

LAPORAN TUGAS BESAR 2

IF2123 ALJABAR LINEAR DAN GEOMETRI



Disusun oleh:

Fadil Fauzani	13520032
Gibran Darmawan	13520061

TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
SEMESTER 1 TAHUN 2021/2022

Daftar Isi

LAPORAN TUGAS BESAR 2	1
IF2123 ALJABAR LINEAR DAN GEOMETRI	1
BAB I	3
1.1 Membuat website kompresi gambar	3
1.2 Penggunaan Program	3
BAB II	5
2.1 Perkalian Matriks	5
2.2 Nilai Eigen dan Vektor Eigen Matriks	5
2.3 Matriks SVD	5
BAB III	6
3.1 Web-stack	7
3.2 Kompresi Gambar	7
BAB IV	9
4.1 Percobaan dengan $k = 20$	9
4.2 Percobaan dengan $k = 25$	9
4.3 Percobaan dengan $k = 50$	10
4.4 Percobaan dengan $k = 75$	10
BAB V	12
5.1 Kesimpulan	12
5.2 Saran	12
DAFTAR REFERENSI	13

BAB I

DESKRIPSI MASALAH

1.1 Membuat website kompresi gambar

Buatlah program kompresi gambar dengan memanfaatkan algoritma SVD dalam bentuk website lokal sederhana. 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.

1.2 Penggunaan Program

Berikut ini adalah input yang akan dimasukkan pengguna untuk eksekusi program.

1. File gambar, berisi file gambar input yang ingin dikompresi dengan format file yang bebas selama merupakan format untuk gambar.

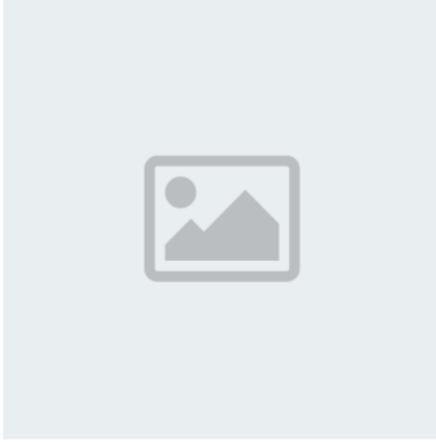
2. Tingkat kompresi, berisi tingkat kompresi dari gambar (formatnya dibebaskan, cth: Jumlah singular value yang digunakan)

Tampilan layout dari aplikasi web yang akan dibangun kurang lebih adalah sebagai berikut. Anda dapat mengubah layout selama layout masih terdiri dari komponen yang sama.

Image Compression

Input Your Image

[Choose File..](#) No File Chosen
Image Compression Rate : %

Before 

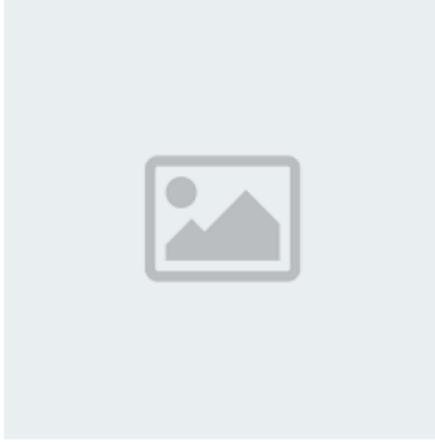
After 

Image pixel difference percentage: 50 %
Image compression time: 0.5 seconds

[Download](#) ↓

Gambar 3. Contoh tampilan layout dari aplikasi web yang dibangun.

Catatan: Warna biru menunjukkan komponen yang dapat diklik.

Anda dapat menambahkan menu lainnya, gambar, logo, dan sebagainya. Tampilan front end dari website dibuat semenarik mungkin selama mencakup seluruh informasi pada layout yang diberikan di atas. Tampilan program merupakan bagian dari penilaian.

BAB II

TEORI SINGKAT

2.1 Perkalian Matriks

Perkalian Matriks dapat dihitung dengan cara berikut:

$$\text{Misal } C_{m \times n} = A_{m \times r} \times B_{r \times n}, \text{ maka } C_{ij} = A_{i1} * B_{1j} + A_{i2} * B_{2j} + \dots + A_{ir} * B_{rj}$$

Perlu dicatat, kedua matriks A dan B dapat dikalian jika dan hanya jika jumlah kolom A = jumlah kolom B

2.2 Nilai Eigen dan Vektor Eigen Matriks

Jika A adalah suatu matriks berukuran $n \times n$, x adalah vektor tak nol di R^n , λ adalah suatu nilai skalar, dan ketiganya memenuhi persamaan berikut:

$$Ax = \lambda x$$

Maka λ adalah nilai Eigen dari matrix A, dan x adalah vektor Eigen matriks A yang berkoresponden dengan λ .

Nilai Eigen suatu matriks A adalah solusi dari persamaan berikut:

$$\det(\lambda I - A) = 0$$

Sedangkan Nilai vektor Eigen yang berkoresponden dengan nilai Eigen λ dapat dicari dengan mencari solusi persamaan berikut:

$$(\lambda I - A)x = 0$$

2.3 Matriks SVD

Matriks SVD (Singular Value Decomposition) adalah salah satu cara menguraikan suatu Matriks berukuran $m \times n$ menjadi 3 bagian yang memenuhi persamaan berikut:

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

U = Matriks orthogonal berukuran $m \times m$,

Σ = Matriks berukuran $m \times n$ yang elemen diagonal utamanya adalah nilai singular dari Matriks A dan elemen-elemen lainnya adalah 0,

V = Matriks orthogonal berukuran $n \times n$

$$\begin{matrix} \mathbf{M} &= & \mathbf{U} & \Sigma & \mathbf{V}^* \\ m \times n & & m \times m & m \times n & n \times n \end{matrix}$$

Nilai singular σ dari matriks A dirumuskan sebagai berikut:

$$\sigma_i = \sqrt{\lambda_i}$$

dengan λ adalah nilai Eigen matriks $A^T A$, dan nilainya teurut mengecil ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$).

Matriks U bisa didapatkan dengan menggabungkan vektor-vektor Eigen yang telah dinormalisasi dari matriks AA^T . sedangkan Matriks V didapatkan dengan menggabungkan vektor-vektor Eigen yang telah dinormalisasi dari matriks $A^T A$.

BAB III

IMPLEMENTASI

3.1 Web-stack

Dalam pengerjaan Tugas in, kami memakai Flask untuk membantu membuat backend dan frontend dari website kami. Flask sendiri adalah salah satu framework web development yang ada di python. Kami membuat beberapa fungsi dalam back-end, yaitu:

- 1) index(), menampilkan layar utama pada website,
- 2) upload_files(), menghandle file unggahan dari pengguna,
- 3) upload(), mengirimkan hasil upload ke web,
- 4) igen(), digunakan untuk mencari Eigen value dan vektor sebuah matrix, proses menggunakan algoritma *power iteration*,
- 5) svd(), mengirimkan hasil svd matriks,
- 6) normalize(), mengirimkan vektor yang sudah di normalisasi,
- 7) compressImage(), melakukan kompresi pada gambar yang diunggah.

3.2 Library Python

Dalam Algortima kami, kami juga menggunakan beberapa library dari python untuk membantu komputasi yang kami lakukan, beberapa diantaranya adalah:

- 1) Flask, digunakan untuk membuat backend dari website kami,
- 2) Pillow, digunakan untuk membaca gambar yang akan diolah,
- 3) Numpy, digunakan untuk membantu perhitungan matriks yang dilakukan,
- 4) Time, digunakan untuk menghitung lama algoritma SVD,
- 5) Werkzeug, digunakan untuk mengubah nama file menjadi lebih aman untuk digunakan dalam program

3.2 Kompresi Gambar

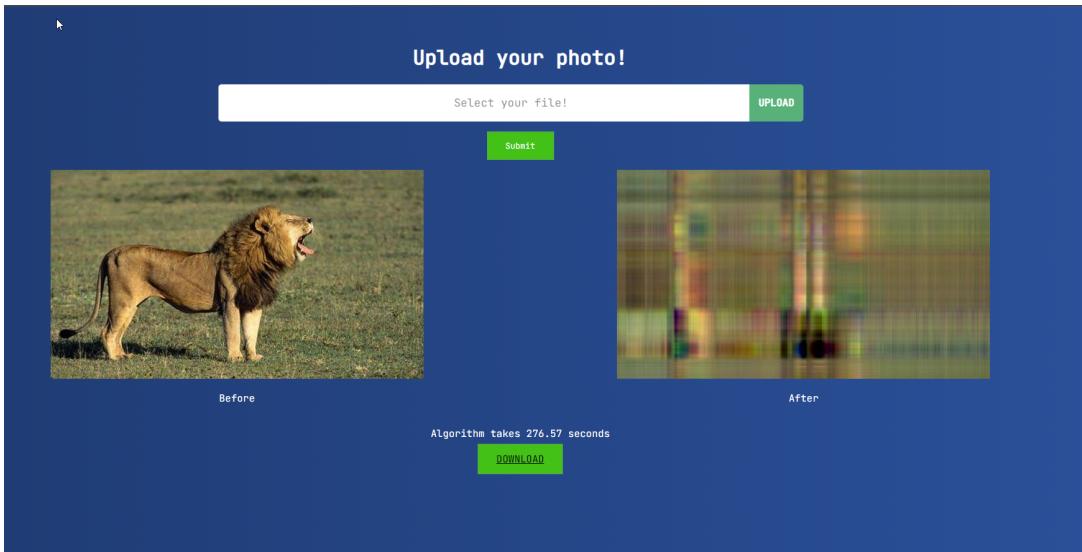
Kompresi gambar dilakukan oleh program python. *Library* python PIL mengkonversi file gambar menjadi sebuah data dalam bentuk array untuk diproses dalam algoritma. Data

gambar tersebut dibagi menjadi 3 data (channel), yaitu menjadi *channel* merah, hijau, dan biru. Data *channel* ini dibagi menjadi beberapa bagian (*chunk*) untuk mulai didekomposisi dengan SVD. Metode SVD yang digunakan yaitu *power iteration*. Hasil dari proses SVD dipotong dan diambil sesuai dengan tingkat kompresi yang diminta oleh pengguna. Tingkat kompresi yang diminta pengguna akan menyesuaikan dengan *rank* untuk menghitung SVD akhir. Hasil dekomposisi data channel kemudian disatukan kembali dengan dekomposisi yang lainnya dan menggantikan hasil dekomposisi yang lama.

BAB IV

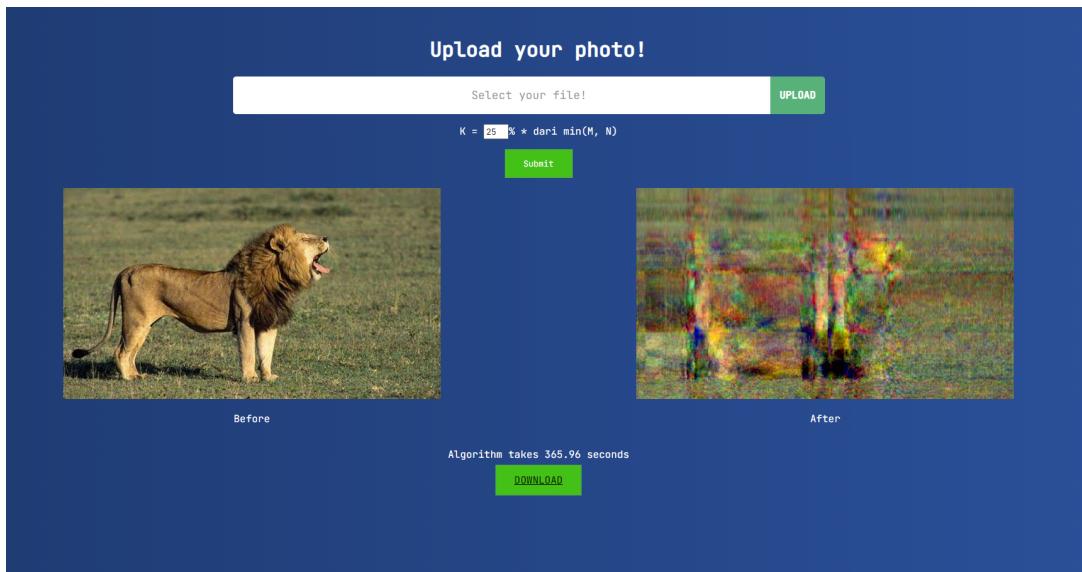
EKSPERIMEN

4.1 Percobaan dengan $k = 20$



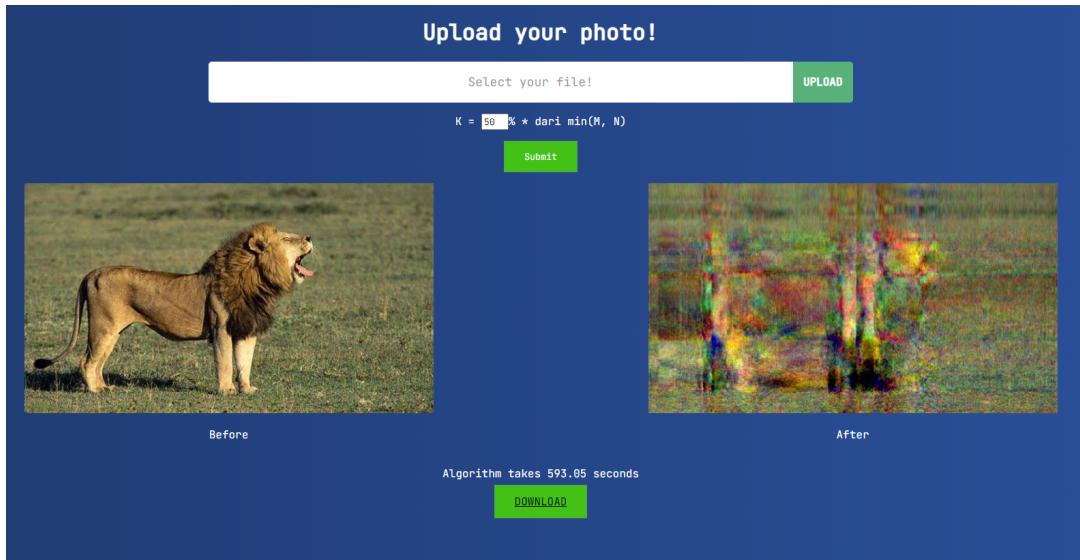
Terlihat untuk $k = 20$, didapatkan hasil diatas dalam waktu 275.57 detik. Hasil gambar sudah tidak jelas (ini mungkin karena kesalahan dalam perhitungan SVD)

4.2 Percobaan dengan $k = 25$



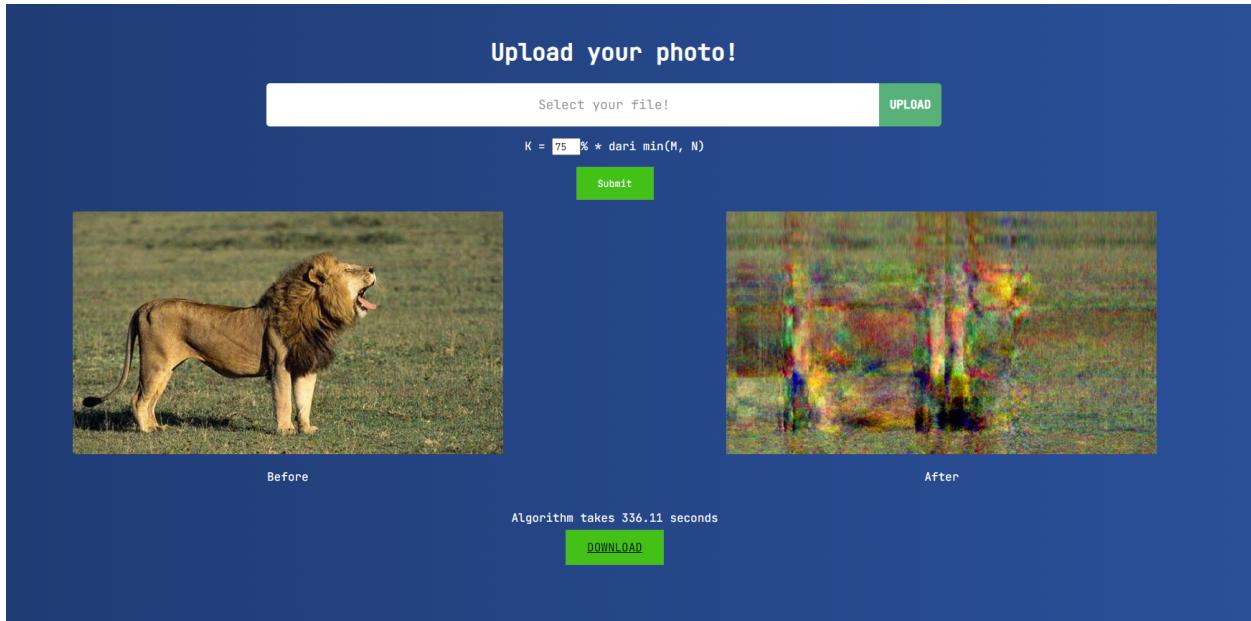
Terlihat untuk $k = 25$, didapatkan hasil diatas dalam waktu 365.96 detik. Hasil gambar sudah tidak jelas (ini mungkin karena kesalahan dalam perhitungan SVD)

4.3 Percobaan dengan $k = 50$



Terlihat untuk $k = 50$, didapatkan hasil diatas dalam waktu 593.05 detik. Hasil gambar sudah tidak jelas (ini mungkin karena kesalahan dalam perhitungan SVD)

4.4 Percobaan dengan $k = 75$



Terlihat untuk $k = 75$, didapatkan hasil diatas dalam waktu 365.96 detik. Hasil gambar sudah tidak jelas (ini mungkin karena kesalahan dalam perhitungan SVD)

BAB V

KESIMPULAN, SARAN

5.1 Kesimpulan

Aplikasi kompresi gambar ini berhasil dibuat dengan mengimplementasi algoritma SVD untuk algoritma program utamanya. Dengan mengikuti mata kuliah Aljabar Linear dan Geometri, kelompok kami berhasil mewujudkan tujuan program ini. Contoh hal yang diimplementasikan pada program ini adalah:

- a. Menggunakan SVD pada matriks isi gambar.
- b. Menentukan nilai eigen, basis eigen, dan vektor eigen.
- c. Menggunakan *library* python seperti numpy, PI.
- d. Menggunakan Flask untuk merealisasikan bagian front-end dan back-end program ini.

Dari hal yang diimplementasikan di atas, tujuan dari tugas besar ini terpenuhi sesuai spesifikasinya walaupun dengan kekurangan yang ada.

5.2 Saran

Aplikasi berbasis web yang kami buat masih banyak kekurangan, salah satunya adalah durasi yang dibutuhkan untuk melakukan kompresi gambar terlalu lama dan kurang tepat. Hal ini disebabkan kekurangan dari kami yang membuat algoritma yang kurang efisien dan memiliki galat yang besar sehingga membutuhkan waktu yang banyak dan kurang tepat. Untuk peningkatan mungkin bisa menggunakan algoritma yang lebih advance seperti yang diimplementasikan pada SVD di library numpy.

DAFTAR REFERENSI

“Image Compression with Singular Value Decomposition”

<https://aaronschlegel.me/image-compression-singular-value-decomposition.html>

“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>

“Understanding Singular Value Decomposition and its Application in Data Science”

<https://towardsdatascience.com/understanding-singular-value-decomposition-and-its-application-in-data-science-388a54be95d>

“Image Compression with Singular Value Decomposition (SVD)”

http://www.math.utah.edu/~goller/F15_M2270/BradyMathews_SVDImage.pdf

Sumber css

<https://bashooka.com/coding/css-javascript-file-upload-examples/>

“Uploading Files”

<https://flask.palletsprojects.com/en/2.0.x/patterns/fileuploads/>