

APLIKASI DEKOMPOSISI NILAI SINGULAR PADA KOMPRESI UKURAN FILE GAMBAR

AMANATUL FIRDAUSI, MAHDHIVAN SYAFWAN, NOVA NOLIZA BAKAR

*Program Studi Matematika,
Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Andalas,
Kampus UNAND Limau Manis Padang, Indonesia.*

Abstrak. Salah satu isu yang sering menjadi kendala dalam pengolahan gambar digital adalah bagaimana cara memperkecil (kompresi) ukuran file gambar tanpa mengurangi kualitas gambar tersebut. Sebuah gambar sejatinya merupakan representasi dari suatu matriks. Dengan menggunakan metode dekomposisi nilai singular (tereduksi), dapat dibuat taksiran terhadap matriks tersebut, sehingga matriks yang dihasilkan dapat merepresentasikan gambar yang sama dengan ukuran file yang lebih kecil. Aplikasi metode dekomposisi nilai singular ini dalam kompresi gambar diimplementasikan dalam pemrograman MATLAB.

Kata Kunci: Dekomposisi nilai singular, kompresi gambar

1. Pendahuluan

Misalkan terdapat sebuah gambar hitam putih berukuran 9 *megapixel*, yaitu 3000×3000 *pixel* (matriks 3000×3000). Setiap *pixel* memiliki tingkatan abu-abu yang nilainya diberikan dalam bilangan bulat dari 0 sampai 255 (0 berarti hitam, 255 berarti putih). Masing-masing *pixel* membutuhkan kira-kira 1 *byte* untuk penyimpanan, sehingga gambar tersebut membutuhkan ruang penyimpanan sekitar 8,6 *megabyte*. Lebih lanjut, sebuah gambar berwarna biasanya mempunyai tiga komponen: *red*, *green*, *blue* [RGB]. Masing-masing dari warna ini diwakili oleh satu matriks, sehingga gambar berwarna berukuran 9 *megapixel* membutuhkan ruang penyimpanan tiga kali lebih besar dari gambar hitam putih (yaitu sekitar 25,8 *megabyte*) [4].

Salah satu cara yang dapat dilakukan untuk mengkompresi ukuran file gambar adalah dengan menerapkan teknik dekomposisi nilai singular. Dekomposisi nilai singular, salah satu alat yang sering dipakai pada aljabar linier, adalah faktorisasi dan teknik aproksimasi yang secara efektif dapat mereduksi setiap matriks menjadi matriks persegi yang dapat dibalik dengan ukuran yang lebih kecil.

Kajian teoritis dari dekomposisi nilai singular ini telah dibahas oleh Devina Amalia [5]. Selanjutnya penulis membahas aplikasi dekomposisi nilai singular pada masalah kompresi ukuran file gambar digital.

2. Dekomposisi Nilai Singular

Secara umum, dekomposisi nilai singular adalah penyusunan kembali suatu matriks atas beberapa matriks baru yang berhubungan dengan nilai singularnya. Berikut

diberikan teorema tentang dekomposisi nilai singular tersebut.

Teorema 2.1. [1] *Jika A adalah matriks $m \times n$ dengan rank k , maka A dapat difaktorkan menjadi*

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

$$= [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_k | \mathbf{u}_{k+1} \ \cdots \ \mathbf{u}_m] \left[\begin{array}{cccc|c} \sigma_1 & 0 & \cdots & 0 & \\ 0 & \sigma_2 & \cdots & 0 & \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & \sigma_k & \\ \hline & & & & 0_{(m-k) \times (n-k)} \end{array} \right] \begin{bmatrix} \mathbf{v}_1^t \\ \mathbf{v}_2^t \\ \vdots \\ \mathbf{v}_k^t \\ \mathbf{v}_{k+1}^t \\ \vdots \\ \mathbf{v}_n^t \end{bmatrix} \quad (2.1)$$

dimana U, Σ, V berturut-turut mempunyai ukuran $m \times m$, $m \times n$ dan $n \times n$, dan

- (a) $V = [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_n]$ mendiagonalkan $A^t A$ secara ortogonal.
- (b) Entri diagonal taknol dari Σ adalah $\sigma_1 = \sqrt{\lambda_1}$, $\sigma_2 = \sqrt{\lambda_2}$, \cdots , $\sigma_k = \sqrt{\lambda_k}$ dimana $\lambda_1, \lambda_2, \cdots, \lambda_k$ adalah nilai eigen taknol dari $A^t A$ yang bersesuaian dengan vektor-vektor kolom V .
- (c) Vektor-vektor kolom V diurutkan sedemikian sehingga $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0$.
- (d) $\mathbf{u}_i = \frac{A\mathbf{v}_i}{\|A\mathbf{v}_i\|} = \frac{1}{\sigma_i} A\mathbf{v}_i$ ($i = 1, 2, \cdots, k$).
- (e) $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\}$ adalah basis ortonormal untuk $\text{col}(A)$.
- (f) $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k, \mathbf{u}_{k+1}, \cdots, \mathbf{u}_m\}$ adalah perluasan dari $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\}$ terhadap suatu basis ortonormal untuk \mathbb{R}^m .

Bukti. Teorema ini akan dibuktikan dengan menggunakan matriks $n \times n$. Untuk kasus matriks $m \times n$ hanya perlu dimodifikasi dengan menyesuaikan notasi yang diperlukan untuk memperhitungkan kemungkinan $m > n$ atau $n > m$. Perhatikan bahwa matriks $A^t A$ adalah simetris, sehingga ia mempunyai dekomposisi nilai eigen

$$A^t A = V D V^t$$

dimana vektor-vektor kolom dari

$$V = [\mathbf{v}_1 | \mathbf{v}_2 | \cdots | \mathbf{v}_n]$$

adalah vektor-vektor eigen satuan dari $A^t A$, dan D adalah matriks diagonal yang entri-entri diagonal utamanya, berturut-turut merupakan nilai-nilai eigen dari $A^t A$ yang bersesuaian dengan vektor-vektor kolom dari V , misalkan $\lambda_1, \lambda_2, \cdots, \lambda_n$.

Selanjutnya karena A diasumsikan memiliki rank k , maka $A^t A$ juga memiliki rank k . Akibatnya D juga mempunyai rank k . Jadi D dapat dinyatakan dengan

$$D = \begin{pmatrix} \lambda_1 & & & & 0 \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \lambda_k & \\ & & & & 0 \\ & & & & & \ddots \\ 0 & & & & & & 0 \end{pmatrix} \quad (2.2)$$

Dari (2.4) dan fakta bahwa $A\mathbf{v}_i = 0$ untuk $i > k$, maka

$$\begin{aligned} U\Sigma &= [\sigma_1\mathbf{u}_1 \ \sigma_2\mathbf{u}_2 \ \cdots \ \sigma_k\mathbf{u}_k \ 0 \ \cdots \ 0] \\ &= [A\mathbf{v}_1 \ A\mathbf{v}_2 \ \cdots \ A\mathbf{v}_k \ A\mathbf{v}_{k+1} \ \cdots \ A\mathbf{v}_n] \\ &= AV. \end{aligned}$$

Dengan menggunakan ortogonalitas V , bentuk terakhir dapat ditulis kembali menjadi

$$A = U\Sigma V^t. \quad \square$$

3. Dekomposisi Nilai Singular Tereduksi

Secara aljabar, baris nol dan kolom nol dari matriks Σ pada Teorema 1 dinilai mubazir dan dapat dihilangkan dengan melakukan perkalian blok dan partisi matriks pada bentuk $U\Sigma V^t$ seperti yang ditunjukkan pada rumus. Hasil perkalian matriks tersebut akan memuat blok nol sebagai faktor yang dapat dikeluarkan, dan menyisakan

$$A = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_k] \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^t \\ \mathbf{v}_2^t \\ \vdots \\ \mathbf{v}_k^t \end{bmatrix}. \quad (3.1)$$

Persamaan (3.1) disebut **dekomposisi nilai singular tereduksi** dari A . Selanjutnya matriks-matriks pada ruas kanan persamaan (3.1) berturut-turut dinotasikan dengan U_1 , Σ_1 , dan V_1^t , sehingga persamaan tersebut dapat ditulis

$$A = U_1 \Sigma_1 V_1^t.$$

Perhatikan bahwa ukuran U_1 , Σ_1 , dan V_1^t masing-masing adalah $m \times k$, $k \times k$, dan $k \times n$. Perhatikan juga bahwa matriks Σ_1 dapat dibalik karena entri-entri diagonalnya positif.

Jika perkalian pada ruas kanan persamaan (3.1) dilakukan dengan menggunakan aturan kolom-baris, maka diperoleh

$$A = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^t + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^t + \cdots + \sigma_k \mathbf{u}_k \mathbf{v}_k^t,$$

yang disebut **ekspansi nilai singular tereduksi** dari A . Hasil ini berlaku untuk semua jenis matriks (baik simetris maupun tak-simetris).

Sebagai contoh, misalkan ingin ditentukan dekomposisi nilai singular tereduksi

dan ekspansi nilai singular tereduksi dari matriks $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Dekomposisi nilai singular dari A , diberikan oleh

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{6}}{3} & 0 & -\frac{1}{\sqrt{3}} \\ \frac{\sqrt{6}}{6} & -\frac{\sqrt{2}}{2} & \frac{1}{\sqrt{3}} \\ \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{2} & \frac{1}{\sqrt{3}} \end{pmatrix} \begin{pmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{pmatrix}. \quad (3.2)$$

Karena A memiliki rank 2, maka dekomposisi nilai singular tereduksi dari A yang berkorespondensi dengan (3.2) adalah

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{6}}{3} & 0 \\ \frac{\sqrt{6}}{6} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} \sqrt{3} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{pmatrix}.$$

Ini menghasilkan ekspansi nilai singular tereduksi

$$\begin{aligned} \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} &= \sigma_1 \mathbf{u}_1 \mathbf{v}_1^t + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^t \\ &= \sqrt{3} \begin{pmatrix} \frac{\sqrt{6}}{3} \\ \frac{\sqrt{6}}{6} \\ \frac{\sqrt{6}}{6} \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} + (1) \begin{pmatrix} 0 \\ -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{pmatrix} \\ &= \sqrt{3} \begin{pmatrix} \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{6} \\ \frac{\sqrt{3}}{6} & \frac{\sqrt{3}}{6} \end{pmatrix} + (1) \begin{pmatrix} 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{pmatrix}. \end{aligned}$$

Sekarang misalkan nilai-nilai singular $\sigma_{r+1}, \dots, \sigma_k$ cukup kecil sedemikian sehingga dengan mengabaikan nilai-nilai singular tersebut diperoleh aproksimasi

$$A_r = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^t + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^t + \dots + \sigma_r \mathbf{u}_r \mathbf{v}_r^t \quad (3.3)$$

yang 'dapat diterima' untuk A . Untuk selanjutnya persamaan (3.3) disebut **aproksimasi rank r dari A** .

4. Implementasi Kompresi Ukuran File Gambar pada MATLAB

4.1. Proses Input Gambar

Gambar yang akan dikompresi pada program MATLAB ini adalah gambar berwarna (dengan komponen RGB [Red, Green, Blue]). Sebagai contoh ilustrasi pada simulasi ini, gambar yang digunakan adalah gambar baboon (lihat Gambar 1(a)). Potongan dari Gambar 1(a) di pojok kiri atas yang berukuran 8×8 *pixel* ditunjukkan pada Gambar 1(b).

4.2. Algoritma Pemrograman

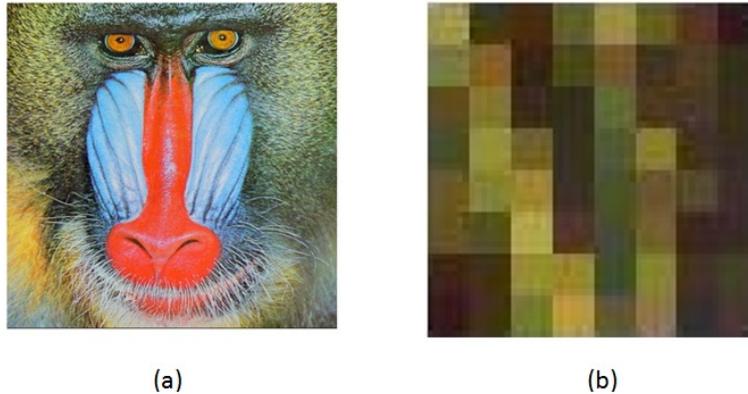
Algoritma pemrograman dari kompresi ukuran file gambar dengan menggunakan dekomposisi nilai singular adalah sebagai berikut:

Masukan: Gambar yang akan dikompresi
 r banyak rank dalam aproksimasi matriks

Keluaran: Gambar hasil kompresi

Langkah-Langkah:

1. Gambar dibaca sebagai tiga buah matriks
2. Definisikan secara terpisah matriks penyusun merah, hijau, dan biru (RGB)



Gambar 1. (a) Gambar yang akan dikompresi. (b) Potongan Gambar (a) di pojok kiri atas.

3. Konversi entri matriks antara 0 dan 1
4. Lakukan dekomposisi nilai singular untuk setiap matriks penyusun RGB (dalam hal ini digunakan *function svd.m* yang sudah *built-in* pada MATLAB)
5. Rekonstruksi matriks aproksimasi rank r untuk setiap komponen RGB [gunakan persamaan (3.3)]
6. Gabung matriks penyusun RGB hasil dari langkah 5
7. Tampilkan gambar hasil kompresi

4.3. Proses Simulasi

Berikut dijelaskan proses simulasi program pada setiap langkah

Langkah 1. Membaca file gambar.

Untuk membaca Gambar 1(a) pada MATLAB, digunakan perintah

```
>> Gambar1=imread('Gambar1.bmp')
```

Perintah di atas mengkonversi Gambar 1(a) menjadi data angka yang tersimpan dalam tiga buah matriks yang masing-masing menyusun warna merah, hijau, dan biru untuk setiap *pixel*. Setiap entri matriks menandakan kepekatan warna.

Langkah 2. Definisikan matriks penyusun warna merah, hijau dan biru.

Misalkan matriks yang merepresentasikan warna merah adalah matriks A_r . Dengan menuliskan perintah berikut:

```
>> Ar=Gambar1(:,:,1)
```

diperoleh matriks A_r yang membangun warna merah untuk Gambar 1(a). Hal yang sama dapat dilakukan untuk mendefinisikan A_g sebagai matriks penyusun warna hijau dan A_b sebagai matriks penyusun warna biru, dengan menuliskan perintah

```
>> Ag=Gambar1(:,:,2)
```

```
>> Ab=Gambar1(:,:,3)
```

Langkah 3. Konversi entri matriks antara 0 dan 1.

Sebelum menghitung dekomposisi nilai singular dari masing-masing matriks penyusun warna, nilai entri-entri matriks dikonversi antara 0 dan 1. Hal ini dilakukan agar perhitungan nantinya menjadi lebih efisien (tidak melibatkan bilangan besar). Jika x_{ij} adalah elemen matriks awal yang bernilai bilangan bulat antara 0 dan 255, maka elemen matriks hasil konversi y_{ij} adalah

$$y_{ij} = \frac{x_{ij}}{255}.$$

Pada MATLAB, hal ini dapat dilakukan dengan menggunakan perintah:

```
>> AR=im2double(AR)
>> AG=im2double(AG)
>> AB=im2double(AB)
```

Langkah 4. Menghitung dekomposisi nilai singular matriks penyusun warna.

Untuk menghitung dekomposisi nilai singular dari matriks AR, AG dan AB, digunakan *function* `svd.m`, yaitu dengan menuliskan perintah berikut:

```
>> [UR,SR,VR]=svd(AR)
```

output UR, SR, dan VR berturut-turut menyatakan matriks U , Σ , dan V pada persamaan (2.1).

Langkah 5. Bentuk matriks aproksimasi rank r untuk penyusun warna merah, hijau, dan biru.

Seperti yang telah dibahas sebelumnya, aproksimasi rank r dari matriks A mengikuti persamaan

$$A_r = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^t + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^t + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^t.$$

Kemudian untuk mencari matriks aproksimasi rank r (misalkan $r = 5$) dari matriks AR, ditulis perintah

```
>> C=zeros(size(AR))
>> r=5;
>> for j=1:r
C=C+SR(j,j)*UR(:,j)*VR(:,j).';
end
```

Dengan melakukan hal yang sama, definisikan juga E sebagai matriks aproksimasi penyusun warna hijau dengan rank $r = 5$ dan F sebagai matriks aproksimasi penyusun warna biru dengan rank $r = 5$.

Langkah 6. Susun kembali matriks aproksimasi penyusun warna.

Untuk menyimpan kembali matriks penyusun warna ke dalam suatu matriks dituliskan perintah

```
>> Anew=cat(3,C,E,F);
```

Kemudian diperoleh matriks Anew sebagai matriks hasil kompresi dari A.

Langkah 7. Tampilkan hasil.

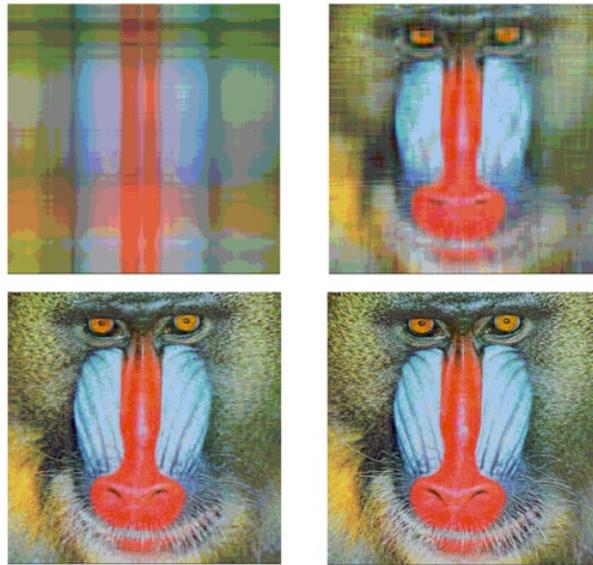
Untuk menampilkan gambar hasil kompresi, digunakan perintah berikut:

```
>> imshow(Anew)
>> print -dbitmap 'babon400.bmp'
```

4.4. Hasil

Sebagai ilustrasi, misalkan Gambar 1(a) akan dikompresi. Gambar tersebut berukuran 512×512 *pixel* dengan ukuran file sebesar 769 *KB*. Dengan menggunakan perintah `imread`, gambar tersebut terbaca sebagai tiga buah matriks (matriks penyusun merah, hijau, dan biru) yang berukuran 512×512 .

Pada Gambar 2 ditampilkan hasil kompresi dengan aproksimasi rank $r = 1, 10, 50, 100$. Dari gambar dapat dilihat bahwa gambar hasil kompresi semakin mendekati gambar asli untuk nilai r yang semakin besar. Untuk gambar hasil kompresi dengan aproksimasi rank 100 dapat dilihat bahwa kualitasnya tidak begitu jauh berbeda jika dibandingkan dengan gambar asli, padahal file gambar hasil kompresi tersebut hanya berukuran sebesar 396 *KB* atau sekitar 51,5% lebih kecil dari ukuran file gambar asli.



Gambar 2. Gambar hasil kompresi dengan aproksimasi rank $r = 1$ (kiri atas), 10 (kanan atas), 50 (kiri bawah), dan 100 (kanan bawah).

5. Ucapan Terima kasih

Penulis mengucapkan terima kasih kepada Bapak Narwen, M.Si, Bapak Dr. Admi Nazra dan Bapak Zulakmal, M.Si yang telah memberikan masukan dan saran sehingga paper ini dapat diselesaikan dengan baik.

Daftar Pustaka

- [1] Anton, H and Chris R. 2013. *Elementary Linear Algebra 11th edition*. Wiley, New Jersey
- [2] Jacob, Bill. 1990. *Linear Algebra*. W.H. Freeman and Company. New York
- [3] Leon, Steven. J. 2001. *Aljabar Linear dan Aplikasinya*. Edisi ke-5. Erlangga, Jakarta
- [4] Dostert, Paul. 2009. *An Application of Linear Algebra to Image Compression*. (<http://math.arizona.edu/brio/VIGRE/ThursdayTalk.pdf>, diakses 16 Januari 2015)
- [5] Amelia, Devina. 2006. Faktorisasi Matriks dengan Menggunakan Metode Dekomposisi Nilai Singular. Skripsi Sarjana Matematika Universitas Andalas (tidak diterbitkan)