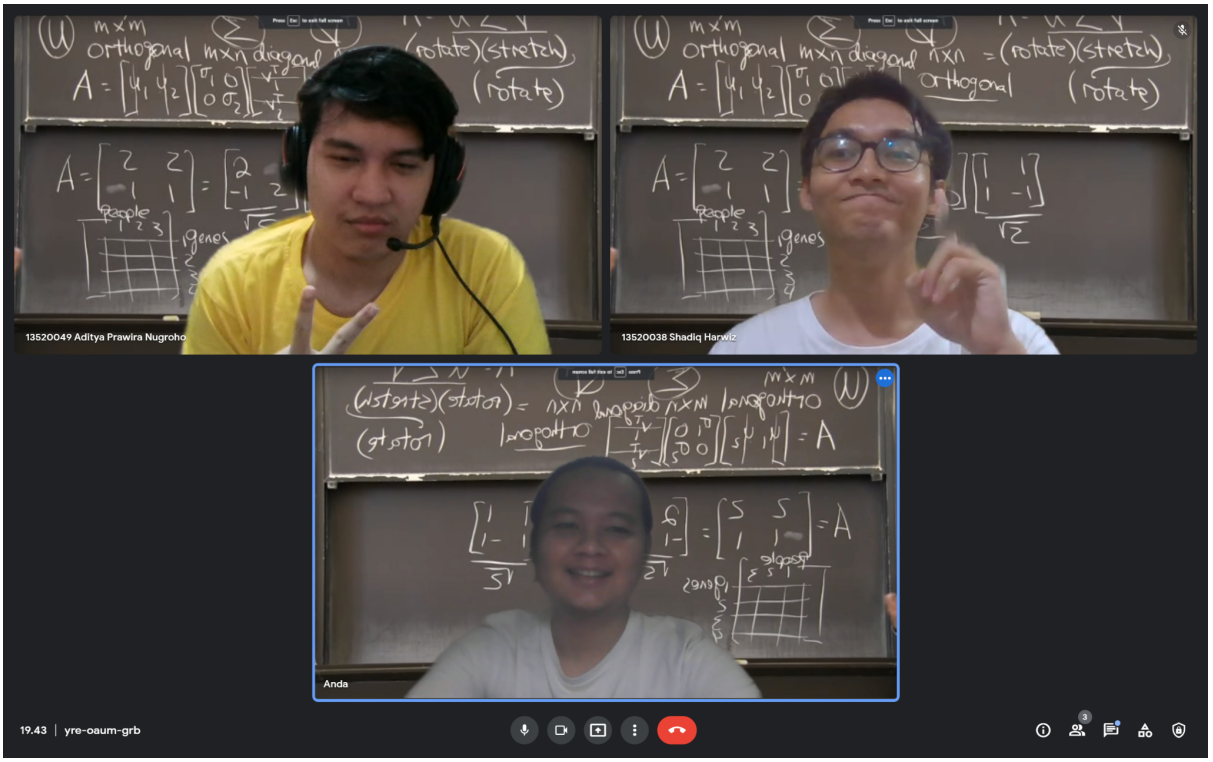
Tugas Besar 2 IF 2123 Aljabar Linier dan Geometri: Aplikasi Nilai Eigen dan Vektor Eigen dalam Kompresi Gambar



Laporan ini Dibuat sebagai Tugas Besar II Mata Kuliah IF2123 Aljabar dan Geometri

oleh

Shadiq Harwiz	13520038
Aditya Prawira Nugroho	13520049
Rizky Akbar Asmaran	13520111

TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

KOTA BANDUNG

TAHUN 2020

BAB I

DESKRIPSI MASALAH

Gambar adalah suatu hal yang sangat dibutuhkan pada dunia modern ini. Kita seringkali berinteraksi dengan gambar baik untuk mendapatkan informasi maupun sebagai hiburan. Gambar digital banyak sekali dipertukarkan di dunia digital melalui file-file yang mengandung gambar tersebut. Seringkali dalam transmisi dan penyimpanan gambar ditemukan masalah karena ukuran file gambar digital yang cenderung besar.

Kompresi gambar merupakan suatu tipe kompresi data yang dilakukan pada gambar digital. Dengan kompresi gambar, suatu file gambar digital dapat dikurangi ukuran filenya dengan baik tanpa mempengaruhi kualitas gambar secara signifikan. Terdapat berbagai metode dan algoritma yang digunakan untuk kompresi gambar pada zaman modern ini.

Salah satu algoritma yang dapat digunakan untuk kompresi gambar adalah algoritma SVD (Singular Value Decomposition). Algoritma SVD didasarkan pada teorema dalam aljabar linier yang menyatakan bahwa sebuah matriks dua dimensi dapat dipecah menjadi hasil perkalian dari 3 sub-matriks yaitu matriks ortogonal U, matriks diagonal S, dan transpose dari matriks ortogonal V. Dekomposisi matriks ini dapat dinyatakan sesuai persamaan berikut.

Matriks U adalah matriks yang kolomnya terdiri dari vektor eigen ortonormal dari matriks AA^T. Matriks ini menyimpan informasi yang penting terkait baris-baris matriks awal, dengan informasi terpenting disimpan di dalam kolom pertama. Matriks S adalah matriks diagonal yang berisi akar dari nilai eigen matriks U atau V yang terurut menurun. Matriks V adalah matriks yang kolomnya terdiri dari vektor eigen ortonormal dari matriks A^TA. Matriks ini menyimpan informasi yang penting terkait kolom-kolom matriks awal, dengan informasi terpenting disimpan dalam baris pertama.

Gambar 1. Ilustrasi Algoritma SVD dengan rank k

Dapat dilihat di gambar di atas bahwa dapat direkonstruksi gambar dengan banyak *singular values* **k** dengan mengambil kolom dan baris sebanyak k dari U dan V serta *singular value* sebanyak **k** dari S atau Σ terurut dari yang terbesar. Kita dapat mengaproksimasi suatu gambar yang mirip dengan gambar aslinya dengan mengambil **k** yang jauh lebih kecil dari jumlah total *singular value* karena kebanyakan informasi

disimpan di *singular values* awal karena singular values terurut mengecil. Nilai **k** juga berkaitan dengan rank matriks karena banyaknya *singular value* yang diambil dalam matriks S adalah *rank* dari matriks hasil, jadi dalam kata lain **k** juga merupakan rank dari matriks hasil. Maka itu matriks hasil rekonstruksi dari SVD akan berupa informasi dari gambar yang terkompresi dengan ukuran yang lebih kecil dibanding gambar awal.

Pada kesempatan kali ini, kami ditugaskan untuk membuat sebuah program berupa website kompresi gambar sederhana dengan menggunakan algoritma SVD dengan Spesifikasi website adalah sebagai berikut:

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

BAB II

TEORI SINGKAT

A. Perkalian Matriks

1. Perkalian Dua Buah Matriks

Dua buah matriks dapat dikalikan jika jumlah kolom matriks pertama sama dengan jumlah baris matriks kedua. Misalkan $A = [a_{ij}]$ adalah matriks *m x n* dan $B = [b_{ij}]$ adalah matriks *n x r*. Maka, perkalian *A* dan *B*, dilambangkan dengan *AB*, menghasilkan matriks $C = [c_{ij}]$ yang berukuran *m x r*, yang dalam hal ini

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj}$$

dalam perkalian dua buah matriks terdapat beberapa sifat yang perlu diperhatikan seperti berikut:

1. Perkalian matriks tidak bersifat komutatif, yaitu $AB \neq BA$.
2. Hukum asosiatif berlaku pada operasi matriks: $(AB)C = A(BC)$
3. Hukum distributif berlaku pada operasi matriks:
   - *A(B + C) = AB + AC (hukum distributif kiri)*
   - *(B + C)A = BA + CA (hukum distributif kanan)*
4. Perkalian matriks dengan matriks identitas I tidak mengubah matriks, yaitu AI = IA = A.
5. Perpangkatan matriks didefinisikan sebagai berikut: $A^0 = I$
6. A adalah matriks ortogonal jika $AA^T = A^T A = I$.

2. Perkalian Matriks dengan Skalar

Misalkan k adalah sebuah skalar. Perkalian matriks A dengan skalar k adalah mengalikan setiap elemen matriks dengan k.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$kA = \begin{bmatrix} ka_{11} & ka_{12} & \cdots & ka_{1n} \\ ka_{21} & ka_{22} & \cdots & ka_{2n} \\ \vdots & \vdots & & \vdots \\ ka_{m1} & ka_{m2} & \cdots & ka_{mn} \end{bmatrix}$$

B. Nilai Eigen & Vektor Eigen

Sebuah matriks persegi $n \times n$ memiliki nilai dan vektor karakteristik yang lebih sering disebut sebagai nilai dan vektor eigen. Berikut diberikan definisi nilai dan vektor eigen dari suatu matriks. Definisi 1 [3] Jika A adalah sebuah matriks $n \times n$, maka sebuah vektor tak nol $\mathbf{v}$ pada $\boldsymbol{R}$ n disebut vektor eigen (eigenvector) dari A jika $A\mathbf{v}$ adalah sebuah kelipatan skalar dari $\mathbf{v}$ yaitu

$$A\mathbf{v} = \lambda\mathbf{v}$$

untuk sebarang skalar λ. Skalar λ disebut nilai eigen (eigenvalue) dari A, dan $\mathbf{v}$ disebut sebagai vektor eigen dari A yang bersesuaian dengan λ. Untuk memperoleh nilai eigen dari sebuah matriks A berukuran $n \times n$, dapat ditulis sebagai $A\mathbf{v} = \lambda I\mathbf{v}$ atau ekuivalen dengan

$$(\lambda I - A)\mathbf{v} = \mathbf{0} \quad \text{.....(1)}$$

agar λ dapat menjadi nilai eigen, harus terdapat paling sedikit satu solusi tak nol dari Persamaan 1. Namun, Persamaan 1 memiliki solusi tak nol jika dan hanya jika

$$det(\lambda I - A) = 0 \quad \text{.....(2)}$$

Persamaan 2 disebut persamaan karakteristik matriks A. Skalar-skalar yang memenuhi persamaan tersebut adalah nilai-nilai eigen matriks A.

C. Matriks SVD (Singular Value Decomposition)

Metode *Singular Value Decomposition* (SVD) adalah salah satu teknik dalam analisis numerik yang digunakan untuk "mendiagonalkan" matriks. Dalam sudut pandang pengolahan citra, *singular value* dari suatu citra memiliki stabilitas yang baik, dimana ketika ada sedikit gangguan diberikan pada citra tersebut, *singular value* tidak berubah secara signifikan. Keuntungan lain adalah ukuran matriks dari transformasi metode SVD tidak tetap dan dapat berupa persegi. Kemudian *singular value* mengandung informasi properti persamaan linear citra.

Dekomposisi nilai singular atau yang lebih dikenal sebagai *Singular Value Decomposition* (SVD) adalah salah satu teknik dekomposisi yang cukup terkenal. SVD berkaitan erat dengan nilai singular dari sebuah matriks yang merupakan salah satu karakteristik matriks.

Suatu matriks misalkan kita namai matriks A dengan nilai eigen dari matriks A^TA yaitu $\boldsymbol{\lambda}_i$ untuk setiap $1 \leq i \leq n$ dengan n yaitu banyak nilai eigen, maka *nilai singular* matriks A yaitu $\sigma_i = \sqrt[2]{\lambda_i}$ dan v_i merupakan vektor eigen matriks A^TA yang bersesuaian dengan nilai $\boldsymbol{\lambda}_i$.

secara umum algoritma *Singular Value Decomposition* (SVD) adalah sebagai berikut:

input berupa sebuah matriks A dan akan mengeluarkan output berupa matriks U, V, dan matriks singular S sehingga $A = USV^T$. Dengan langkah langkah berikut ini :

1. dibentuk matriks A^TA dengan nilai eigen $\boldsymbol{\lambda}_i$ untuk setiap $1 \leq i \leq n$ maka nilai singular matriks A^TA yaitu $\sigma_i = \sqrt[2]{\lambda_i}$
2. dibentuk matriks diagonal S = $\begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_n \end{bmatrix}$
3. dicari himpunan vektor eigen dari matriks A^TA misalkan $\{v_1, v_2, \dots v_n\}$ merupakan vektor-vektor eigen matriks A^TA dengan v_i merupakan vektor eigen yang bersesuaian dengan nilai $\boldsymbol{\lambda}_i$.
4. dibentuk matriks ortogonal $V = [v_1, v_2, \dots . v_n]$
5. dibentuk himpunan vektor $U = \{u_1, u_2, \dots . u_n\}$ dengan $u_1 = \frac{1}{\sigma i} Avi$ untuk setiap $1 \leq i \leq n$
6. dibentuk matriks ortogonal U = $[u_1, u_2, \dots . u_n]$
7. dibentuk dekomposisi SVD berupa $A = USV^T$

Dekomposisi tersebut disebut sebagai *Singular Value Decomposition*(SVD). Nilai $a_1 \ldots a_n$ dari S disebut nilai-nilai singular dari A, kolom-kolom dari U merupakan vektor-vektor singular kiri dari A dan kolom-kolom dari V merupakan Vektor-vektor singular kanan dari A.

BAB III

IMPLEMENTASI PROGRAM

Implementasi Program Kompresi Gambar dilakukan dalam bahasa pemrograman *python* untuk bagian *back-end* dan bahasa pemrograman HTML, CSS, Javascript, FastApi untuk bagian *front-end.* Fungsi-fungsi dalam program dipisahkan dalam beberapa berkas terpisah berbentuk modul *python*. Terdapat beberapa modul program yang diimplementasikan dalam bahasa pemrograman *python* lalu di-import ke program utama. Modul-modul yang digunakan yaitu

BackEnd Modules:

1. FungsiEigen

   Modul ini digunakan untuk membuat fungsi yang mengembalikan vektor eigen dan nilai-nilai eigen dari suatu matriks A yang berukuran n x n. Pencarian nilai eigen dan vektor eigen menggunakan algoritma qr, dan kemudian dilakukan *power iteration* sebanyak 20 kali. Secara singkat, algoritma ini akan mendekomposisikan matriks n x n tersebut menjadi matriks q dan r. Matriks q adalah matriks yang ortogonal, sehingga kolom matriks q adalah vektor yang ortogonal dengan kolom lainnya, dan matriks r adalah matriks segitiga atas. Kemudian, nilai eigen matriksnya didapatkan dari hasil kali matriks r dengan q. Lalu, hasil perkaliannya didekomposisi lagi menjadi matriks q1 dan r1 yang baru, dan kita kalikan lagi matriks r1 dengan q1. Cara mendapatkan vektor eigen dari matriks A adalah mengalikan q dengan q1, dan seterusnya . Ulangi langkah tersebut sebanyak 20 kali sehingga perhitungan menjadi lebih akurat.

Function	Deskripsi
def eigenValues(matrix)	Mengembalikan nilai eigen dari input user yang berupa matrix dengan menggunakan QR Algorithm

2. FungsiSVD

   Modul ini digunakan untuk mendapatkan matriks U, S (sigma), dan V. Misalkan kita memiliki matriks A berukuran m x n, kita dapat mendapatkan matriks V dengan cara mengalikan A^T dengan A, kemudian mencari nilai eigen dan vektor eigen dari hasil perkaliannya. Setelah kita mendapatkan nilai eigen dari matriks V, kita matriks S (sigma) dengan ukuran yang sama dengan matriks A, yaitu m x n, dan kita isi diagonal utama matriks S dengan akar dari nilai eigen. Setelah itu, kita bisa mendapatkan kolom ke-i matriks U dengan $U_i = AV_i/sigma_i$, dengan V_i adalah baris ke-i dari matriks V yang ditranspos dan $sigma_i$ adalah nilai singular pada diagonal utama matriks S.

Function	Deskripsi
def matriksVdanS(A)	Mengembalikan nilai matrix V, dan matriks S (sigma) atau singular values
def MatriksU(A, V, S, EigVal)	mengembalikan matriks U
def compress(matriks, percentage)	Mengembalikan perhitungan matriks yang sudah di dekomposisi dengan cara SVD, kemudian disatukan kembali dengan k (rank) yang ditentukan.

3. FungsiCompress

   Implementasi modul ini ada dua, yaitu untuk kompres gambar berekstensi “png” dan selain “png”. Fungsi kompres ini menggunakan library PIL dalam python untuk mengekstrak matriks RGBA untuk gambar “png” dan RGB untuk format lainnya. Fungsi ini akan mengkompres gambar, sekaligus menyimpan lama waktu yang digunakan untuk mengkompres gambar.

Function	Deskripsi
def Compress_png(filepath, percentage, extension)	Fungsi akan mengembalikan gambar berekstensi png yang sudah dikompres sesuai percentage yang diinput.
def Compress_img(filepath, percentage, extension)	Fungsi akan mengembalikan gambar berekstensi selain png yang sudah dikompres sesuai percentage yang diinput.

4. Main

   Module main digunakan untuk menggabungkan kedua fungsi kompres pada modul FungsiCompress. Dalam program main akan dibaca ekstensi dari gambar. Setelah itu, program akan memanggil fungsi kompres sesuai dengan ekstensi dari gambar.

Function	Deskripsi
def start(filepath, percentage)	Fungsi ini menyatukan semua fungsi compress dan membaca ekstensi file yang diunggah.

5. NumPy

   Modul numpy digunakan untuk melakukan operasi pada matriks serta dekomposisi qr matriks. Operasi-operasi yang dilakukan pada matriks antara lain perkalian matriks, perkalian skalar, transpose matriks. Selain itu, numpy juga digunakan untuk melakukan operasi matematika seperti akar (square root). Modul ini juga digunakan untuk mencari dimensi matriks, seperti baris, kolom, dan dimensinya, lalu mengkonversi matriks menjadi array pada numpy.

FrontEnd Modules:

1. FASTApi

   Modul FastApi digunakan sebagai framework yang menghubungkan back-end dan front-end dari web. Modul ini digunakan supaya web mampu berjalan di local host dengan memanfaatkan class Request. Modul ini juga memanggil template html dengann memanfaatkan class Jinja2Templates serta javascript dari folder lain dengan memanfaatkan class UploadFile dan File.

2. HTML & Javascript

   Modul HTML ini digunakan sebagai struktur dari website image compressor. Modul ini dikombinasikan dengan Javascript sehingga dapat menerima input gambar dari user, menyimpan gambar dari user, dan mendownload hasil dari *compression.*

3. CSS

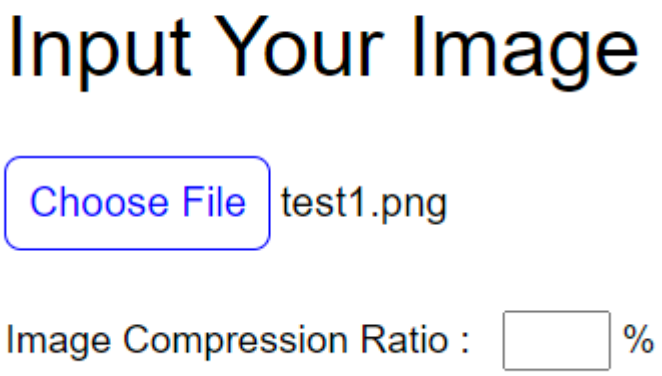
   Modul CSS ini digunakan untuk memperindah atau mempercantik tampilan dari website sederhana Image Compressor ini dengan mengatur tata letak dan pemilihan warna.

BAB IV

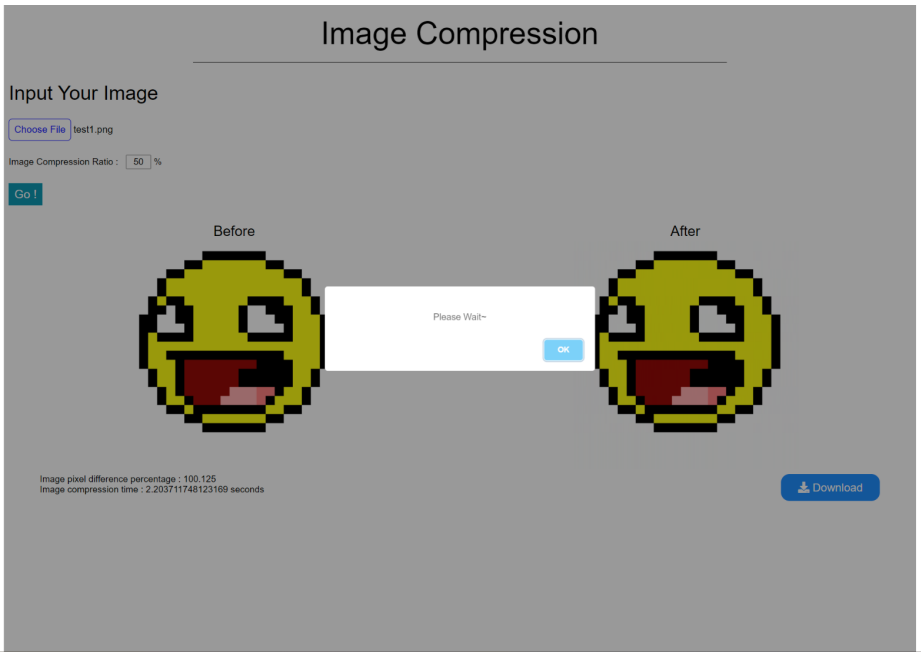
EKSPERIMEN

1. Tampilan Website

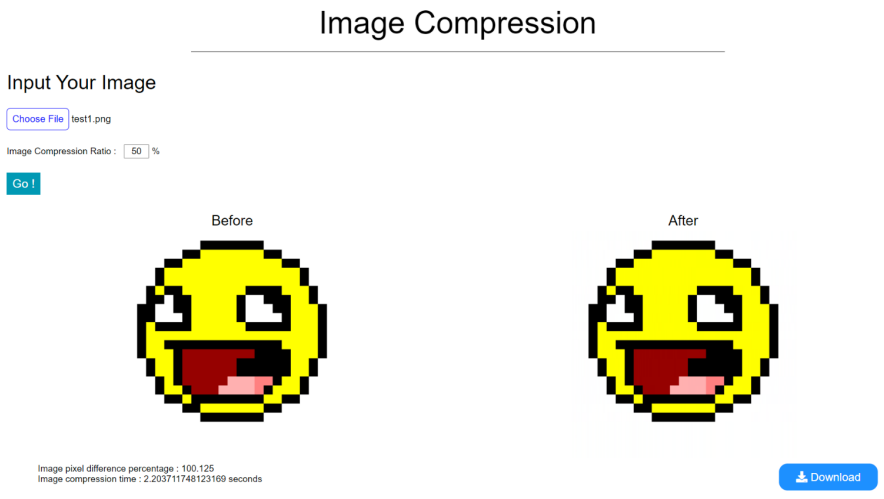
2. Upload Gambar



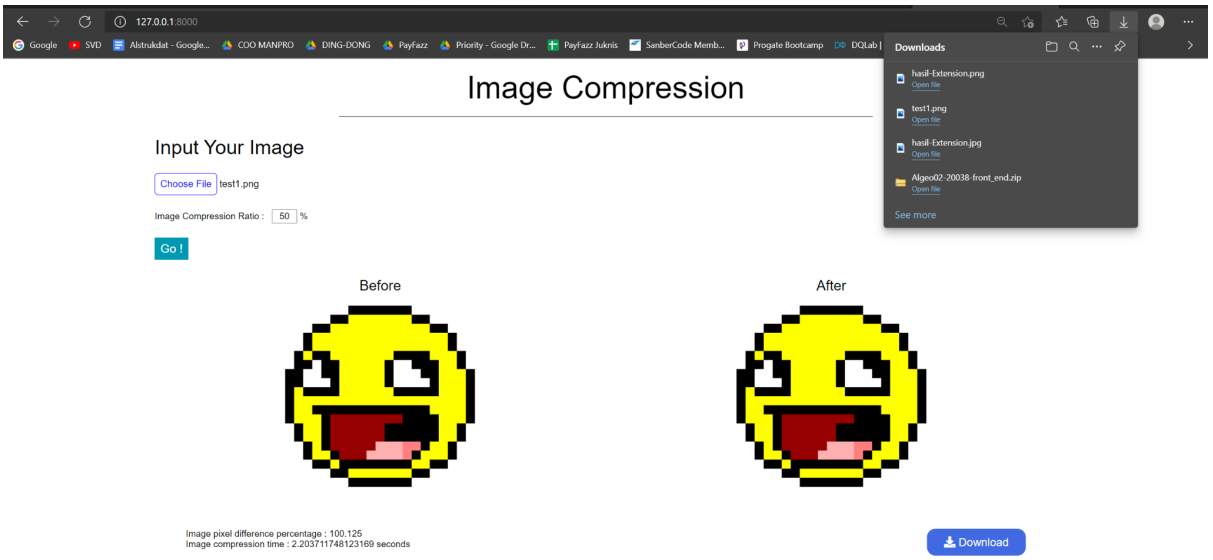
3. Proses Compress



4. Hasil Compress



5. Pengunduhan Hasil Kompresi



Before & After Compression:

	
Sebelum	**Sesudah**

BAB V

KESIMPULAN, SARAN & REFLEKSI

Kesimpulan

Kami telah membuat sebuah program *Image Compressor* berbasis *Website* yang dapat mengkompres ukuran pixel dari suatu gambar atau foto yang dimasukkan oleh pengguna menggunakan konsep dari nilai Eigen, vektor Eigen dan SVD (*Singular Value Decomposition*).

Saran

Kami menyadari jika program yang dibuat masih memiliki banyak kekurangan dan jauh dari kata sempurna. oleh karena itu, saran untuk pengembangan program ini kedepannya adalah dengan lebih mengembangkan algoritma terkait *Singular Value Decomposition* yang membutuhkan waktu yang cukup banyak agar nantinya hasil foto yang dikompres akan lebih cepat dan efisien.

Refleksi

Dalam mengerjakan tugas besar ini, terdapat beberapa kendala. Yang pertama, kurangnya pemahaman kami terhadap mengimplementasikan konsep SVD (*Singular Value Decomposition*) ke dalam bentuk pemrograman berbasis python, sehingga membutuhkan waktu yang lumayan lama untuk *trial & error* dalam mengimplementasikan konsep SVD tersebut. Kedua, kami kesulitan dalam melakukan integrasi terhadap frontend dan backend sehingga cukup memakan waktu yang cukup lama.

Kesalahan tersebut murni datangnya dari kekurangan kami, sehingga kami akan lebih serius dan belajar lebih giat kembali agar hal-hal yang sebelumnya menjadi hambatan kami dapat teratasi dengan baik.

PEMBAGIAN TUGAS

No	BAGIAN	PJ	STATUS
1	Fungsi SVD & Image Compression	Aditya Prawira Nugroho	DONE
2	Website	Shadiq Harwiz	DONE
3	Fungsi OpenImg & Laporan	Rizky Akbar Asmaran	DONE

Note:

Backend menggunakan Python

Frontend menggunakan HTML, CSS, JS. dan FrameWork FastApi

REFERENSI

"Nilai Eigen dan Vektor Eigen Bagian 1" by Rinaldi Munir
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2020-2021/Algeo-18- Nilai-Eigen-dan-Vektor-Eigen-Bagian1.pdf

"Image Compression using Singular Value Decomposition (SVD)"
http://www.math.utah.edu/~goller/F15_M2270/BradyMathews_SVDImage.pdf

"Understanding Singular Value Decomposition and its Application in Data Science" by Reza Bagheri
https://towardsdatascience.com/understanding-singular-value-decomposition-and-its application-in-data-science-388a54be95d

"The Singular Value Decomposition"
https://math.mit.edu/~gs/linearalgebra/linearalgebra5_7-1.pdf

"Image Compression with Singular Value Decomposition"
http://timbaumann.info/svd-image-compression-demo/

"Developing Web Applications with Flask"
https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/Python3_Flask.ht ml

"How To Create a React + Flask Project"
https://blog.miguelgrinberg.com/post/how-to-create-a-react--flask-project

"Orthogonal Bases and the QR Algorithm"
https://www-users.cse.umn.edu/~olver/aims_/qr.pdf

"MIT 18.065 Matrix Methods in Data Analysis, Signal Processing, and Machine Learning, Spring 2018"
https://www.youtube.com/watch?v=rYz83XPxiZo