Laporan Tugas Besar II

IF2123 Aljabar Linier dan Geometri

APLIKASI NILAI EIGEN DAN VEKTOR EIGEN DALAM KOMPRESI GAMBAR

Oleh:

Kelompok: Kompresiana

Louis Yanggara	13520063
Wesly Giovano	13520071
Claudia	13520076

TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2021

DAFTAR ISI

BAB I
DESKRIPSI MASALAH

1.1 Latar Belakang

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 1.1. 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.

$$A_{m\times n} = U_{m\times m}\, S_{m\times n}\, V^{T}_{n\times n}$$

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

Gambar 1.2. 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.

1.2 Spesifikasi Tugas

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 $A_{m\times r} = [a_{ij}]$ dan $B_{r\times n} = [b_{ij}]$ akan menghasilkan $C_{m\times n} = A_{m\times r} \times B_{r\times n} = [c_{ij}]$ dengan $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}$, atau secara umum dapat dinyatakan sebagai $c_{ij} = \sum_{k=1}^{r} a_{ik}\, b_{kj}$. Untuk itu, syarat dari perkalian matriks adalah jumlah kolom matriks pertama harus sama dengan jumlah baris matriks kedua. Perkalian matriks tidak bersifat komutatif, artinya $A \times B \neq B \times A$.

2.2 Nilai Eigen dan Vektor Eigen

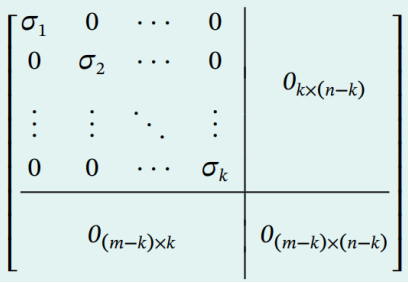
Untuk setiap matriks persegi $A_{n\times n}$ terdapat λ dan $\vec{x}$ sedemikian sehingga $Ax = \lambda\vec{x}$ dengan λ adalah skalar yang disebut nilai eigen dan $\vec{x}$ adalah vektor yang disebut vektor eigen. Nilai eigen dari suatu matriks A dapat dihitung melalui persamaan karakteristik $det(\lambda I - A) = 0$ dengan I adalah matriks identitas. Nilai-nilai eigen kemudian dapat dihitung dengan mencari akar-akar dari persamaan karakteristik tersebut. Vektor eigen untuk masing-masing nilai eigen dapat dihitung dengan mencari solusi x dari persamaan $(\lambda I - A)\vec{x} = 0$.

2.3 Singular Value Decomposition

Singular value decomposition (SVD) adalah suatu metode dekomposisi matriks dengan memanfaatkan nilai singular matriks. Matriks yang akan didekomposisi tidak memiliki batasan ukuran, sehingga dekomposisi ini dapat dilakukan pada matriks selain matriks persegi. Metode SVD dapat memfaktorkan matriks A menjadi bentuk: $A_{m\times n} = U_{m\times m}\, \Sigma_{m\times n}\, V^{T}_{n\times n}$ dengan U adalah matriks vektor singular kiri, Σ adalah matriks yang berisi nilai singular, dan V adalah matriks vektor singular kanan.

Matriks Σ adalah matriks nol berukuran $m \times n$ dengan entri diagonal utamanya adalah nilai-nilai singular $\sigma_i > 0$ dari matriks $A^T A$. Nilai-nilai singular dari matriks $A^T A$ berukuran $m \times n$ adalah $\sigma_i = \sqrt{\lambda_i}$ dengan $i = 1, 2, \ldots, n$. Nilai eigen tersebut adalah nilai eigen dari matriks $A^T A$ dan diurut sedemikian sehingga $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$, sehingga nilai singular matriks A adalah $\sigma_1 \geq \sigma_2 \geq$

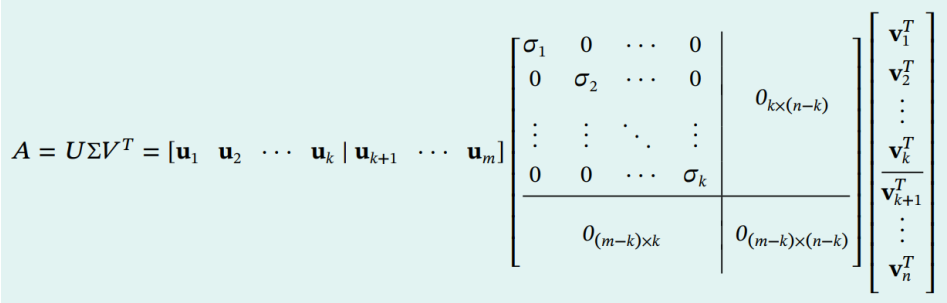
$\cdots \geq \sigma_n > 0$. Secara ekuivalen, entri diagonal utama dari matriks Σ adalah $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0$ dengan k adalah rank dari matriks A.



Gambar 2.1. Penyusunan matriks Σ dalam SVD.

Matriks V adalah matriks yang tersusun dari vektor-vektor eigen secara terurut dari matriks $A^T A$. Dengan kata lain, $V = [\,\overrightarrow{v_1} \mid \overrightarrow{v_2} \mid \cdots \mid \overrightarrow{v_n}\,]$ dengan $\overrightarrow{v_i}$ bersesuaian dengan λ_i yang terurut mengecil untuk $i = 1, 2, \ldots, n$.

Matriks U adalah matriks yang tersusun dari vektor-vektor eigen secara terurut dari matriks AA^T. Dengan kata lain, $U = [\,\overrightarrow{u_1} \mid \overrightarrow{u_2} \mid \cdots \mid \overrightarrow{u_m}\,]$ dengan $\overrightarrow{u_i}$ bersesuaian dengan λ_i yang terurut mengecil untuk $i = 1, 2, \ldots, m$.



Gambar 2.2. Keseluruhan persamaan dekomposisi SVD.

BAB III
IMPLEMENTASI PROGRAM

Program kompresi gambar dengan metode SVD ditulis dengan bahasa Python dengan framework Flask.

3.1 Algoritma SVD

SVD pada awalnya memanfaatkan pencarian nilai eigen berdasarkan persamaan karakteristik matriks AA^T dan matriks A^TA, kemudian menggunakan operasi baris elementer untuk mendapatkan vektor eigen dan menyusunnya ke dalam matriks V dan U. Namun algoritma tersebut terbukti tidak efisien dan memiliki kompleksitas yang sangat tinggi. Eksperimen perlakuan fungsi SVD pada sebuah matriks 4×20 membutuhkan waktu sekitar 1 menit.

Untuk itu, algoritma perhitungan SVD dalam program memanfaatkan algoritma QR yang jauh lebih cepat. Modul ini terbagi menjadi dua buah fungsi, yaitu sebagai berikut.

- Fungsi eigen(A: np.matrix) → (np.array, np.matrix)
  Fungsi ini menerima sebuah matriks Hermitian (matriks AA^T atau matriks A^TA), kemudian mengembalikan daftar nilai eigen yang turut mengecil dan vektor eigen yang tersusun dalam bentuk matriks sesuai dengan nilai eigen dengan memanfaatkan algoritma QR.

- Fungsi svd(A: np.matrix) → (np.matrix, np.matrix, np.matrix)
  Fungsi ini menerima sebuah matriks sembarang misalnya $A_{m \times n}$, dan mengembalikan tiga buah matriks yaitu U, Σ, dan V^T dengan $A = U\Sigma V^T$ sesuai algoritma SVD.

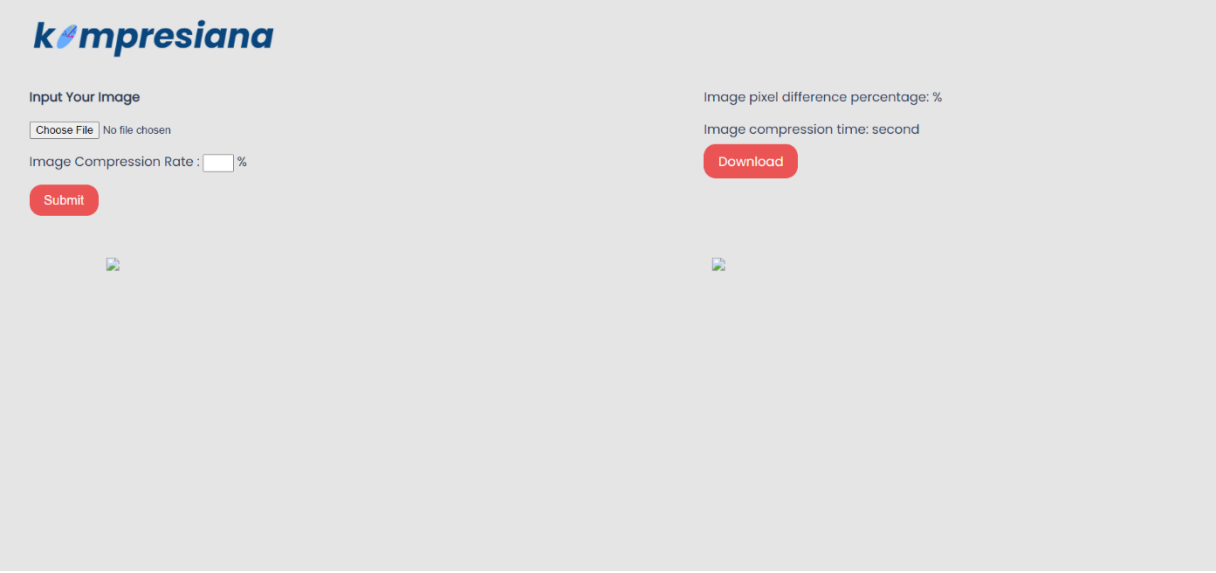
3.2 Kompresi Gambar

Proses kompresi gambar dimulai dari menerima input file image yang akan dikompres. Gambar yang sudah diterima akan dipecah menjadi 3 matriks yang melambangkan warna Merah, Hijau, dan Biru yang ada pada gambar. Setelah ketiga matriks terbentuk, maka proses untuk melakukan SVD terhadap ketiga matriks akan dimulai. Untuk menentukan ratio pengurangan yang diinginkan, digunakan LimitSVD atau banyaknya kolom dan baris yang akan digunakan untuk Image hasil

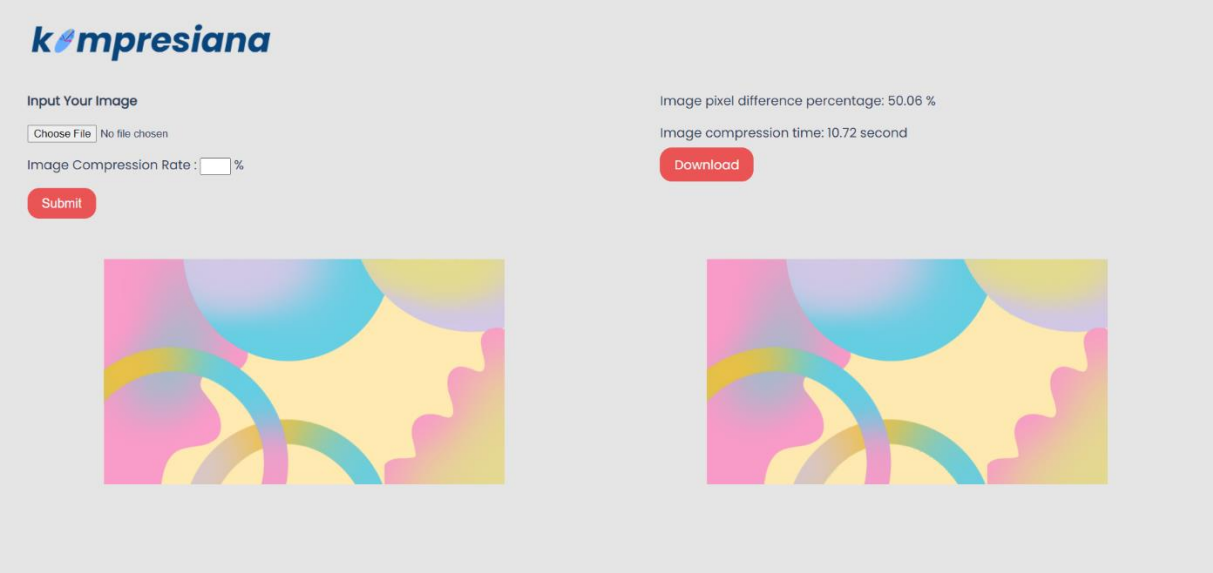
kompresi. Setelah proses SVD selesai, masing-masing matriks akan dibuat kembali ke image-image yang bersesuaian dengan warnanya sehingga diperoleh 3 buah image berupa Red, Green, dan Blue. Ketiga gambar ini kemudian di merge sehingga diperoleh final Image yang merupakan gambar hasil kompresi. Proses pengolahan gambar pada bagian ini menggunakan library Pillow yang ada di Python.

3.3 Website

Tampilan website dirancang secara kasar terlebih dahulu menggunakan Figma, lalu direalisasikan menggunakan HTML (HyperText Markup Language) dan CSS (Cascading Style Sheets) tanpa menggunakan framework untuk bagian front end website. HTML digunakan untuk membuat struktur dasar konten pada halaman web, setelah itu digunakan CSS untuk melakukan styling dan memperindah tampilan website seperti pemberian warna, dan lain-lain. Pemilihan warna background website juga bertujuan agar gambar sebelum kompresi dan setelah kompresi dapat lebih mudah terlihat (dibandingkan jika menggunakan warna gelap). Penyusunan konten dilakukan dengan menggunakan flexbox, salah satunya untuk menyusun gambar sebelum kompresi dan setelah kompresi.

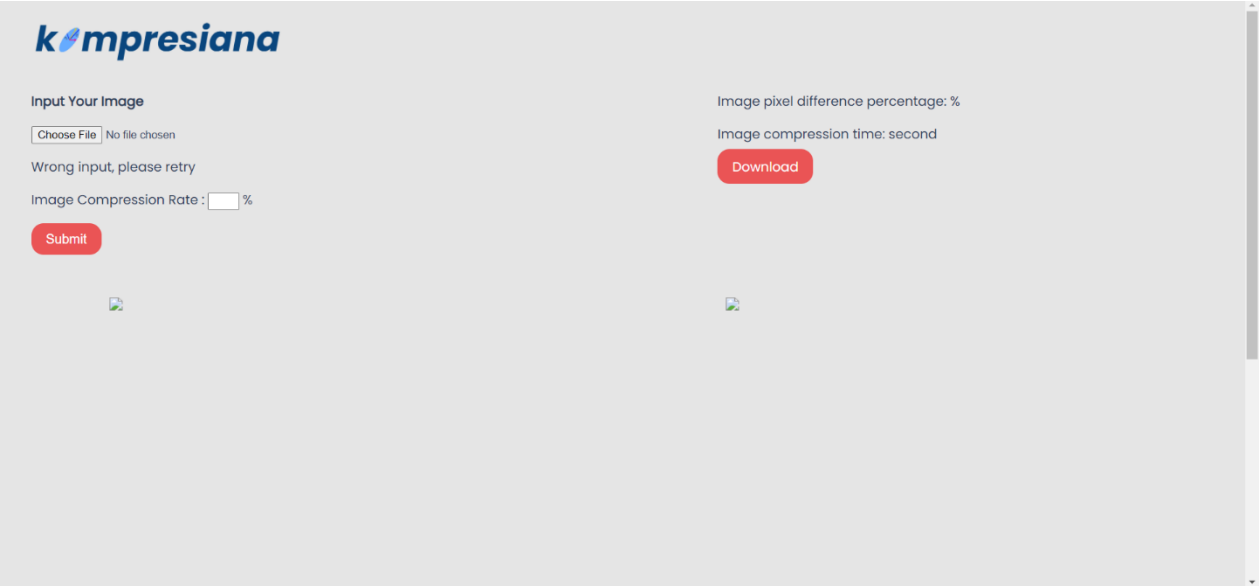


Gambar 3.1. Tampilan website sebelum melakukan kompresi gambar



Gambar 3.2. Tampilan website setelah melakukan kompresi gambar

Untuk menyambungkan bagian back end dengan front end, digunakan framework Flask pada app.py. Pada app.py, fungsi-fungsi yang sudah dibuat pada bagian sebelumnya dipanggil untuk melakukan pemrosesan pada gambar yang sudah diupload dari user. Selain itu, juga dilakukan validasi input dari user karena kompresi hanya dapat dilakukan untuk foto berformat .jpeg, .jpg, atau .png. Jika user menginput file di luar format tersebut, maka website akan menampilkan flash message yang menuliskan bahwa input tidak valid.



Gambar 3.3 Tampilan flash message

Selain itu, gambar yang diupload oleh user akan dimasukkan ke dalam folder static, dikompres, dan gambar hasil kompresan juga akan dimasukkan kembali ke dalam folder static dengan nama compressed_<nama_file>.ekstensi

```
filename = secure_filename(pic.filename) # take the name of the pic
pic.save(os.path.join(app.config['UPLOAD_FOLDER'],filename)) # save to static folder

pic = Image.open(request.files['file'].stream) # convert into PIL Image
finalImage, diffRatio, runtime = compress(pic,perc) # compression

global finalname
finalname = 'compressed_' + filename
finalImage = finalImage.save(os.path.join(app.config['UPLOAD_FOLDER'],finalname)) # save the compressed picture to static folder
```

Gambar 3.4. Kode program save gambar

Gambar hasil kompresi juga dapat diunduh oleh user dengan menekan tombol download. Untuk menampilkan runtime dan image pixel difference percentage, nilai yang sudah diproses dari back end akan ditampilkan ke front end dengan memasukkan value yang direturn pada route di app.py pada template HTML

```
return render_template('index.html', rtime = runtime, pixel = diffRatio, imagename = filename, imagefinal = finalname )
```

Gambar 3.5. Kode program yang mereturn value

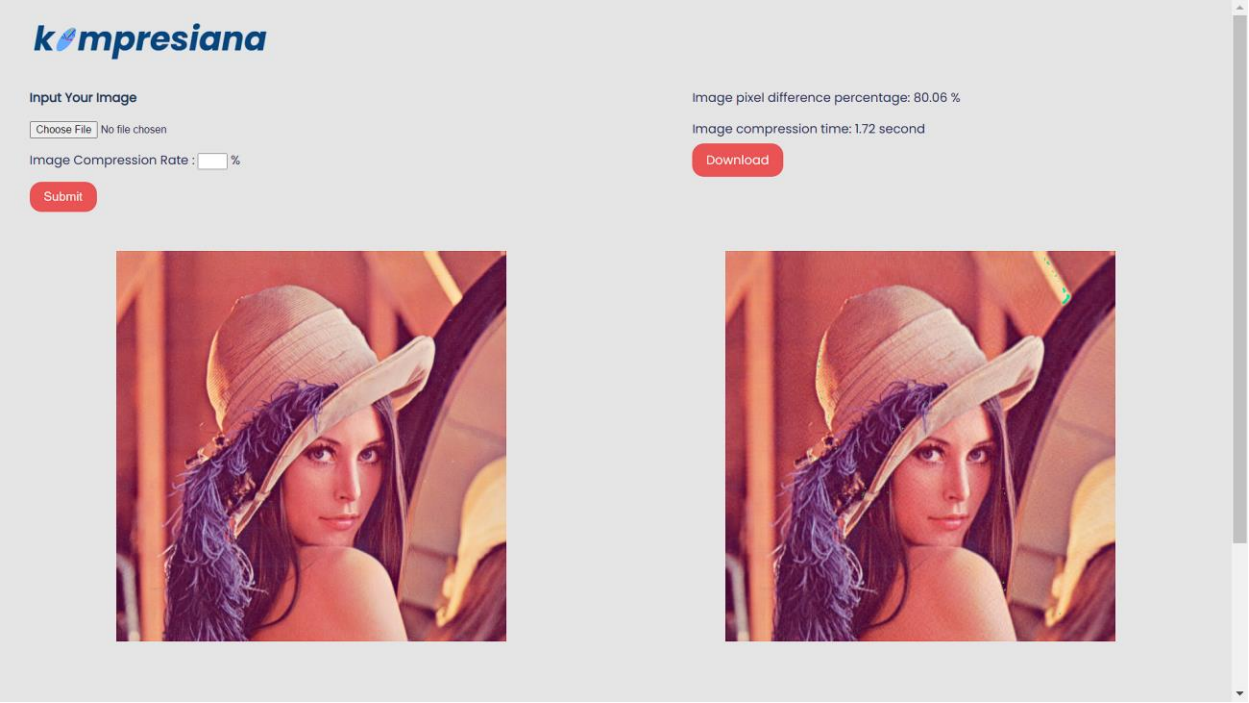
```
<div class="output-box">
    <p>Image pixel difference percentage: {{ pixel }} %</p>
    <p>Image compression time: {{ rtime }} second </p>
    <a href="{{ url_for('download') }}" class="download-button">Download</a>
</div>
```

Gambar 3.6. Kode HTML yang menampilkan value yang direturn

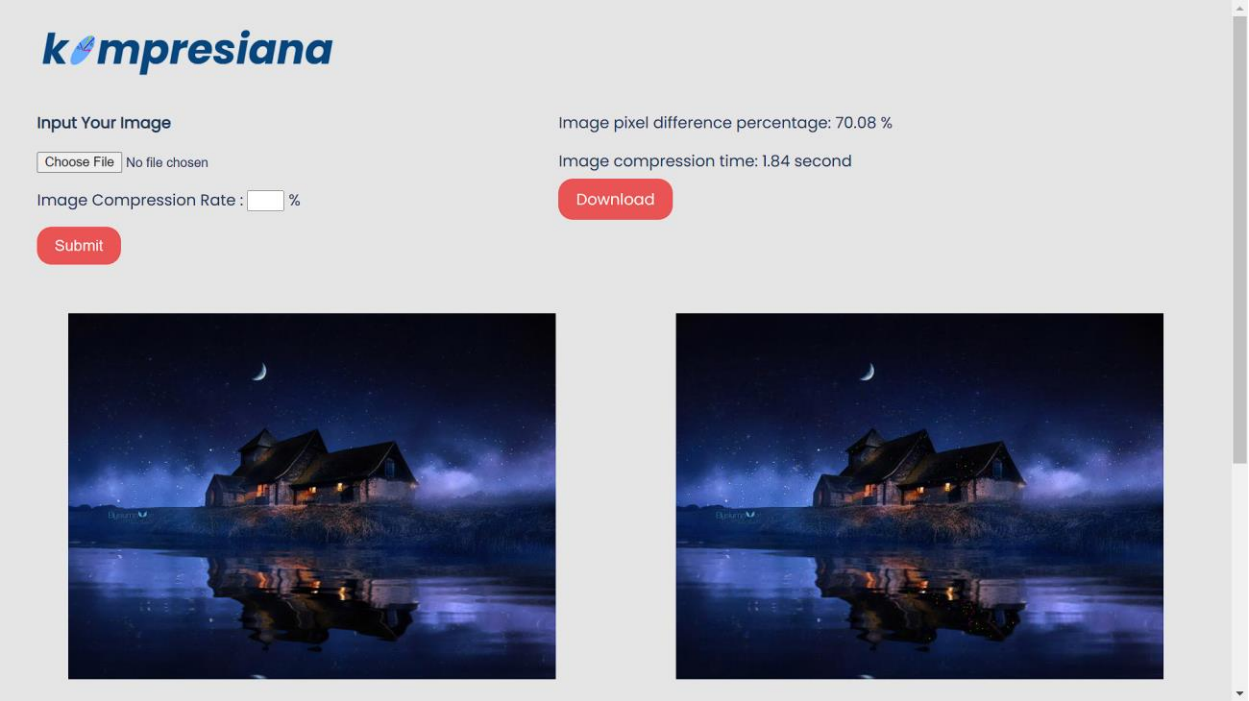
BAB IV
EKSPERIMEN

Eksperimen dilakukan terhadap program dengan menggunakan gambar dengan resolusi yang berbeda-beda. Secara umum, program menerima sebuah file png/jpg melalui tombol “Choose File” dan besaran kompresi yang dilakukan terhadap file gambar. Program kemudian akan melakukan kompresi terhadap gambar sesuai dengan masukan pengguna, dan menampilkan gambar sebelum dan sesudah kompresi pada layar. Program juga akan menampilkan waktu yang dibutuhkan proses kompresi tersebut.

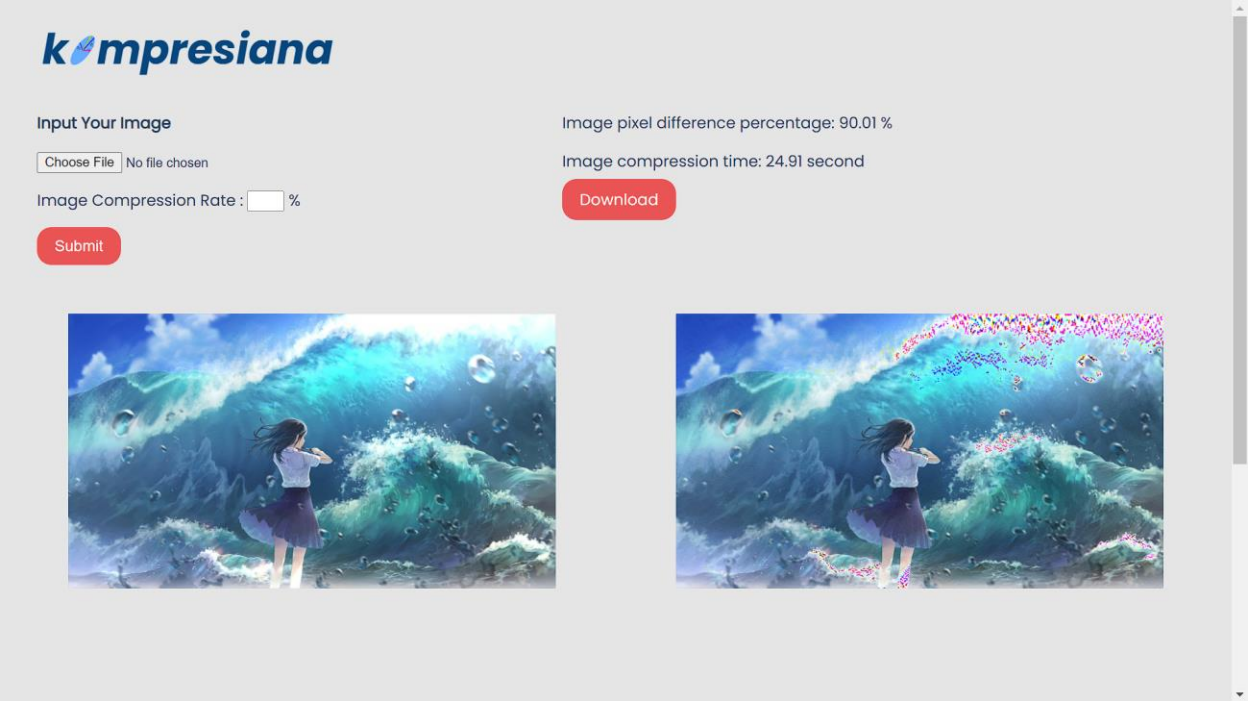
Berikut adalah lima buah gambar tampilan web hasil eksperimen kompresi gambar dengan beragam resolusi dan besaran nilai kompresi.



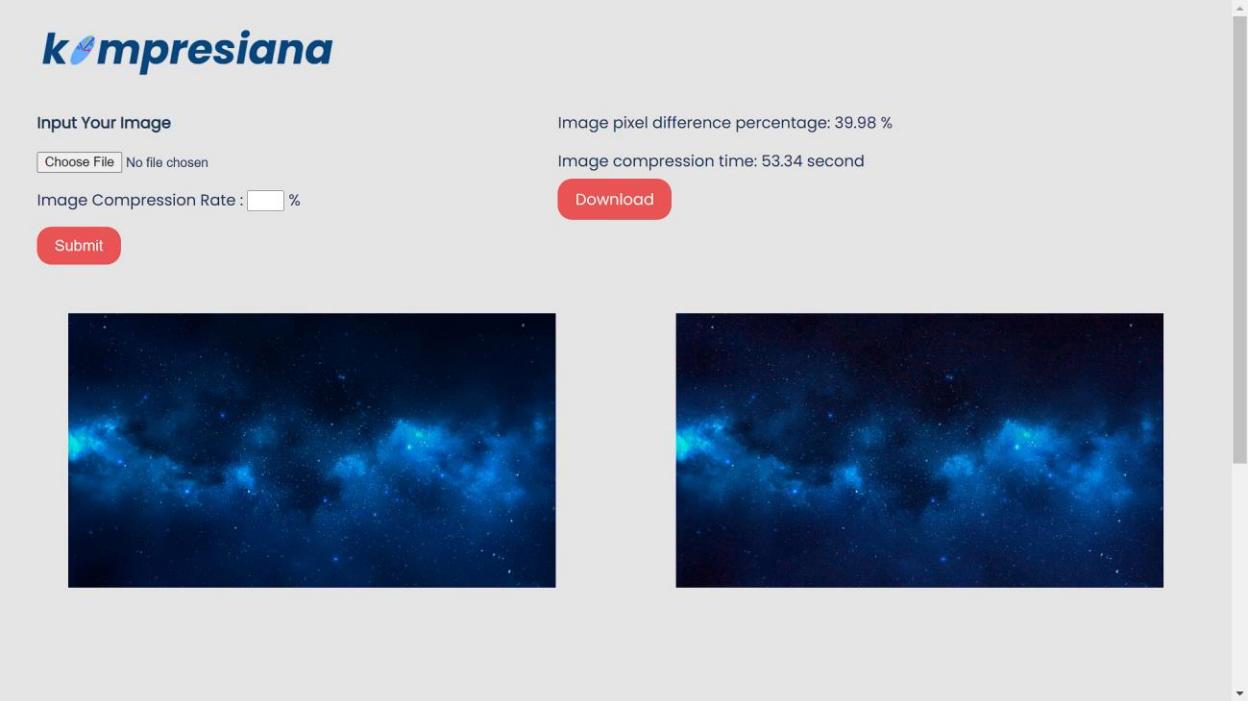
Gambar 4.1. Kompresi sebesar 80% terhadap gambar berukuran 512 x 512.



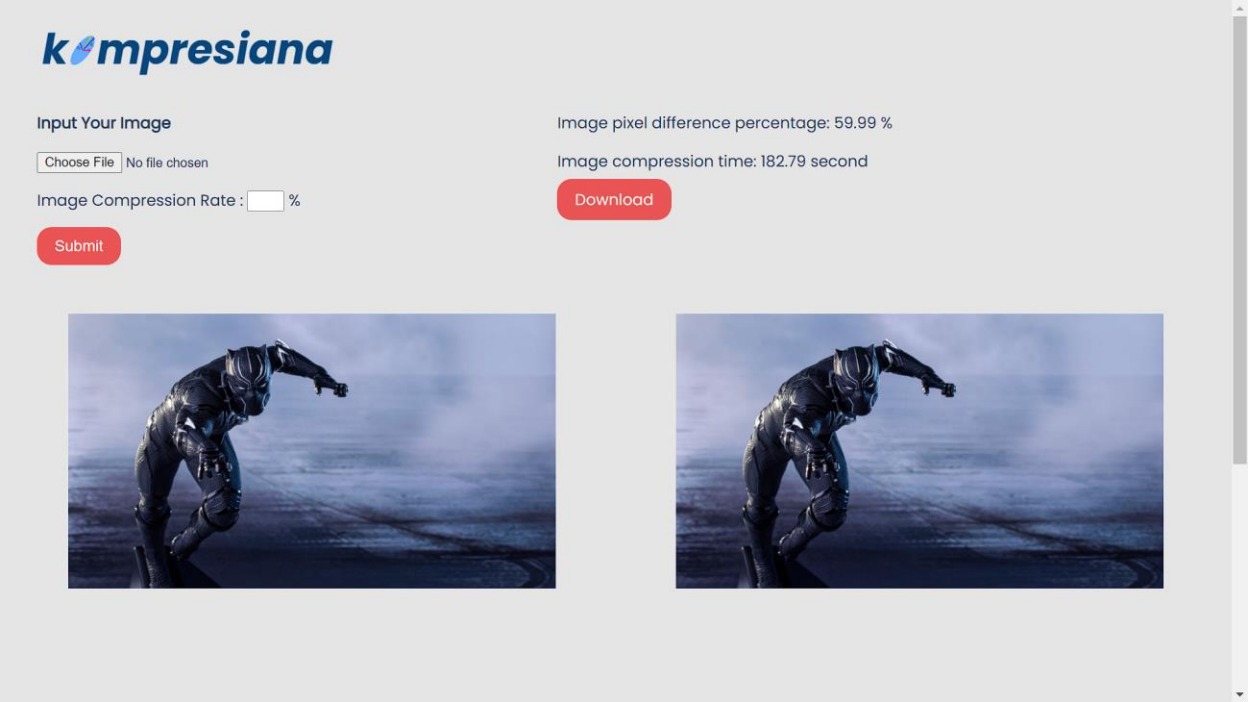
Gambar 4.2. Kompresi sebesar 70% terhadap gambar berukuran 640 x 480.



Gambar 4.3. Kompresi sebesar 90% terhadap gambar berukuran 1920 x 1080.



Gambar 4.4. Kompresi sebesar 40% terhadap gambar berukuran 2560 x 1440.



Gambar 4.5. Kompresi sebesar 60% terhadap gambar berukuran 3840 x 2160.

BAB V
PENUTUP

5.1 Simpulan

Program website kompresi gambar berhasil dirancang dengan menggunakan bahasa Python dengan framework Flask dan tampilan yang dirancang menggunakan HTML dan CSS. Program ini dapat menerima input gambar dari user dan melakukan kompresi gambar. Proses kompresi gambar dilakukan dengan menggunakan metode SVD dengan algoritma QR untuk kemudian direkonstruksi kembali gambar yang diinputkan. Hasil kompresi gambar ini kemudian dapat ditampilkan di website dan diunduh ke folder lokal user.

5.2 Saran

Saran yang dapat penulis berikan antara lain optimasi terhadap program agar proses kompresi dapat berjalan lebih cepat dan dapat menerima lebih banyak format gambar, selain itu user interface dari website dapat ditingkatkan lagi.

5.3 Refleksi

Dari tugas besar ini kami belajar banyak hal, antara lain membuat front-end dari sebuah website dengan HTML, CSS, serta konsep flexbox pada CSS, cara menghubungkan front-end dan back-end dengan framework Flask, dan yang paling utama, bahwa algoritma SVD dapat digunakan untuk mengompresi gambar berdasarkan konsep nilai eigen. Selain itu, kami juga ingin berterima kasih kepada dosen pengampu mata kuliah IF2123 dan asisten atas kesempatan yang diberikan, kami yakin apa yang sudah kami pelajari selama membuat tugas ini akan sangat berguna ke depannya.

REFERENSI

https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2021-2022/algeo21-22.htm

Anton, Howard dkk. 2019. Elementary Linear Algebra (edisi ke-12). Wiley: Amerika Serikat.

QRalgorithm (cornell.edu)