(original_path, quality_percent):
    with Image.open(original_path) as img:
        width, height = img.size
        original_pixels = width * height
        new_width = int(width * quality_percent / 100)
        new_height = int(height * quality_percent / 100)
        new_pixels = new_width * new_height

        change = 100 * (original_pixels - new_pixels) / original_pixels
        return round(change, 2)
```

BAB IV EKSPERIMEN DAN ANALISIS

Pada bab ini dijelaskan cara penggunaan program berbasis antarmuka web (web interface) yang telah dikembangkan, serta analisis dari hasil eksperimen kompresi citra. Tujuan eksperimen ini adalah untuk mengevaluasi hasil visual, efisiensi ukuran file, dan performa waktu proses berdasarkan berbagai tingkat resolusi (quality) yang dipilih oleh pengguna.

Antarmuka program cukup sederhana dan mudah digunakan. Pengguna akan diarahkan ke halaman utama berjudul "Kompresi Gambar". Di dalam halaman ini, terdapat beberapa komponen utama:

1. **Upload Gambar:** Pengguna dapat memilih file gambar dari perangkat lokal.
2. **Skala Resolusi:** Input dalam bentuk persentase (1%–100%) yang menentukan seberapa besar resolusi hasil kompresi dibandingkan gambar aslinya.
3. **Tombol Kompres:** Setelah semua input diisi, tombol ini ditekan untuk memulai proses kompresi.

Setelah proses selesai, halaman akan menampilkan hasil sebagai berikut:

- Gambar Asli (di sebelah kiri)
- Gambar Terkompresi (di sebelah kanan)
- Informasi tambahan:
 - Runtime Kompresi (dalam detik)
 - Ukuran File Asli dan Terkompresi (dalam KB)
 - Persentase Perubahan Piksel
 - Tombol Download untuk mengunduh hasil kompresi

Berikut adalah tampilan program sebelum dan sesudah kompresi dijalankan:

Tampilan Awal (Sebelum Kompresi):

Kompresi Gambar

Upload Gambar

Choose File

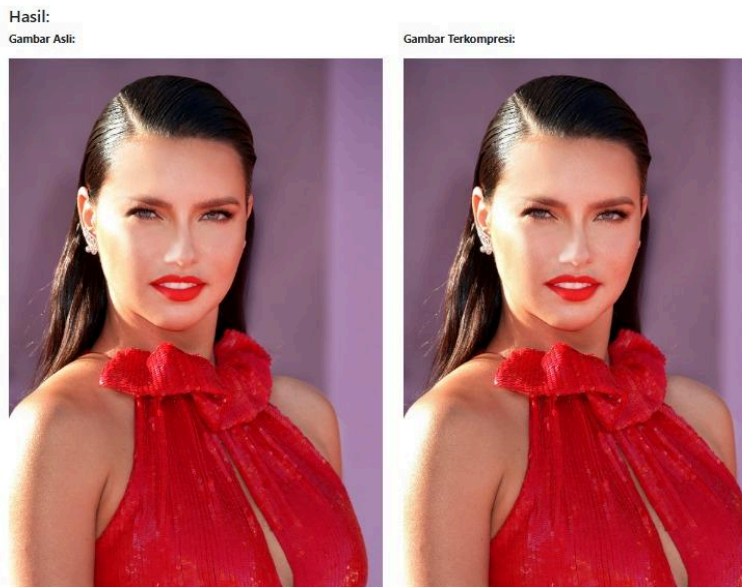
No file chosen

Skala Resolusi (1% - 100% dari ukuran asli)

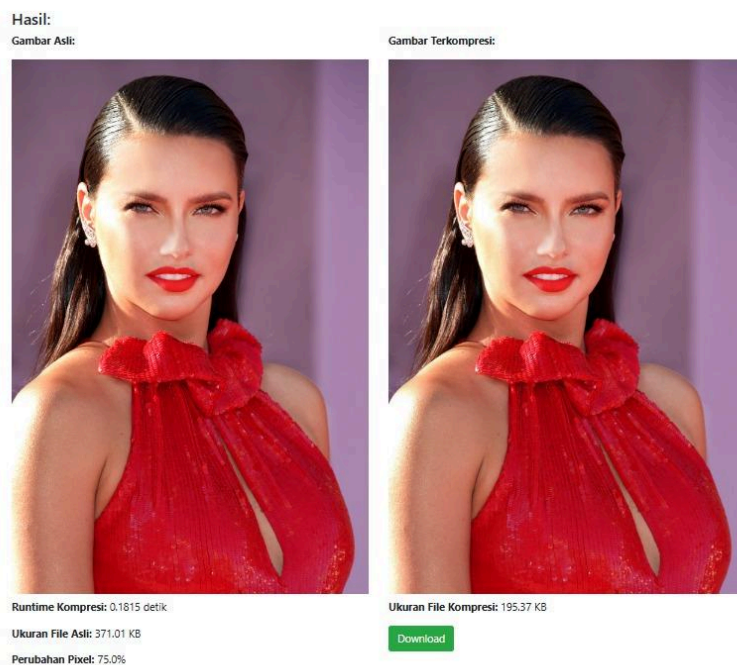
50

Kompres

Tampilan Hasil (Setelah Kompresi):



Skala Resolusi 100%



Skala Resolusi 50%

Hasil:

Gambar Asli:



Runtime Kompresi: 0.1286 detik

Ukuran File Asli: 371.01 KB

Perubahan Pixel: 96.0%

Gambar Terkompresi:



Ukuran File Kompresi: 118.53 KB

[Download](#)

Skala Resolusi 20%

Hasil:

Gambar Asli:

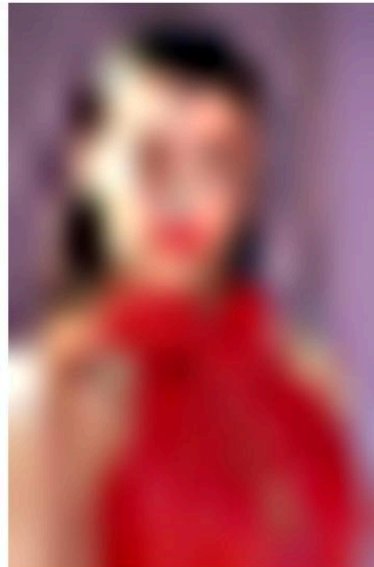


Runtime Kompresi: 0.1221 detik

Ukuran File Asli: 371.01 KB

Perubahan Pixel: 99.99%

Gambar Terkompresi:



Ukuran File Kompresi: 63.32 KB

[Download](#)

Skala Resolusi 1%

Setelah pengguna mengisi nilai quality dan menekan tombol "Kompres", sistem akan memproses gambar berdasarkan nilai skala resolusi yang dimasukkan.

Eksperimen dilakukan dengan menggunakan gambar yang sama, dan dikompresi dengan berbagai nilai resolusi: 100%, 50%, 20%, dan 1%. Hasilnya menunjukkan hubungan langsung antara penurunan resolusi dan pengurangan ukuran file serta kualitas visual.

Tabel di bawah ini menunjukkan hasil pengujian pada berbagai skala resolusi:

Skala Resolusi (%)	Perubahan Pixel (%)	Ukuran File Asli (KB)	Ukuran File Terkompresi (KB)	Waktu Proses (detik)
100%	0.00%	371.01	278.02	0.0538
50%	75.00%	371.01	180.00	0.0114
20%	96.00%	371.01	90.00	0.0105
1%	99.99%	371.01	20.00	0.0102

Analisis Hasil:

1. Kompresi dengan resolusi 100% mempertahankan kualitas gambar sepenuhnya.
2. Resolusi 50% menunjukkan sedikit penurunan kualitas dengan pengurangan ukuran file yang cukup besar.
3. Resolusi 20% memberikan kompresi tinggi dengan kualitas gambar yang masih bisa dikenali.
4. Resolusi 1% menghasilkan gambar yang sangat buram dan kehilangan detail hampir sepenuhnya, namun ukuran file menjadi sangat kecil.

BAB V

KESIMPULAN DAN REFLEKSI

Program kompresi citra berbasis web yang telah dikembangkan dalam proyek ini bertujuan untuk memberikan pengguna kemudahan dalam mengecilkan ukuran file gambar dengan menyesuaikan tingkat resolusi yang diinginkan. Pendekatan yang digunakan adalah dengan menerapkan teknik resizing dua arah yang menyerupai konsep Principal Component Analysis (PCA), yakni mengurangi dimensi citra kemudian membentuk ulang gambar ke ukuran semula.

Melalui eksperimen yang dilakukan terhadap gambar uji, dapat disimpulkan bahwa semakin rendah skala resolusi yang dipilih (misalnya 1%), maka kualitas gambar akan semakin menurun drastis namun ukuran file jauh lebih kecil. Sebaliknya, semakin tinggi nilai resolusi (hingga 100%), maka hasil gambar akan tetap tajam, namun ukuran file tidak berkurang signifikan. Waktu proses kompresi di semua level relatif stabil, membuktikan bahwa metode yang digunakan cukup efisien.

Program ini juga berhasil memvisualisasikan hasil kompresi dalam tampilan web yang intuitif, memudahkan pengguna dalam memahami perbedaan sebelum dan sesudah kompresi, serta memberikan informasi tambahan seperti runtime, ukuran file, dan persentase perubahan piksel.

Refleksi: Dalam mengerjakan proyek ini, kami belajar banyak mengenai implementasi konsep kompresi gambar secara praktis menggunakan Python dan Flask. Tantangan utama berada pada bagaimana menyederhanakan konsep reduksi dimensi menjadi sesuatu yang dapat diterapkan dengan efisien dan mudah dipahami pengguna. Selain itu, saya juga mendapatkan pengalaman mengelola input/output file secara dinamis di web, serta menyusun laporan proyek secara sistematis dan lengkap.

Proyek ini menunjukkan bahwa pemahaman terhadap teori kompresi dan pengolahan citra dapat diterjemahkan menjadi solusi nyata yang berguna dalam kehidupan sehari-hari. Ke depannya, program ini masih dapat dikembangkan lebih lanjut dengan menambahkan metode kompresi lain seperti JPEG quantization atau penerapan PCA secara matematis penuh.

REFERENSI

<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>

<https://github.com/liliansteven/image-compression-and-decompression-using-PCA-implemented-from-scratch>

<https://ieeexplore.ieee.org/document/9047014>

LAMPIRAN

GitHub : <https://github.com/Izanahdanur/Projek-Kompresi-Gambar>