

LAPORAN TUGAS BESAR 2

APLIKASI NILAI EIGEN DAN VEKTOR EIGEN DALAM KOMPRESI GAMBAR

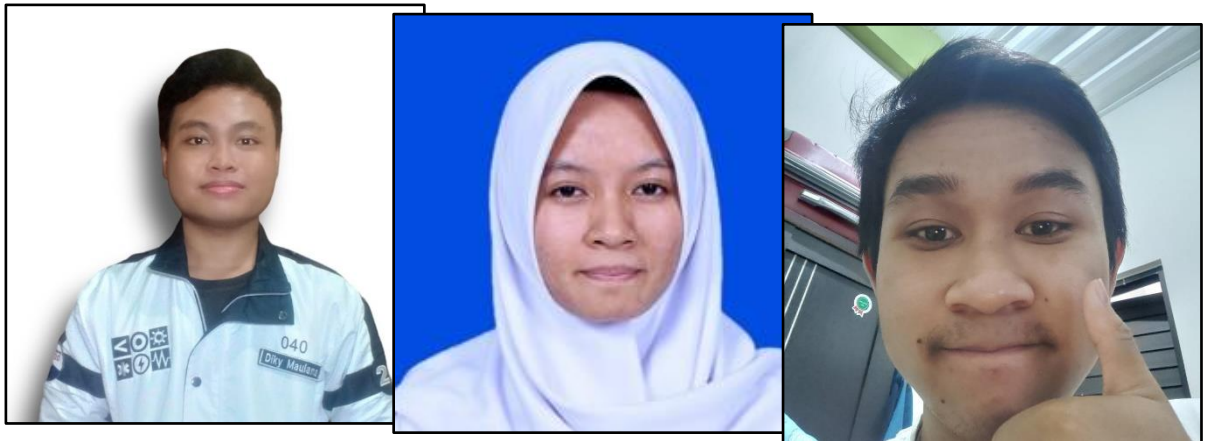
Mata Kuliah IF2123 Aljabar Linier dan Geometri

Dosen Pengampu:

Dr. Judhi Santoso

Dr. Rinaldi Munir

Dr. Rila Mandala



Kelompok 46 – Kompres Air Hangat

Nama Anggota:

Diky Restu Maulana 13520017

Hilya Fadhilah Imania 13520024

Fikri Ihsan Fadhiilah 13520148

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

DAFTAR ISI

DAFTAR ISI	2
BAB I DESKRIPSI MASALAH.....	3
1.1. Abstraksi.....	3
1.2. Penggunaan Program.....	4
1.3. Spesifikasi Program.....	5
BAB II TEORI SINGKAT	7
2.1. Perkalian Matriks.....	7
2.2. Nilai Eigen dan Vektor Eigen.....	7
2.3. QR Decomposition	7
2.4. Matriks SVD.....	8
BAB III IMPLEMENTASI	9
3.1. Algoritma SVD.....	10
3.2. Kompresi Gambar	10
BAB IV EKSPERIMEN.....	11
4.1. Gambar PNG	11
4.2. Gambar JPG.....	12
4.3. Gambar PNG Transparan	14
BAB V KESIMPULAN, SARAN, DAN REFLEKSI	16
5.1. Kesimpulan.....	16
5.2. Saran	16
5.3. Refleksi.....	16
REFERENSI.....	18

BAB I

DESKRIPSI MASALAH

1.1. Abstraksi

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.



Three levels of JPG compression. The left-most image is the original. The middle image offers a medium compression, which may not be immediately obvious to the naked eye without closer inspection. The right-most image is maximally compressed.

Gambar 1. Contoh kompresi gambar dengan berbagai tingkatan

Sumber : Understanding Compression in Digital Photography (lifewire.com)

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.

$$A_{m \times n} = U_{m \times m} S_{m \times n} V_{n \times n}^T$$

Gambar 1. Algoritma SVD

Matriks U adalah matriks yang kolomnya terdiri dari vektor eigen ortonormal dari matriks $A^T A$. 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 $A A^T$. Matriks ini menyimpan informasi yang penting terkait kolom-kolom matriks awal, dengan informasi terpenting disimpan dalam baris pertama.



Gambar 2. 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, kalian mendapatkan tantangan untuk membuat website kompresi gambar sederhana dengan menggunakan algoritma SVD.

1.2. Penggunaan Program

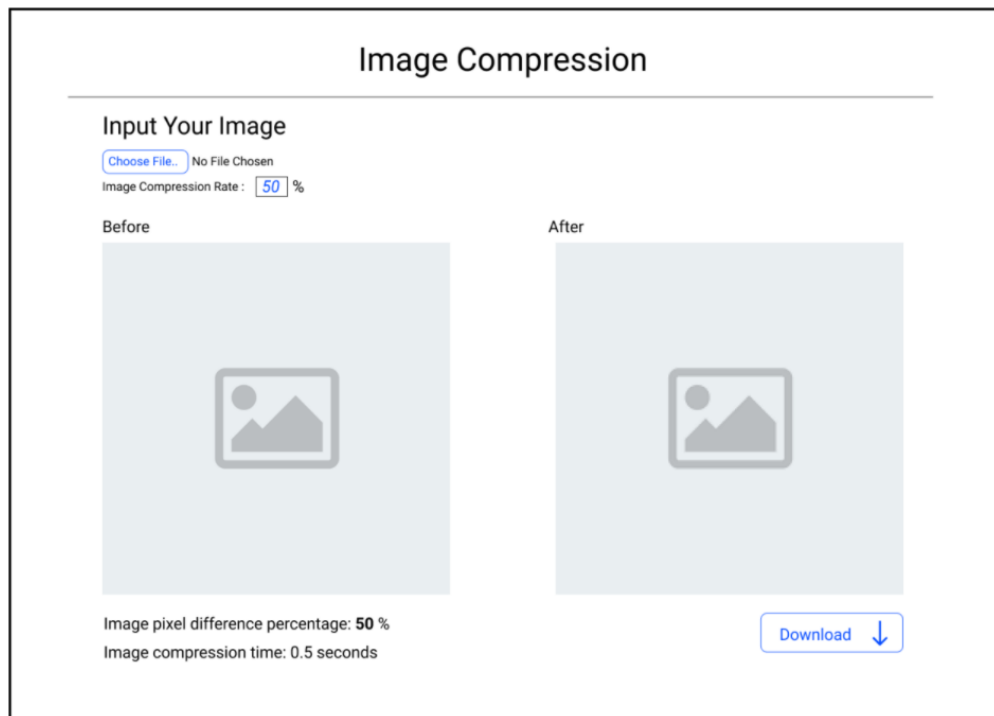
Berikut ini adalah input yang akan dimasukkan pengguna untuk eksekusi program.

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.



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

Catatan: Warna biru menunjukkan komponen yang dapat di klik.

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.

1.3. Spesifikasi Program

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.

BAB II TEORI SINGKAT

2.1. Perkalian Matriks

Perkalian dua buah matriks dilakukan sebagai berikut

$$C_{m \times n} = A_{m \times k} \times B_{k \times n}$$

Misal $A = [a_{ij}]$ dan $B = [b_{ij}]$

Maka $C = A \times B = [c_{ij}]$, $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}$

2.2. Nilai Eigen dan Vektor Eigen

Jika A adalah matriks $n \times n$, maka vector tak nol \mathbf{x} di \mathbb{R}^n disebut vector eigen dari A jika $A\mathbf{x}$ sama dengan perkalian suatu scalar λ dengan \mathbf{x} , yaitu

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

Skalar λ disebut nilai eigen dari A , dan \mathbf{x} dinamakan nilai vector eigen yang berkoresponden dengan λ . Nilai eigen menyatakan karakteristik dari sebuah matriks persegi. Vektor eigen \mathbf{x} menyatakan vector kolom yang apabila dikalikan dengan sebuah matriks persegi akan menghasilkan vector lain yang merupakan kelipatan dari vector itu sendiri. Dengan kata lain, operasi $A\mathbf{x} = \lambda\mathbf{x}$ menyebabkan vector \mathbf{x} menyusut atau memanjang dengan faktor λ dengan arah yang sama jika λ positif dan arah berkebalikan jika λ negatif.

Persamaan $\det(\lambda I - A) = 0$ disebut persamaan karakteristik dari matriks A dan akar-akar persamaan tersebut, yaitu λ dinamakan akar-akar karakteristik atau nilai-nilai eigen.

2.3. QR Decomposition

Mendekomposisi matriks artinya memfaktorkan sebuah matriks. Pada QR decomposition, sebuah matriks difaktorkan menjadi matriks Q yang merupakan matriks orthogonal dan matriks R yang merupakan upper diagonal.

$$\begin{array}{c} \mathbf{A} \\ \left[\begin{array}{c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{array} \right] \end{array} = \begin{array}{c} \mathbf{Q} \\ \left[\begin{array}{c|c|c} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{array} \right] \end{array} \begin{array}{c} \mathbf{R} \\ \left[\begin{array}{ccc} \mathbf{e}_1^T \cdot \mathbf{a}_1 & \mathbf{e}_1^T \cdot \mathbf{a}_2 & \mathbf{e}_1^T \cdot \mathbf{a}_3 \\ 0 & \mathbf{e}_2^T \cdot \mathbf{a}_2 & \mathbf{e}_2^T \cdot \mathbf{a}_3 \\ 0 & 0 & \mathbf{e}_3^T \cdot \mathbf{a}_3 \end{array} \right] \end{array}$$

Orthogonal Unit vectors
Upper Diagonal Matrix

2.4. Matriks SVD

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 $m \times n$ U, matriks diagonal S, dan transpose dari matriks $n \times n$ V. Dekomposisi matriks ini dapat dinyatakan sesuai persamaan berikut.

$$A_{m \times n} = U_{m \times m} S_{m \times n} V_{n \times n}^T$$

Matriks U adalah matriks yang kolomnya terdiri dari m vector 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 n vector eigen ortonormal dari matriks $A^T A$. Matriks ini menyimpan informasi yang penting terkait kolom-kolom matriks awal, dengan informasi terpenting disimpan dalam baris pertama.



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 $m \times n$ gambar awal.

BAB III

IMPLEMENTASI

Bahasa Pemrograman : Python, Javascript

Framework :

1. Frontend : Vue
2. Backend : Flask

Frontend :

1. App.vue
Base layout dari laman web
2. components/Main.vue
Komponen utama yang mengontrol perubahan view dari input ke hasil
3. components/FileInput.vue
Komponen untuk menginput file gambar
4. components/RateInput.vue
Komponen untuk menginput tingkat kompresi
5. components/Comparator.vue
Komponen untuk menampilkan hasil
6. services/api.js
Script untuk melakukan POST request ke backend

Backend:

1. run.py
Inisialisasi server dan routing dengan Flask dan flask_cors
2. compress/lib.py
Fungsi utama untuk melakukan kompresi gambar

Beberapa fungsi/prosedur yang kami gunakan untuk `compress/lib.py`, yaitu

```
function find_eig(input A: Matrix)→(array, Matrix)  
mengembalikan sebuah array berisi nilai eigen dan matriks yang  
kolomnya merupakan 9ector eigen  
  
function svd(input A: Matrix)→(Matrix, array, Matrix)
```

mengembalikan matriks singular kiri, array berisi nilai singular terurut mengecil, dan matriks singular kanan

function openImage(string)→(Matrix, Matrix, Matrix, Matrix)

mengembalikan matriks Red, matriks Green, matriks Blue, dan matriks PIL Image

**function compressSingleChannel(input A: Matrix, input B: Matrix)
→Matrix**

mengembalikan matriks yang sudah dikompresi menggunakan SVD

3.1. Algoritma SVD

SVD dilakukan dengan memanfaatkan QR decomposition. Kemudian, dilakukan power iteration sebanyak 100 kali untuk mendapatkan dua buah matriks. Diagonal matriks pertama akan menghasilkan nilai-nilai eigen, sedangkan matriks kedua akan menjadi matriks vektor eigen.

3.2. Kompresi Gambar

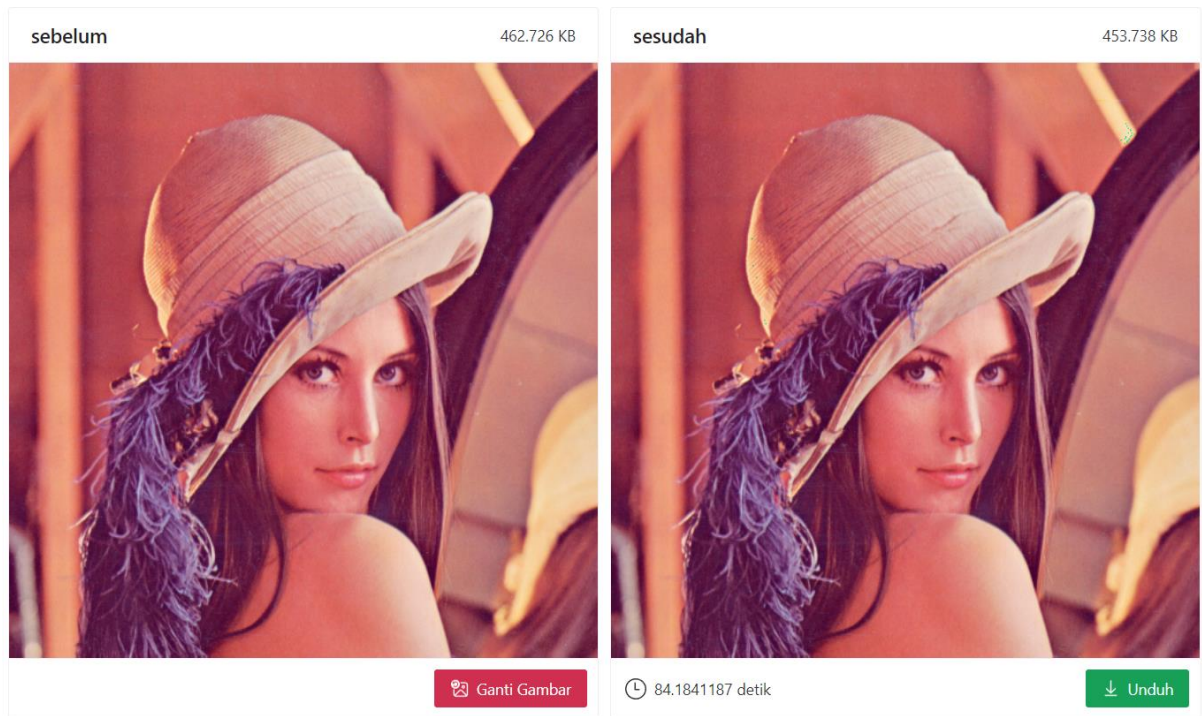
Proses kompresi diawali dengan mengubah gambar menjadi numpy array berdimensi tiga dengan memanfaatkan fungsi Image.open pada Python Image Library. Matriks yang didapatkan adalah matriks RGB. Kemudian, dilakukan pemisahan menjadi tiga buah matriks berdasarkan color channel, yaitu red, green dan blue. Program akan meminta masukan tingkat kompresi, lalu akan dihitung limit singular value yang dapat digunakan. Masing-masing matriks color channel akan dilakukan SVD dan hasilnya akan digabungkan kembali menjadi matriks RGB dengan memanfaatkan fungsi Image.merge. Akhirnya, matriks RGB tersebut akan diubah kembali menjadi gambar menggunakan prosedur show.

BAB IV

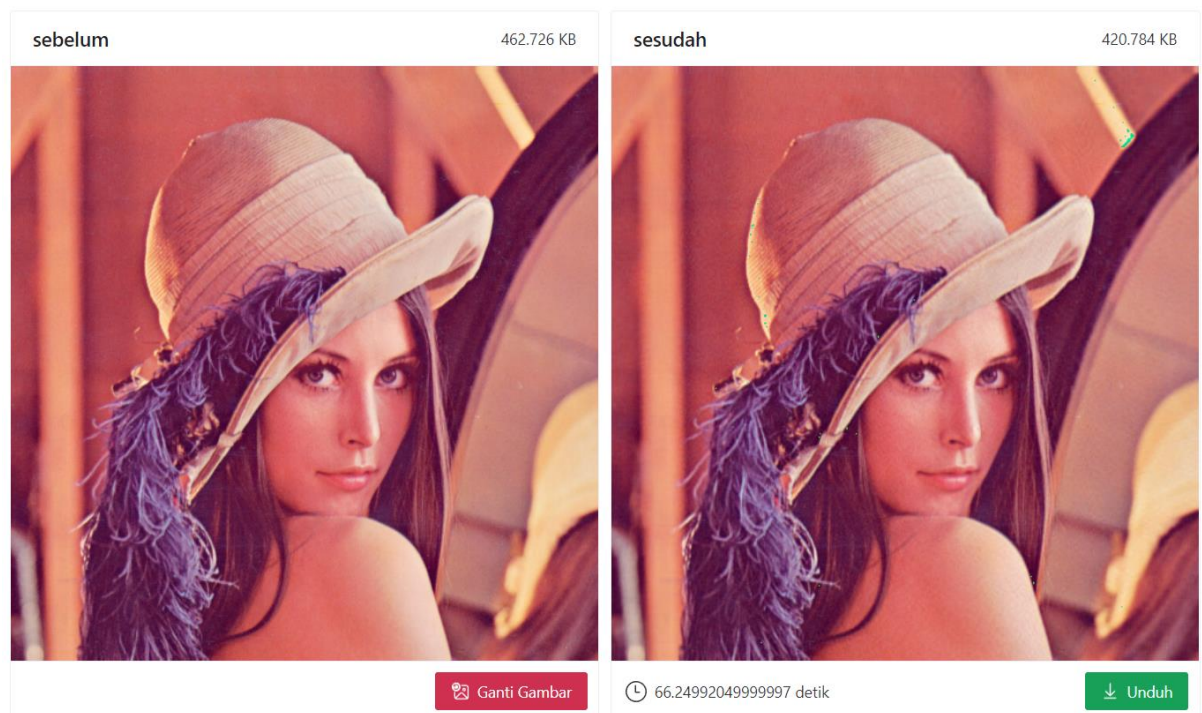
EKSPERIMEN

4.1. Gambar PNG

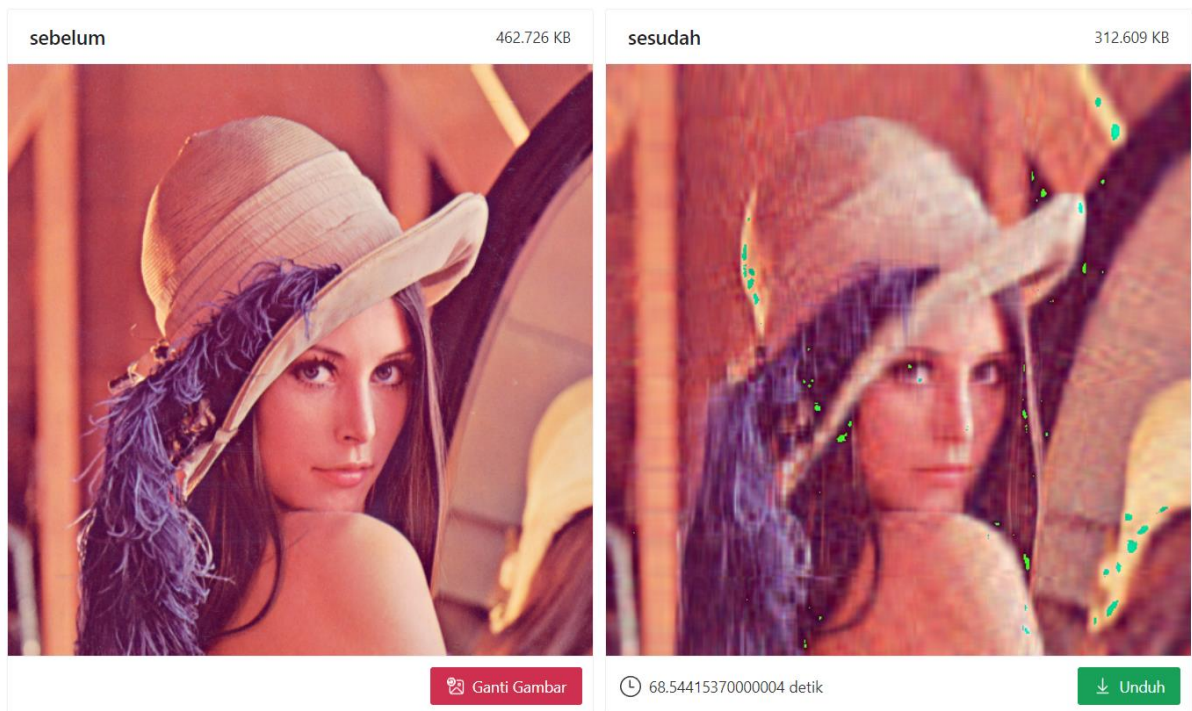
90%



50%

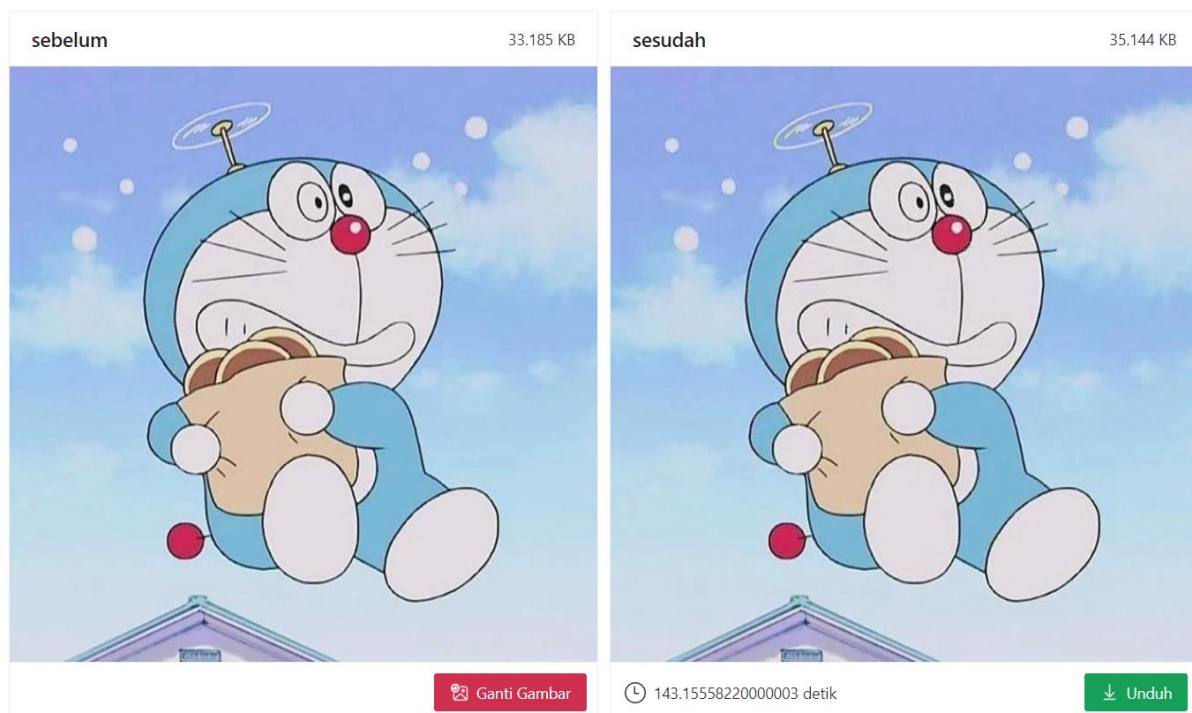


10%



4.2. Gambar JPG

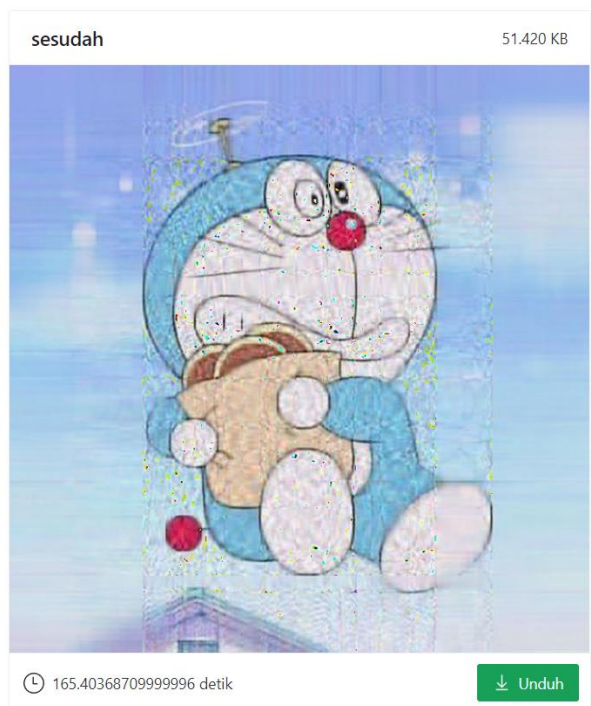
90%



50%

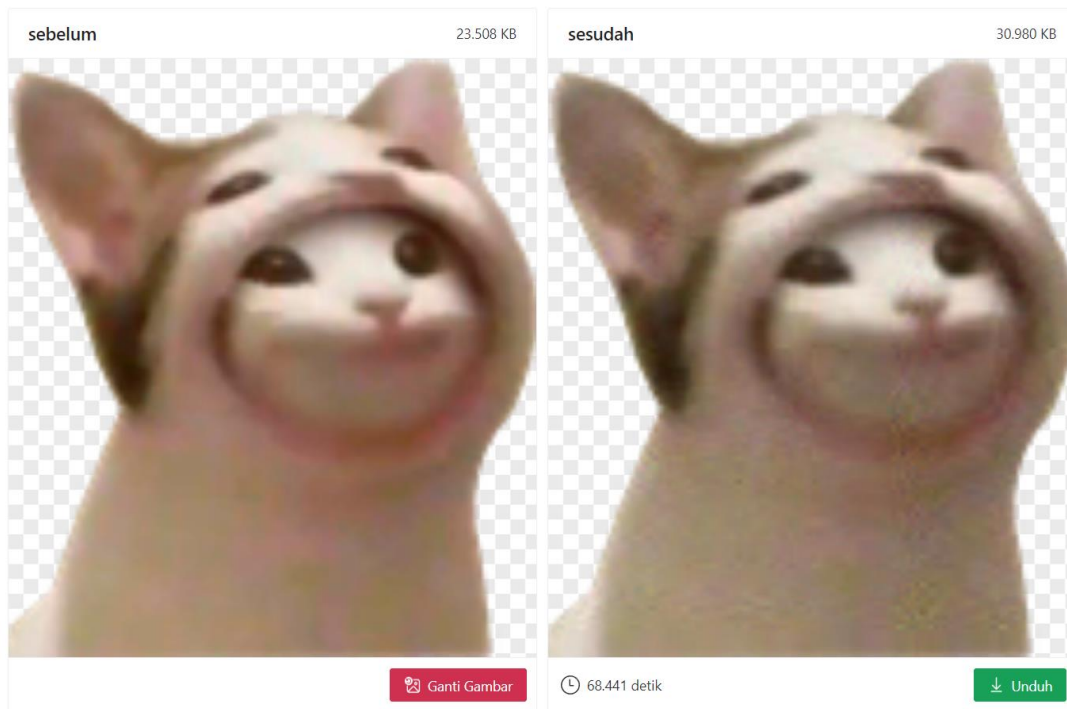


10%

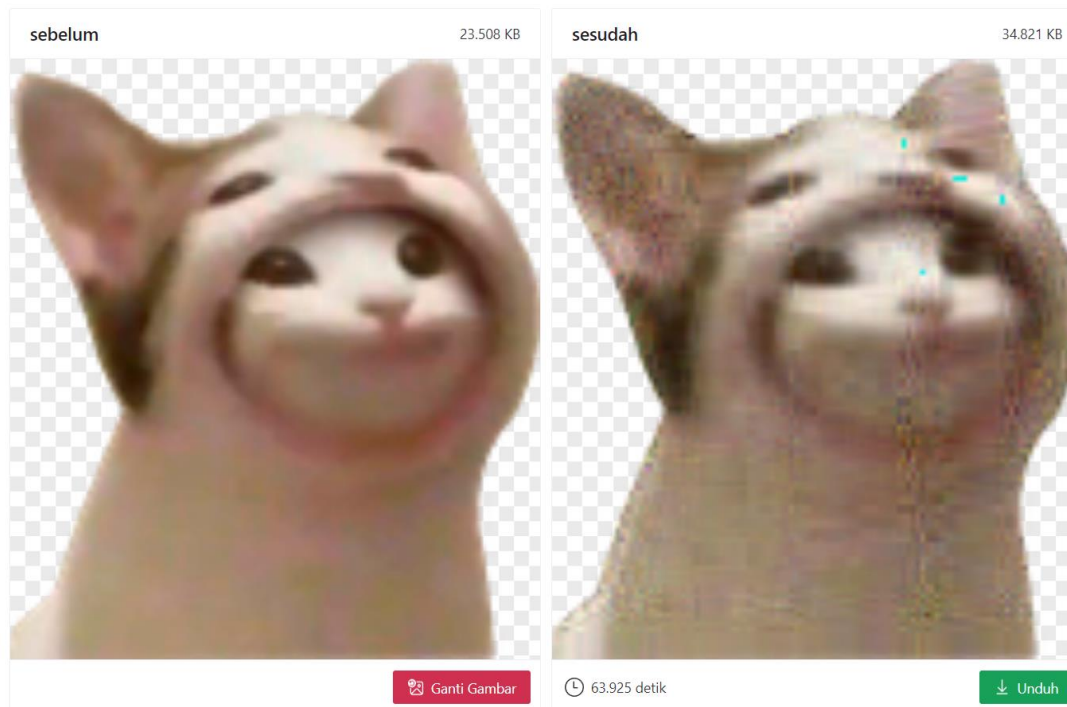


4.3. Gambar PNG Transparan

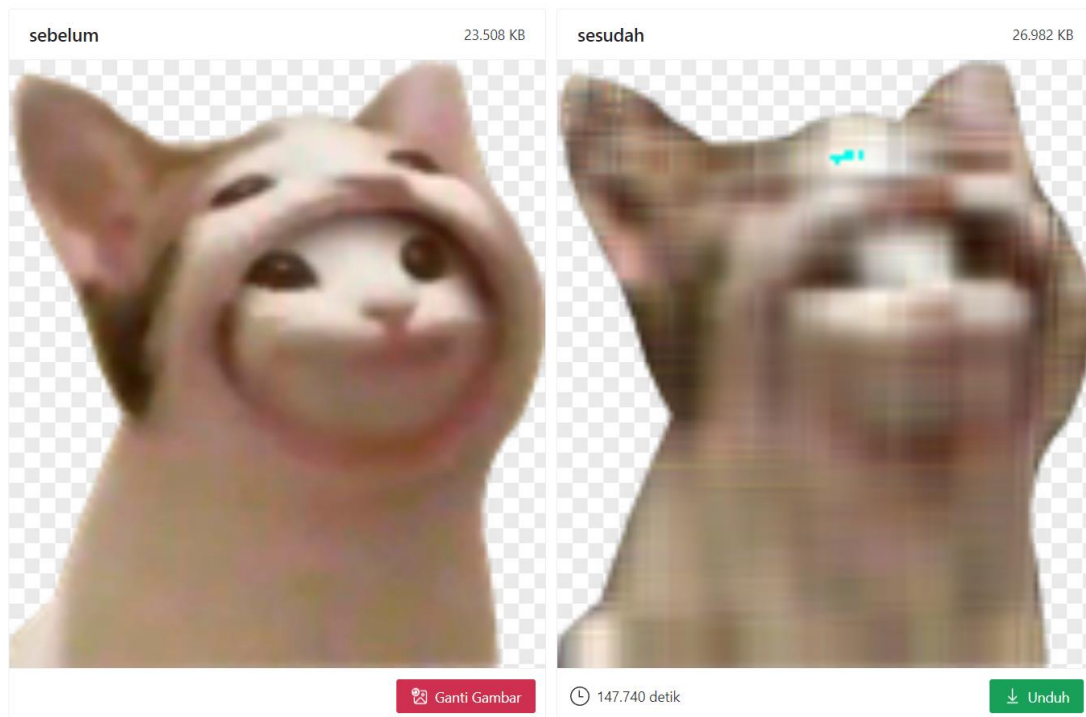
90%



50%



10%



Dari ketiga sampel gambar, yaitu JPG, PNG, dan PNG transparan, yang masing-masing dilakukan pengujian terhadap tiga tingkat kompresi, yaitu 10%, 50%, dan 90%, didapatkan hasil yang memuaskan. Tidak terjadi error selama program dijalankan. Hasil yang didapatkan juga selaras dengan tingkat kompresi yang dimasukkan. Gambar hasil kompresi bisa diunduh dari website.

Tingkat kompresi (rasio) adalah perbandingan ukuran gambar sesudah dan sebelum dikompresi dan dinyatakan dalam persen. Tingkat kompresi sebuah matriks berukuran $m \times n$ dan rank k dinyatakan sebagai berikut

$$r = \frac{k(1 + m + n)}{m \times n} \times 100\%$$

Dari rumusan di atas, terbukti bahwa semakin besar tingkat kompresi, maka rank akan semakin besar. Artinya, jumlah singular value yang digunakan semakin banyak. Akibatnya, gambar yang didapatkan akan semakin jelas. Hal ini sesuai dengan yang kami dapatkan selama eksperimen.

Dapat dilihat pula bahwa ukuran byte gambar setelah kompresi lebih besar daripada sebelum kompresi. Dugaan kami, hal itu disebabkan oleh penambahan warna pada gambar akibat kompresi.

BAB V

PENUTUP

5.1. Kesimpulan

Dari tugas besar IF 2123 Aljabar Linier dan Geometri semester I 2020/2021 berjudul “Aplikasi Nilai Eigen dan Vektor Eigen dalam Kompresi Gambar”, kami berhasil membuat sebuah program kompresi gambar pada sebuah website pada server lokal.

5.2. Saran

Saran-saran yang dapat kami berikan untuk tugas besar IF 2123 Aljabar Linier dan Geometri semester I 2020/2021 adalah:

1. Algoritma yang digunakan pada Tugas Besar ini masih memiliki banyak kekurangan sehingga sangat memungkinkan untuk dilakukan efisiensi, misalnya dengan penggunaan beberapa library yang tersedia. Oleh karena itu, dalam pengembangan program ini, masih bisa dilakukan efisiensi kinerja.
2. Program ini dapat dikembangkan lebih lanjut baik dari segi UI/UX supaya semakin *user-friendly* atau dari segi fungsionalitas program yang dapat dikembangkan untuk cek plagiarisme.
3. Memperjelas spesifikasi dan batasan-batasan setiap program pada file tugas besar untuk mencegah adanya multitafsir dan kesalahpahaman pada proses pembuatan program.
4. Menimbang fungsionalitas dari Program pada Tugas Besar ini, sebaiknya program ini bisa dipublikasikan setelah dikembangkan lebih lanjut. Supaya program ini memiliki kebermanfaatan yang lebih luas.

5.3. Refleksi

Setelah menyelesaikan tugas besar IF 2123 Aljabar Linier dan Geometri semester I 2020/2021, kami dapat merefleksikan beberapa hal, yaitu:

1. Komunikasi antar anggota kelompok berjalan dengan baik, sehingga tidak terjadi miskomunikasi atau kesalahpahaman selama pengerjaan.

2. Selama pengerjaan Tugas Besar, telah dibuat beberapa *milestone* untuk setiap orangnya dengan *deadline* yang bervariasi dan semua terselesaikan tepat waktu.
3. Selama proses pembuatan program, ketika ditemukan ketidaksesuaian saat proses eksperimen, temuan langsung dikomunikasikan kepada anggota kelompok lainnya dan bersama-sama mencari solusi.
4. Perlunya untuk mempelajari *web development* agar kedepannya dapat membuat sebuah website yang lebih fungsional dan lebih *user-friendly* dari segi UI/UX.
5. Lebih merapikan *source code* program karena ada beberapa fungsi yang didefinisikan secara tidak modular.
6. Mengerjakan tugas besar dengan perasaan gembira, karena *it's not worth it if you're not have fun*.

REFERENSI

Welcome to Flask. (n.d.). Retrieved November 3, 2020, from <https://flask.palletsprojects.com/>

(Referensi penggunaan Flask).

Overview – Numpy v1.21 Manual. Retrieved June 22, 2021, from <https://numpy.org/doc/1.21/>

(Referensi library Numpy)