

Penerapan SVD Sederhana Dalam Kompresi Gambar

Ferdinand Gabe Tua Sinaga and 13523051^{1,2}

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13523051@mahasiswa.itb.ac.id, ferdinandgts5@gmail.com

Abstract—SVD (Singular Value Decomposition) adalah salah satu teknik dekomposisi matriks dalam aljabar linier yang dapat dimanfaatkan untuk melakukan kompresi terhadap suatu gambar. Teknik ini memungkinkan pengurangan dimensi data dengan tetap mempertahankan informasi penting dalam gambar. Dalam konteks kompresi gambar, SVD dapat digunakan untuk menghasilkan representasi gambar yang lebih efisien dengan mengurangi ukuran data tanpa mengorbankan kualitas visual secara signifikan.

Keywords—SVD, Image Compression, Data Reduction and Factorization Matrix.

I. INTRODUCTION

Gambar adalah salah satu alat bagi kita untuk mengabadikan momen berharga dalam hidup kita. Seiring perkembangan teknologi digital, produksi dan penyimpanan gambar semakin menjadi bagian yang tidak terpisahkan dalam berbagai aspek kehidupan, mulai dari kebutuhan pribadi hingga profesional. Dalam era digital saat ini, jumlah data gambar yang dihasilkan setiap hari sangatlah besar. Menurut laporan terbaru dari Statista, lebih dari 1,8 miliar gambar diunggah ke internet setiap harinya. Jumlah ini menciptakan tantangan besar dalam hal penyimpanan dan transmisi data, terutama bagi layanan cloud dan penyedia penyimpanan berbasis digital.

Semakin baik kualitas gambar tersebut tentu akan semakin banyak ruang penyimpanan yang diperlukan untuk menyimpannya akan tetapi, tidak semua perangkat keras mumpuni dalam menyimpan sejumlah dataset gambar dalam resolusi yang sangat tinggi. Untuk mengatasi hal ini berbagai Teknik dekomposisi dikembangkan untuk mengurangi ukuran file tersebut dengan tetap mempertahankan fitur utamanya.

Salah satu teknik kompresi yang efektif adalah Singular Value Decomposition atau disingkat SVD. Teknik ini tidak hanya bermanfaat dalam kompresi gambar tetapi juga memiliki aplikasi yang luas dalam berbagai bidang, seperti pengolahan citra medis, analisis data besar (big data), dan machine learning. Sebagai contoh, dalam dunia

medis, kompresi gambar berbasis SVD digunakan untuk mengoptimalkan penyimpanan citra MRI dan CT-scan tanpa kehilangan informasi kritis yang dibutuhkan untuk diagnosis. Di sisi lain, dalam bidang pembelajaran mesin, SVD digunakan untuk mengurangi dimensi data, sehingga meningkatkan kecepatan dan efisiensi komputasi.



Contoh Kompresi

<https://edsmathscholar.com/diagonalization-of-symmetric-matrices/>

Teknik SVD bekerja dengan cara mendekomposisi matriks data gambar menjadi tiga komponen utama, yaitu matriks ortogonal kiri (U), matriks diagonal (Σ), dan matriks ortogonal kanan (V^T). Matriks diagonal berisi nilai singular yang merepresentasikan tingkat kontribusi masing-masing komponen terhadap struktur keseluruhan gambar. Dengan mengabaikan nilai singular yang kecil, ukuran data dapat dikurangi secara signifikan tanpa menghilangkan informasi penting dari gambar tersebut. Pendekatan ini tidak hanya efektif untuk kompresi, tetapi juga memastikan bahwa kualitas visual gambar tetap terjaga, menjadikannya pilihan yang ideal dalam berbagai aplikasi komputasi.

Melalui penjelasan diatas tujuan dari dibuatnya makalah ini adalah untuk menjelaskan dan menerapkan Teknik SVD (Single Value Decomposition) untuk memperkecil ukuran suatu file gambar dengan mempertahankan beberapa fitur utama dari gambar agar gambar tersebut tetaplah bermakna.

II. DASAR TEORI

Dalam dunia digital, kompresi gambar adalah teknik penting untuk mengurangi ukuran data tanpa kehilangan kualitas visual yang signifikan. Secara umum, terdapat dua kategori utama kompresi gambar:

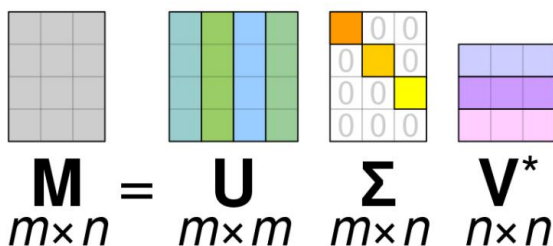
1. Kompresi Lossless: Teknik ini memastikan data asli dapat dipulihkan sepenuhnya setelah proses dekompresi. Metode ini digunakan untuk data yang memerlukan akurasi tinggi, seperti dokumen digital atau citra medis. Contoh format lossless adalah PNG dan BMP.
2. Kompresi Lossy: Teknik ini mengorbankan sebagian informasi untuk mencapai tingkat kompresi yang lebih tinggi. Informasi yang hilang biasanya berupa detail halus yang tidak terlalu signifikan untuk persepsi manusia. Contoh format lossy adalah JPEG dan WebP.

Kompresi berbasis lossy lebih sering digunakan pada aplikasi gambar digital karena mampu menghasilkan pengurangan ukuran yang signifikan dengan tetap mempertahankan kualitas visual. Teknik ini sering memanfaatkan algoritma seperti Discrete Cosine Transform (DCT) atau metode dekomposisi matriks seperti Singular Value Decomposition (SVD).

Singular Value Decomposition atau SVD adalah teknik dekomposisi yang mengubah representasi suatu matriks A menjadi 3 buah matriks yang terdiri dari matriks ortogonal kiri (U), matriks diagonal (Σ), dan matriks ortogonal kanan (V^T).

SVD memungkinkan kita memisahkan sebuah matriks ke dalam komponen-komponen dasar yang mencerminkan struktur data tersebut. Matriks U menggambarkan basis ortonormal untuk ruang baris, V menggambarkan basis ortonormal untuk ruang kolom, dan Σ mengandung nilai singular yang mewakili kontribusi masing-masing dimensi terhadap struktur matriks.

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



Ilustrasi SVD

<https://chaosarium.xyz/singular-value-decomposition/>

Matriks orthogonal kiri dan kanan adalah matriks yang nilai hasil perkalian dot antar kolomnya sama dengan nol. Matriks tersebut disusun berdasarkan nilai dari vector eigennya. Sementara itu, matriks diagonal adalah matriks

diagonal yang elemen diagonalnya merupakan nilai singular dari matriks $A^T A$. Dimana nilai dari singularnya diurutkan dari yang nilai singular terbesar ke nilai singular terkecil.

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \dots \geq \sigma_N \geq 0$$

Untuk menghitung nilai singular hingga ke n kita harus mengetahui nilai eigen dari $A^T A$. Salah satu cara untuk menghitung nilai eigen dan vector eigennya adalah dengan mencari nilai yang memenuhi persamaan karakteristik.

Persamaan karakteristik dapat dicapai melalui beberapa penurunan rumus seperti di bawah ini:

Diberikan sebuah matriks B berukuran $n \times n$. Vektor eigen dan nilai eigen dari matriks B dihitung sebagai berikut:

$$\begin{aligned}
 Bx &= \lambda x \\
 I(Bx) &= \lambda Ix \text{ (kalikan kedua ruas dengan I)} \\
 Bx &= \lambda Ix \\
 (\lambda I - B)x &= 0 \\
 \det(\lambda I - B) &= 0
 \end{aligned}$$

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

Nilai yang memenuhi persamaan karakteristik tersebut adalah nilai eigennya. Setelah kita mendapatkan nilai eigennya maka kita bisa mencari vector eigennya dengan mensubstitusi lambda dengan nilai eigen tersebut agar mendapatkan vector eigen untuk nilai lambda tertentu.

Untuk menentukan komponen-komponen seperti matriks ortogonal dan matriks diagonal dari sebuah matriks sembarang berukuran $m \times n$ kita data menerapkan beberapa Langkah seperti berikut:

1. Untuk vektor singular kiri, hitung nilai-nilai eigen dari AA^T .
2. Tentukan vektor-vektor eigen yang berkoresponden dengan nilai-nilai eigen dari AA^T . Setelah itu, normalisasikan dengan cara membagi setiap komponen vektor dengan panjang vektornya. Matriks U diperoleh dari himpunan vektor-vektor eigen ini.
3. Untuk vektor singular kanan, hitung nilai-nilai eigen dari $A^T A$ lalu tentukan nilai-nilai-singularnya.
4. Tentukan vektor-vektor eigen yang berkoresponden dengan nilai-nilai eigen dari $A^T A$. Setelah itu, normalisasikan dengan cara membagi setiap komponen vektor dengan panjang vektornya. Matriks V diperoleh dari himpunan vektor-vektor eigen ini. Setelah itu transpose matriks V untuk mendapatkan V^T .

5. Bentuklah matriks Σ berukuran $m \times n$ dengan elemen-elemen diagonalnya adalah nilai-nilai singular tidak nol dari matriks A dengan susunan dari besar ke kecil. Nilai singular di dalam Σ adalah akar pangkat dua dari nilai-nilai eigen yang tidak nol dari $A^T A$.

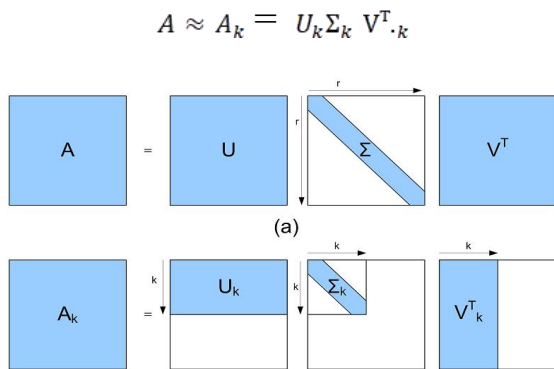
6. Dengan komponen-komponen tersebut, matriks dapat direkonstruksi menggunakan formula:

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

Dengan mengikuti langkah-langkah diatas kita dapat dengan mudah mendapatkan representasi matriks orthogonal dan diagonal yang dibutuhkan untuk membangun matriks A , akan tetapi dalam hal kompresi kita tidak memerlukan seluruh nilai baik dari matriks orthogonal dan diagonalnya.

Dalam hal kompresi kita memiliki pilihan untuk mengambil beberapa nilai singular dari matriks A . Hal itu bisa terjadi karena jika kita analisis lebih lanjut terhadap pernyataan “nilai singular diurutkan secara mengecil” kita bisa mendapatkan beberapa kesimpulan seperti beberapa nilai singular terbawah bisa saja memiliki nilai yang sangat kecil sehingga dapat dianggap sebagai noise.

Berdasarkan penjelasan tersebut dapat diambil kesimpulan bahwa nilai dari matriks A dapat diaproksimasi menggunakan persamaan dibawah ini dengan k merepresentasikan seberapa banyak nilai yang diambil.



Gambar yang dihasilkan dari aproksimasi tersebut akan memiliki ukuran sebesar $k \times (m+n+1)$ byte. Sementara gambar asli akan memiliki ukuran sebesar $m \times n$ byte. Tujuan dari dilakukannya kompresi dengan SVD adalah untuk membuat ukuran dari gambar yang telah diaproksimasi lebih kecil ukurannya dari gambar aslinya atau dalam notasi matematikanya.

$$k \times (m+n+1) < m \times n.$$

Berdasarkan notasi matematika diatas, nilai k yang dipilih haruslah memenuhi ketentuan $k < m \times n / (m+n+1)$. Pemilihan nilai k ini menjadi sangat krusial karena harus menyeimbangkan antara tingkat kompresi yang optimal

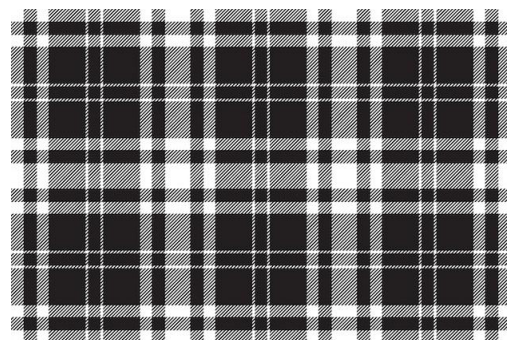
dengan kualitas gambar yang masih dapat diterima. Untuk mengevaluasi efektivitas berbagai teknik kompresi dan mengukur tingkat kompresi suatu gambar, perlu dilakukan perbandingan hasil kompresi menggunakan berbagai nilai k yang berbeda.

Beberapa metode yang valid untuk menghitung nilai efektifitas dari hasil kompresi dengan nilai k adalah

1. Compression Ratio: Ratio yang menghitung perbandingan antara ukuran gambar asli dengan ukuran hasil kompresi
2. MSE (Mean Square Error): MSE dimanfaatkan dalam perhitungan efektivitas hasil kompresi karena MSE dapat menjadi indikator untuk menilai kualitas kompresi dengan cara menghitung rata-rata kuadrat dari perbedaan nilai piksel antara gambar asli dengan hasil kompresinya.
3. SSIM (Structural Similarity Index Measure): SSIM merupakan metode pengukuran yang menilai kemiripan struktural antara gambar asli dan hasil kompresi dengan mempertimbangkan komponen luminance (kecerahan), contrast (kontras), dan structure (struktur), sehingga lebih mendekati persepsi visual manusia dalam menilai kualitas gambar.
4. PSNR (Peak Signal-to-Noise Ratio): PSNR adalah metrik yang mengukur kualitas rekonstruksi gambar terkompresi dengan menghitung perbandingan antara kekuatan sinyal maksimum dengan noise yang mempengaruhi representasi gambar, dimana semakin tinggi nilai PSNR (dalam dB) menunjukkan kualitas kompresi yang semakin baik.

III. HASIL DAN DISKUSI HASIL EKSPERIMEN

Dalam eksperimen yang dilakukan kami memilih 2 buah gambar seperti berikut untuk dianalisis pengaruh banyaknya k dengan hasil kompresi gambar



Gambar Simple Pattern

bit.ly/3ZVSGD9



Gambar anjing
<http://surl.li/paoubc>

Pada percobaan kali ini kami menggunakan library numpy dalam python untuk melakukan dekomposisi dan mengambil nilai singular hingga ke k. Dimana nilai k ini akan berubah seiring berjalannya iterasi dalam loop. Hasil dari iterasi tersebut akan disatukan dan dijadikan gambar oleh library PIL dalam python untuk menghasilkan gambar baru. Berikut kode yang kami implementasikan:

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image

def compress_image_with_svd(image_path, k):
    # Load Gambar dan konversi ke grayscale
    img = Image.open(image_path).convert("L")
    img_matrix = np.array(img)
    U,S,VT=np.linalg.svd(img_matrix,full_matrices=False)

    # Pertahankan nilai singular hingga k
    S_k = np.diag(S[:k])
    U_k = U[:, :k]
    VT_k = VT[:k, :]

    # Membangun gambar hasil dekomposisi
    compressed_img_matrix = np.dot(U_k, np.dot(S_k,
    VT_k))

    return img_matrix, compressed_img_matrix

image_path = "image_bw.png"
k_values = [5, 20, 50, 100]

original = None
plt.figure(figsize=(15, 8))

for i, k in enumerate(k_values):
    if original is None:
        original, _ = compress_image_with_svd(image_path,
k)

    _,compressed= compress_image_with_svd(image_path,
k)

    plt.subplot(1, len(k_values), i + 1)
```

```
plt.title(f"k={k}")
plt.imshow(compressed, cmap="gray")
plt.axis("off")

plt.suptitle("Compressed Images for Different k Values")
plt.show()
```

Berikut kami sajikan hasil kompresi untuk 2 gambar yang menjadi objek eksperimen:



Anjing k5



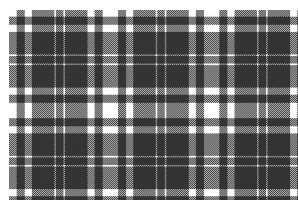
Anjing k20



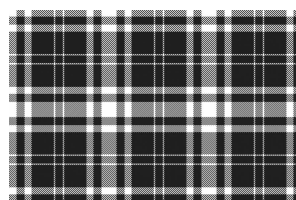
Anjing k50



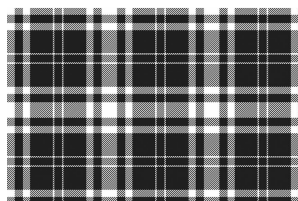
Anjing k100



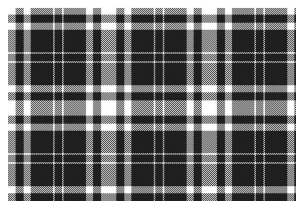
Simple k5



Simple k20



Simple k50



Simple k100

Berdasarkan hasil eksperimen diatas dapat diketahui bahwa perbedaan nilai k yang diambil sangat berpengaruh terhadap hasil kompresi. Saat nilai dari k = 5 kita dapat melihat bahwa gambar sangat blur sehingga lumayan sulit untuk mengetahui gambar apa yang direpresentasikan.

Dengan mengamati lebih dalam terutama pada gambar anjing dapat dilihat bahwa semakin besar nilai k/ nilai eigen yang digunakan untuk merepresentasikan sebuah gambar semakin baik juga gambar hasil kompresinya. Dari hal itu kita bisa mengetahui bahwa k berbanding lurus dengan kualitas gambar yang ada. Tetapi pada beberapa kasus seperti pada gambar simple ia hanya membutuhkan k = 20 supaya kita dapat melihat dengan jelas gambar yang ada. Hal ini merepresentasikan tidak

ada nilai k yang pasti untuk semua gambar, bisa menghasilkan gambar kompresi yang cukup baik.

Hubungan dari banyaknya k yang diambil dengan kualitas gambar dapat dijelaskan melalui metric yang dijelaskan dalam dasar teori seperti metric compression ratio, MSE, SSIM dan PSNR. Berikut hubungan kedua hal tersebut.

Value of k	Compression Ratio	MSE (dB)	PSNR (dB)	SSIM
5	95.95	588.63	20.43	0.74
20	23.99	119.98	27.34	0.82
50	9.60	49.91	31.15	0.87
100	4.80	24.97	34.16	0.91

Table Hubungan K dengan metrik Gambar Anjing Dimensi Gambar Anjing 800x1200

Value of k	Compression Ratio	MSE (dB)	PSNR (dB)	SSIM
5	71.95	560.92	20.64	0.92
20	17.99	9.77	38.23	0.99
50	7.20	6.99	39.68	0.99
100	3.60	4.71	41.40	0.99

Table Hubungan K dengan metrik Gambar Simple Dimensi Gambar simple 600x900

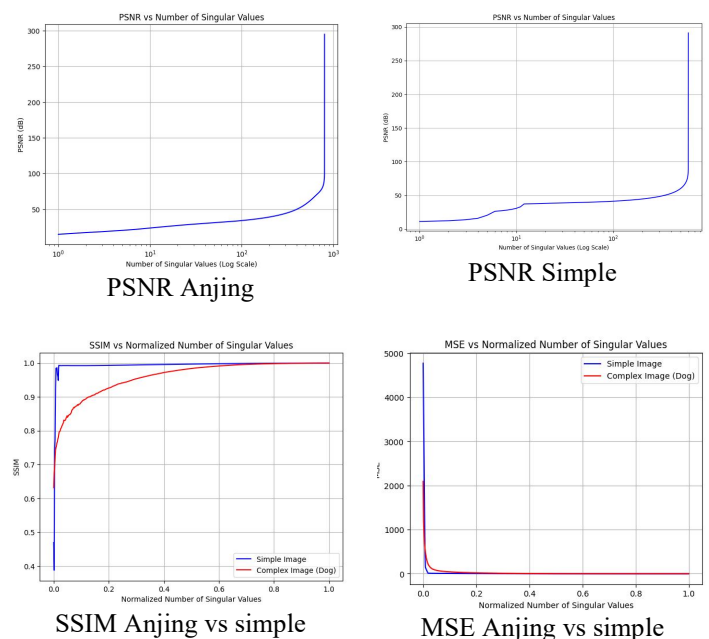
Table diatas memperlihatkan metric MSE & PSNR direpresentasikan dalam dB. Sementara ratio kompresi dalam table itu merepresentasikan seberapa baik hasil kompresi tersebut gambar asli. Berdasarkan gambar diatas kita dapat menarik beberapa kesimpulan:

1. Banyaknya k merepresentasikan seberapa banyak nilai singular/nilai eigen yang dimanfaatkan dalam kompresi tersebut.
2. Nilai dari k bisa merepresentasikan seberapa banyak nilai yang dipakai dalam kompresi sehingga ketika nilai k yang digunakan kecil maka rasio kompresinya akan besar walaupun dengan rasio yang besar tersebut ia mengorbankan kualitasnya
3. Ketika nilai k semakin besar maka kualitas dari gambar kompresi akan semakin besar Hal ini tercermin dari MSE yang menurun, PSNR yang meningkat, dan SSIM yang mendekati 1. .Namun kenaikan tersebut diiringi oleh berkurangnya rasio kompresi karena nilai k yang dipakai semakin mendekati rank asli dari gambar yang di kompresi.
4. Nilai k yang efektif untuk gambar yang berbeda dapat sangat bervariasi .Seperti pada table gambar anjing dan gambar simple nilai k yang paling efektifnya bisa sangat berbeda. Pada gambar simple nilai yang efektif dengan kualitas paling mendekati adalah saat k = 20 sementara pada gambar anjing yang lebih kompleks nilai yang efektifnya disekitar k = 100 atau 50 dengan MSE <30 dan nilai SSIM yang berada disekitar 0,90
5. Berdasarkan poin di atas, pemilihan nilai k harus ditinjau berdasarkan kebutuhan aplikasi. Jika prioritasnya adalah kualitas gambar, maka nilai k yang lebih besar sebaiknya dipilih. Sebaliknya, jika

efisiensi ruang penyimpanan lebih penting, maka nilai k yang lebih kecil dapat dipertimbangkan.

6. Semakin sederhana pola suatu gambar, semakin sedikit nilai k yang dibutuhkan untuk mencapai kualitas kompresi yang baik. Ini menunjukkan bahwa pola gambar memengaruhi tingkat efisiensi kompresi.
7. Berdasarkan penjelasan pada dasar teori, nilai k yang dipilih harus memenuhi persamaan $k < (m*n / (m+n+1))$. Untuk gambar anjing, nilai k harus kurang dari 479, sedangkan untuk gambar simple, nilai k harus kurang dari 359. Jika nilai k yang dipilih adalah 100 untuk gambar anjing dan 20 untuk gambar simple, maka nilai k yang dipilih telah memenuhi persyaratan kompresi.
8. Pada table tersebut kita dapat melihat bahwa nilai dari metric SSIM > PSNR untuk semua nilai k yang dicoba. Walaupun SSIM dan PSNR sama sama menilai kualitas dari gambar kompresi tetapi kedua metric tersebut merepresentasikan hal berbeda. Nilai SSIM yang tinggi mengindikasikan bahwa hasil kompresi mempertahankan struktur visual utama dengan baik, meskipun PSNR menunjukkan adanya perbedaan piksel numerik yang cukup signifikan. Oleh karena itu, kualitas visual gambar kompresi tetap dianggap baik meskipun PSNR belum mencapai nilai yang sangat tinggi misal > 50.
9. Gambar sederhana (600x900) memiliki batas nilai k maksimum yang lebih kecil dibandingkan gambar anjing (800x1200). Ini menunjukkan bahwa gambar dengan dimensi lebih kecil dapat dikompresi lebih efektif menggunakan nilai k yang lebih kecil tanpa kehilangan kualitas visual yang signifikan.

Poin poin yang kita dapatkan dari table tersebut dapat dilihat hubungannya dengan lebih baik melalui grafik berikut:



Berdasarkan grafis hubungan banyak singular value dengan beberapa metric kualitas dapat diambil kesimpulan bahwa secara garis besar semakin besar atau banyak nilai k / singular yang diambil semakin baik kualitas gambar ditandai dengan nilai PSNR dan SSIM yang semakin besar k semakin membesar juga nilai mereka dan semakin kecil nilai MSE. Walaupun secara garis besar memiliki pola yang sama dalam beberapa metric kualitas seperti PSNR dan SSIM terdapat beberapa perbedaan dalam representasinya hal ini disebabkan oleh perbedaan kompleksitas gambar yang dikompresi.

VI. CONCLUSIONS

Berdasarkan hasil pengamatan dan eksperimen kita dapat mengambil kesimpulan bahwa SVD adalah salah satu Teknik kompresi yang cukup baik karena dapat dengan mudah mendeteksi pola dari suatu gambar dan mempertahankan pola pola tersebut pada gambar hasil kompresi. Namun disini lain Teknik ini bukanlah yang paling optimal dibidangnya. Hal ini terjadi karena Teknik Single Value Decomposition atau SVD membutuhkan sumber daya komputasi yang lebih tinggi disbanding Teknik kompresi lossy lain seperti JPEG. Walaupun memiliki kekurangan tersebut Teknik SVD dalam bidang kompresi lossy memiliki beberapa kelebihan dibandingkan JPEG dimana Teknik SVD bisa lebih fleksibel untuk menentukan seberapa besar rasio kompresi dari suatu gambar yang akan dikompresi. Walaupun lebih fleksibel, untuk menentukan k yang paling optimal masih diperlukan pengamatan dan pembobotan untuk menimbang prioritas dari pengaplikasiannya apakah lebih mementingkan ukuran atau kualitas atau keduanya. Secara garis besar algoritma kompresi dengan SVD memiliki banyak potensi terutama pada bidang bidang seperti machine learning dll.

VII. REFERENCES

1. <https://ieeexplore.ieee.org/abstract/document/4426357>. Diakses 28 Desember 2024
2. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=fdec68af9beb220cf76dab8f65164cf5487fb4d9>. Diakses 28 Desember 2024
3. <https://www.math.utah.edu/~gustafso/s2019/2270/projects-2017/markVanderMerweAnnWilcoxAndrewHaa/~/ImageCompressionreport.pdf>. Diakses pada 29 Desember 2024
4. <https://sites.math.washington.edu/~morrow/49813/svdphoto.pdf>. Diakses pada 29 Desember 2024
5. <https://ijarce.com/upload/2016/december-16/IJARCCCE%2046.pdf>. Diakses pada 29 Desember 2024.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 Desember 2024



Ferdinand Gabe Tua Sinaga dan 13523051